

# Tézy na skúšku z INFORMATIKY.

**1. Jednorozmerné pole - načítanie a výpis údajov** Jednorozmerné pole patrí medzi zložené typy údajov. Zložené typy údajov sú také, ktorých hodnoty môžu byť zložené z viacerých jednoduchých hodnôt (položiek). Zložené typy pole – pevný počet zložiek rovnakého typu údajov záznam – pevný počet zložiek rôznych typov

**Definícia poľa:** *typ premenná [n]*; Premenná sa skladá z **n** zložiek(okienok):



**1 2 3 4 . . . . . n** Hodnota **n** predstavuje maximálny počet položiek poľa. Poradovým číslom okienok hovoríme indexy a pomocou nich sa odvolávame na konkrétne okienko, napr. ak máme pole ceny[5], tak na cenu v 3. chate sa odvoláme ceny[2], lebo v jazyku C sa automaticky číslujú okienka od nuly. Index **i** je celé číslo, deklaruje ho takto: `int i`; Na cenu v *i*-tej chate sa odvoláme ceny[i].

**Príklad 1:** Zostavte program, ktorý načíta počet čísel (maximálny počet čísel je 25) a potom jednotlivé čísla do jednorozmerného poľa. Obsah poľa vypíše na obrazovku a pripíše k nemu súčet zadaných čísel a ich aritmetický priemer.

**Riešenie:**

```
#include <stdio.h>
#include <conio.h>
int i,n;
float a[25],ap,sum;
main()
{clrscr(); // citanie vstupnych udajov do pola
printf("Zadaj pocet cisel: ");
scanf("%d",&n);
for(i=0;i<n;i++)
{printf("\nZadaj %d. cislo: ",i+1);
scanf("%f",&a[i]);
}
// vypocty vysledkov
sum=0.0;
for(i=0;i<n;i++) sum+=a[i];
ap=sum/(1.0*n); // vypis obsahu pola a vysledkov
clrscr();
printf("\n\nNacital som tieto cisla:");
for(i=0;i<n;i++) printf(" %.2f",a[i]);
printf("\n\nSucet zadanых cisel je %.2f",sum);
printf("\nAritmeticky priemer zadanых cisel je %.2f",ap);
if(getch()==0) getch();
}
```

## 2.úloha – c, Autorské právo, prípady jeho porušovania

### Autorské práva –

Autor programu alebo firma, ktorá program vytvorí, sú chránení autorským zákonom.

**Autorský zákon** určuje práva povinnosti autora, t.j. autor má právo rozhodovať o tom, ako bude sa jeho dielo (program) používať a šíriť.

### Licencia (licenčná zmluva)

- Pri zakúpení programu (softvéru) získame licenciu, ktorá stanovuje, za akých podmienok sa môže používať. Ak dodržíme tieto podmienky, hovoríme o **legálnom** softvéri.
- Licencie sú súčasťou programu a vzťahuje sa na **jeden PC**.
- Pre viac počítačov sa zakupuje **multilicencia**, ktorá je lacnejšia a využívajú ju firmy a školy.

### Typy licencií

- **Študentská/ školská licencia** – je lacnejšia, výhodnejšia pre školské inštitúcie. Je zakázané používať na komerčné účely.
- **Krabicová licencia** – veľká škatuľa obsahuje CD/DVD, licenčnú zmluvu, licenčné číslo, manuál k používaniu a registračná karta. Po zaregistrovaní sú k dispozícii používateľovi nové verzie programu, doplnky, upgrade (aktualizácia) programu.
- **OEM licencia** – licencia patrí medzi tie lacnejšie a napríklad operačné systémy (ako Windows XP, Vista) sa predávajú spolu s novým PC. Daný operačný systém/ program nemožno inštalovať na iný PC aj po ukončení životnosti daného PC.
- **Shareware** – program môžeme nainštalovať, pracovať s ním a po určitom čase sme povinní zaplatiť registračný poplatok. Potom môžeme legálne používať program.
- **Freeware** – program môžeme používať a rozširovať zadarmo. Treba dodržiavať autorské práva, t.j. neupravovať program
- **Demoverzie** – sú to programy, ktoré majú zablokované niektoré funkcie (napr. tlač). Slúžia pre tých, ktorí potrebujú zistiť, či daný program spĺňa ich požiadavky. Ak áno, zakúpia si ho.
- **Trial** – umožňuje plnohodnotnú prácu s programom ale len na určité časové obdobie. Po uplynutí času sa program **nedá znovu** spustiť ani nainštalovať.
- **Public domain** – program určený k voľnému používaniu. Je možné ho šíriť i upravovať (myslí sa programátorsky upravovať). Autori sa vzdávajú niektorých svojich práv na program. Napr. operačný systém typu Unix (napr. Linux).

V praxi narastá i počet pokusov o ich zneužitie, ktoré často hraničia so zákonom. Jedná sa buď o vedomé cielené zneužívanie alebo o prejavy nevedomosti či nedbalosti. Používateľ výpočtovej techniky by mal mať základné vedomosti o právnej stránke využívania softvéru. Softvérové pirátstvo je nezákonné zneužívanie softvéru, najčastejšie nezákonné kopírovanie programov pre vlastnú potrebu, prípadne kopírovanie a následný predaj. Často sa vykrádajú postupy a kódy používané v iných programoch. Na boj proti softvérovému pirátstvu vznikla Business Software Alliance.

<https://gallova.wordpress.com/2011/04/05/autorske-prava-a-licencie/>

## 2.

### 1. úloha: Vysvetlite kedy je potrebné použiť vnorené podmienky v algoritme. Prečo nie je možné vždy použiť príkaz viacnásobného vetvenia ?

Príkaz vetvenia vykonáva určité príkazy, ak je výraz pravdivý, teda má nenulovú hodnotu. Ak je výraz nepravdivý máme dve možnosti buď pokračujeme v programe, potom kľúčové slovo pre tento príkaz je **if**. Alebo sa vykonávajú príkazy pre nepravdivý výraz, potom kľúčové slovo pre takýto príkaz je **if else**. Ak sa rozhodujeme medzi viacerými možnosťami potom používame príkaz **switch** ( prepínač ). Niekedy je možné nahradiť tento príkaz vnoreným **if else**. Príkaz sa vykoná ak má výraz nenulovú hodnotu, teda podmienka je splnená. Ak je príkazov viac musia byť ohraničené zloženými zátvorkami { }.

### Príkaz zložený z viacerých príkazov if else

Pri vetvení je možné príkazy **if else** do seba vnárať. Pri vzájomnom vnáraní sa časť **else** automaticky priradí najbližšiemu edná sa o viaccestné rozhodovanie (prepínač). Výrazy sa vyhodnocujú v uvedenom poradí, pokiaľ sa nenarazí na pravdivý výraz. Potom sa vykoná k nemu prislúchajúci príkaz. Príkaz za posledným **else** sa vykoná ak žiaden z predchádzajúcich výrazov nebol pravdivý. Môže to byť napríklad chybové hlásenie predchádzajúcemu **if**.

## 2.úloha: Prínosy a negatíva budovania informačnej spoločnosti

**Informačná spoločnosť** je taká spoločnosť, kde práca s informáciami je každodennou záležitosťou. Na prácu s informáciami používame rôzne **informačné a komunikačné technológie (IKT)**, teda techniky, metódy, postupy a prostriedky ako počítač, kalkulačku, elektronický diár, mobilný telefón, ... (**Informatika** je vedný odbor, ktorý sa zaoberá algoritmami a zberom, spracúvaním, vyhľadávaním a prenosom informácií pomocou počítačov a ich programov.)

Prínos informačnej spoločnosti:	Negatíva informačnej spoločnosti
<ul style="list-style-type: none"><li>• Uľahčujú nám prácu s informáciami – získavanie, ukladanie, spracovanie a šírenie.</li><li>• Umožňujú rýchlu a jednoduchú komunikáciu.</li><li>• Sprístupnenie informačných zdrojov pre využívanie širokou verejnosťou.</li><li>• Výkonnejšia štátna správa.</li><li>• Efektívnejšie riadenie organizácií, zvýšenie konkurencieschopnosti.</li><li>• Podpora vzdelávania.</li><li>• Rozšírenie ponuky služieb a zábavy.</li><li>• Nové služby v rámci telekomunikácie.</li><li>• Nové možnosti uplatnenia pre tvorivých ľudí.</li></ul>	<ul style="list-style-type: none"><li>• Nutnosť vysokých počiatočných investícií.</li><li>• Silná závislosť na prvkoch Informačných technológií (elektromagnetická búrka alebo jadrový výbuch by nás pripravil o väčšinu informácií).</li><li>• Nebezpečenstvo terorizmu a zločinov v tejto oblasti.</li><li>• Zmena hodnôt a životného štýlu (závislosť, únik od reality).</li></ul>

**Informačné systémy** predstavujú najčastejší spôsob využívania počítačových systémov. Je to systém určený na zber, udržiavanie a spracúvanie údajov. Príkladom informačných systémov sú rozličné administratívne a logické systémy (účtovníctvo, skladové hospodárstvo, systém na elektronický obchod...) informačné systémy v štátnej správe (polícia – evidencia vozidiel..., mesto – evidencia obyvateľov,...) telekomunikáciách, školstve, zdravotníctve (informačné systémy, evidencie pacientov, zdravotné poisťovne,...), geografické informačné systémy,...

### Súčasný trendy

To čo je dnes realitou bolo pred pár rokmi sci-fi. Informačná spoločnosť prechádza sústavným vývojom. Vedný odbor **Umelá inteligencia** sa zaoberá vytváraním strojov alebo systémov, ktoré používajú pri riešení úloh postupy, ktoré, ak by ich riešil človek, boli by považované za prejav jeho inteligencie. Výskum

umelej inteligencie ovplyvnil aj vývoj **expertných systémov** (počítačový program, ktorý simuluje rozhodovacie činnosti expertov pri riešení problémov úzko zameraných úloh), **robotov** (systém schopný nahradiť pohybové a rozumové funkcie človeka) a **systémy na rozpoznávanie reči**.

Stretávame sa aj s pojmom **virtuálna realita**, ktorá pomocou hardvéru, softvéru a množstva dát popisujúceho virtuálny svet. Softvér spracúva naše pohyby a na základe nich poskytuje používateľovi obraz a zvuk. **Nanotechnológia** je vedný odbor, ktorý sa zaoberá presnou a zámernou manipuláciou hmoty na úrovni atómov.

### **Vplyv informačných technológií na človeka**

Je potrebné vytvoriť si vhodné pracovné prostredie (správne rozmiestnenie jednotlivých častí PC zostavy, správne ergonomické sedenie, osvetlenie, farby, akustika, ) Zdravotné problémy, ktoré môžu byť spojené s použitím počítača: zrakové problémy, pohybové problémy, závislosť,.....

## **3.**

### **1.úloha: logická premenná, hodnoty logickej premennej**

#### **Relačné a logické operátory**

Relačné operátory majú všetky rovnakú prioritu.

Sú to: > , >= , < , <= , == , != . Najnižšiu prioritu má rovnosť == a nerovnosť != . Relačné operátory majú nižšiu prioritu ako aritmetické.

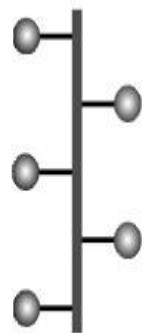
Logické operátory sú: logický súčin && , logický súčet || a negácia ! . Najvyššiu priority má negácia. potom logický súčin a nakoniec logický súčet. Logické operátory majú nižšiu prioritu ako relačné. Operátorom s najnižšou prioritou je operátor priradenia.

### **2.úloha: 2. delenie pc sieti podľa fyzickej topológie**

Fyzická topológia je geografická štruktúra, podľa ktorej sú prepojené jednotlivé uzly v sieti. Medzi základné typy topológie siete LAN patrí: **zbernica, hviezda a kruh**. Zložitejšia topológia sietí môže obsahovať kombináciu základných typov, najčastejšie kombinácia hviezd do **stromovej štruktúry**, topológiu **úplnú** a **chrbticovú**.

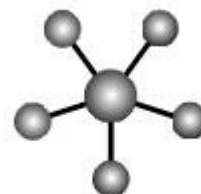
#### **zbernicová**

Základným prvkom zbernicovej topológie je **zbernica** - úsek prenosového média (kábel), ktorá tvorí kostru siete. K nej sú jednotlivé stanice siete pripojené pomocou odbočovacích prvkov jedna vedľa druhej. Prenosovým médium je najčastejšie koaxiálny kábel (s prenosovou rýchlosťou 10 Mbit/s) alebo krútená dvojlinka. V zbernicovej topológii sa nevyskytuje centrálna alebo radiaca stanica. Dátové správy sa šíria vedením všetkými smermi a všetky stanice k nim majú prístup. Sieť zbernicovej topológie je najjednoduchšia a veľmi ľahko sa inštaluje. Zbernica má jeden začiatok a jeden koniec a musí byť ukončená terminátorom (odpor zamedzujúci „odrazu dátových správ“). Veľkou výhodou tejto topológie je funkčnosť siete v prípade výpadku jednej stanice, ale naopak, pri prerušení kábla (prenosového média) sa stáva sieť nefunkčnou. V súčasnosti sa táto topológia využíva len zriedka.



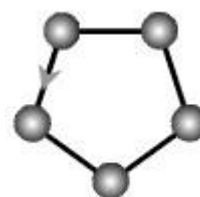
## hviezdicová

V sieti hviezdicovej topológie pôsobí v centre siete **centrálny uzol**, ktorým môže byť prepínač (switch), v starších sieťach rozbočovač (hub) alebo, najmä pri bezdrôtových sieťach, opakovač (repeater). K centrálnemu uzlu sú pripojené stanice siete samostatnými linkami, najčastejšie pomocou symetrického kábla (krútená dvojlinka). Pri poruche centrálného uzla je celá sieť vyradená z prevádzky.



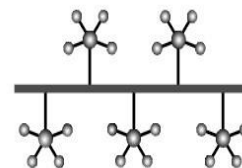
## kruhová (prstencová)

Stanice siete sú prepojené vedením do tvaru súvislého kruhu. Dáta sa pohybujú v kruhu od odosielateľa (prechádzajú postupne k najbližšiemu susedovi) postupne cez všetky následné uzly až k príjemcovi (adresovanej stanici) – smer pohybu je daný spôsobom prepojenia siete. Na riadenie smeru prenosu dát sa používa riadiaca značka - Token. Pomocou nej sa dátové správy odovzdávajú postupne jedným smerom medzi stanicami. Stanica siete, ktorá má riadiacu značku, môže vysielať, ostatné stanice môžu iba prijímať. Týmto je odstránená možnosť vzniku kolízií pri súčasnom vysielaní niekoľkými stanicami. Výpadok ľubovoľnej stanice spôsobí nefunkčnosť celej siete.



## stromová

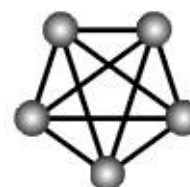
Stromová topológia je prirodzeným rozšírením topológie typu hviezda s kombináciou zbernice. Má aj podobné vlastnosti. Používa sa najčastejšie u širokopásmových sietí a optických vlákien.



## úplná

Každá stanica siete je prepojená zo všetkými ostatnými stanicami. Táto topológia vyžaduje veľký počet káblov. Je veľmi spoľahlivá, ale zle rozširiteľná. Používa sa málo.

V prípade, že nejaké spojenie zlyhá, dáta môžu putovať k cieľu ďalej po iných dostupných linkách (majú viac možností).



## chrbticová

Prepája jednotlivé LAN s ľubovoľnou topológiou.

## 3. vysvetlite pojmy licencia, registrácia , upgrade softveru

**Licencia** Povolenie, ktoré dovoľuje používať softvér na počítači užívateľa. Platí: 1 licencia = 1 počítač (v terminále platí 1 licencia = 1 naštartovaný proces aplikácie) s oficiálne nainštalovaným a registrovaným softvérom. V prípade zaobstarávania softvérového vybavenia nejde o kúpu softvéru, ako sa niekedy zjednodušene hovorí, ale len o získanie LICENCIE. Súčasťou každej dodávky originálneho produktu sú síce inštalačné médiá a manuál, ale hlavným predmetom kúpy je vždy nehmotný majetok - programové vybavenie, duševné vlastníctvo autora. Prostredníctvom svojich obchodných partnerov potom autor môže poskytnúť právo používať svoje duševné vlastníctvo ďalším subjektom - používateľom softvéru. Cena za softvér je teda de facto poplatkom za poskytnutú licenciu - právo na

používanie softvérového produktu. Právo softvér používať (licenciu) nemôžete bez súhlasu autora ďalej šíriť, predat' alebo prenajať. Tento výslovný súhlas obvykle autor udeľuje generálne prostredníctvom [licenčnej zmluvy](#) (End User License Agreement; EULA), ktorá je súčasťou každej dodávky softvéru. Niektoré licenčné zmluvy napr. umožňujú softvér používať v kancelárii aj doma alebo na prenosných počítačoch, u iných to môže byť striktné zakázané. Obvykle platí, že používaním (inštaláciou) softvéru dávate najavo svoj súhlas s ustanoveniami licenčnej zmluvy, prijímate ich ako záväzné a budete sa nimi riadiť.

**Registrácia** Proces, pri ktorom sa váš softvér stane oficiálnym a plne funkčným. Registrovať programy možno telefonicky s dodávateľom programu, alebo cez internet vyplnením registračných formulárov (viz. antivírusové programy, programy slúžiace na edukačné účely....**Aktualizácia programu, softvéru (upgrade)** Proces, pri ktorom dochádza k zmene databázy (pridanie nových tabuliek a dátových položiek do súčasných, vylepšenie funkcionalít programu) a nakopírovanie novej verzie programu, ide o vylepšenia programu (viz. Aktualizácia Windows

## )4.

### **1 .úloha: vysvetlite príkazy vstupu a priradenia, ďalšie možnosti načítania údajov do premenných**

#### **Premenná, výraz, priradenie, vstup údajov**

*Premenná* je časť operačnej pamäte. Každá premenná má svoje meno – *identifikátor*. Aby sme mohli v programe používať premenné, musíme požiadať systém o pridelenie operačnej pamäte pre ne a pomenovať ich. To všetko sa stane deklaráciou premennej. V programe potom už len používame mená premenných a systém vie, s ktorou časťou operačnej pamäte má pracovať. Takže, *deklarácia premennej* je fyzické pridelenie operačnej pamäte pre premennú a dočasné pomenovanie pridelenej pamäte identifikátorom. Aby systém vedel, koľko operačnej pamäte má pre premennú zadovážiť, musíme v deklarácii uviesť jej typ. Typ premennej zároveň určuje, čo do nej smieme uložiť. Jazyk C používa jednoduché a zložené typy premenných. Jednoduché typy sú pre celé čísla, racionálne čísla, znak a logické hodnoty. Zložené typy sú pole, záznam, súbor.

To, čo ste si prečítali doteraz bolo presné odborné vysvetlenie. Rozumeli ste tomu? Ak áno, preskočte tento odsek. Ak nie, pokračujeme obrazným vysvetlením. Skúste si premennú predstaviť ako škatuľku na topánky. Škatuľky môžu byť rôzneho druhu. Druh škatuľky určuje, čo do nej môžeme vložiť. Ak chceme používať v programe premenné – škatuľky, tak si ich musíme zadovážiť. Zadováži nám ich systém, len mu musíme povedať, koľko škatuliek akého druhu potrebujeme a každej škatuľke musíme dať meno – identifikátor, to aby systém vedel, s ktorou má pracovať. Príkazu na zadováženie si škatuliek a pridelenie mien škatuľkám hovoríme deklarácia.

*Deklarácia* v jazyku C začína typ premenných (druh škatuliek) a za ním nasledujú identifikátory premenných oddelené od seba čiarkami. Zoznam premenných je ukončený dvojbodkou, napr.

int a,b; (do premennej a i b sa vojde len jedno celé číslo)

S premennou môžeme robiť dve operácie:

1. uložiť (zapísať) do nej hodnotu – pri tejto operácii sa starý obsah premennej zlikviduje (navždy sa stratí) a do premennej sa uloží nová hodnota. Na to treba vždy pamätať. Do premennej uložíme hodnotu

- operáciou priradenia (= je operátorom priradenia)
- prečítaním hodnoty premennej zo vstupu (z klávesnice). Na čítanie hodnoty premennej používame podprogram z knižnice `stdio.h` `scanf(„%konverzia“, &premenná);` Pre celé čísla je konverzia `d`. Konverzia znamená, že znaky (čísllice) sa zmenia na celé číslo a to sa uloží do premennej. Znak `&` znamená, že podprogramu posielame adresu premennej. Poslať adresu je veľmi dôležité, lebo inak podprogram nedokáže uložiť načítanú hodnotu do premennej.

2. zistiť (prečítať) hodnotu premennej – táto operácia sa vykoná automaticky vždy, keď premennú použijeme v ľubovoľnom príkaze.

Povedzme si pár slov o priradení. *Priradenie* je jednoduchý príkaz, ktorým sa výsledná hodnota výrazu na pravej strane od operátora priradenia zapíše do premennej, ktorej identifikátor je uvedený na ľavej strane výrazu. Napr. `o=2*(a+b);` Tento príkaz priradenia sa vykoná takto: najprv procesor prečíta hodnotu premennej `a`, potom prečíta hodnotu premennej `b`. Prečítané hodnoty spočíta a vynásobí dvomi. Výsledok potom zapíše do premennej `o`. Ešte zopár postrehov. Na ľavej strane príkazu priradenia je vždy iba jedna premenná – cieľová. Na pravej strane je zvyčajne matematický výraz, ktorý pozostáva z operandov, medzi ktorými stoja matematické operátory, napr. `+`, `-`, `*`, `/`, ... Operandom môže byť konštanta, premenná alebo funkcia.

Keď zadávame údaje z klávesnice musíme ich počítaču odoslať, ukončiť zadávanie, potvrdiť klávesou `Enter`.

Obsah premennej vypíšeme podprogramom z knižnice `stdio.h` `printf(„%konverzia“, premenná);` Konverziou `%d` sa zmení celé číslo uložené v premennej na znaky (čísllice). Znak `&` sa v podprograme `printf` nepoužíva, lebo podprogramu stačí poslať hodnotu premennej. Keďže nepotrebuje meniť jej obsah, nepotrebuje ani jej adresu.

#### Príklad 1:

Zostavte program, ktorý do premennej `x` uloží náhodne vybrané číslo z množiny `{-100, -75, -50, -25, 0, 25, 50, 75, 100}` a potom toto číslo vypíše na obrazovku.

#### Riešenie:

Keďže prvky zadanej množiny sú celé čísla, tak premenná `x` musí byť typu `integer`, teda `int`. Vytvoríme si ju deklaráciou: `„int x;“` Do premennej `x` priradíme hodnotu výrazu, ktorým vyberieme náhodne celé číslo zo zadanej množiny. Výraz sme určili podľa vzorca z predchádzajúceho učiva. Príkazom `printf(„%d“, x);` vypíšeme na obrazovku hodnotu premennej `x`.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
```

```
main()
{int x;
```

```
clrscr();
```

```
randomize();
x=random(9)*25-100;
printf("%d",x);
if(getch()==0) getch();
}
```

### Príklad 2:

Zostavte program, ktorý načíta z klávesnice celé číslo a vypíše toto číslo na obrazovku.

### Riešenie:

Keďže z klávesnice máme načítať celé číslo, tak ho prečítame do celočíselnej premennej, teda premennej typu int. Ešte by sme jej mali dať meno, nech sa teda volá a. Aby užívateľ, ktorý spustil program, vedel, aké údaje od neho program požaduje, je nutné mu to oznámiť, napr. príkazom: printf(„Zadaj cele cislo: “); Príkaz: printf(„\nNacital som cislo: “,a); vypíše na obrazovku najprv prázdny riadok, reťazec „Načítal som cislo: “ a potom k nemu pripíše obsah premennej a.

```
#include <stdio.h>
#include <conio.h>
```

```
main()
{int a;

clrscr();
printf("Zadaj cele cislo: ");
scanf(„%d“,&a);
printf(„\nNacital som cislo %d“,a);
if(getch()==0) getch();
}
```

## **2.úloha: b. počítačová sieť, zdieľanie, prenosová rýchlosť, výhody a nevýhody pc sietí súčasnosti**

Počítačová sieť je sústava najmenej dvoch navzájom prepojených počítačov. Umožňuje rýchly a pohodlný prenos dát ako aj vzájomnú spoluprácu, zdieľanie súborov, prácu s jediným zariadením (napr. tlačiarňou) atď. Prepojením počítačov, prípadne s príslušenstvom, tak, že dokážu navzájom komunikovať, bezdrôtovým spôsobom alebo prostredníctvom fyzického prepojenia (kabeláže), vznikne **počítačová sieť**.

Vytvorenie počítačovej siete má svoje výhody, umožňuje predovšetkým:

- **zdieľanie<sup>1</sup> prostriedkov** (hardvéru), napríklad tlačiarňí, diskov
  - **zdieľanie dát** (softvéru) využívajú napríklad databázové a informačné systémy
  - **komunikáciu** medzi užívateľmi posielaním textových správ až po videokonferencie
  - **monitorovanie a riadenie počítačov, zdieľanie výkonu,...**
-



Ak má používateľ svoje programy a údaje uložené na serveri, je možné v prípade poruchy pracovnej stanice pokračovať v práci na inej pracovnej stanici

zdieľanie – sprístupnenie ostatným, spoluvlastníctvo

Zdieľanie je sprístupnenie objektu (priečinku, tlačiarne,...) na použitie v sieti.

#### 1. Zdieľanie diskového priestoru

Pre zdieľanie diskového priestoru môžu nastať tri dôvody:

- ako takého
- kvôli databázovým aplikáciám
- kvôli drahému programovému vybaveniu

#### 2. Zdieľanie databázovej aplikácie

- práca nad databázovou aplikáciou aj viacerých užívateľov súčasne
- jednoduchšie riadenie prístupu do databázy

#### 3. Zdieľanie programového vybavenia

- umiestnenie najčastejšie využívaného programového vybavenia na zdieľane disky

#### 4. Zdieľanie tlačiarní a iných periférnych zariadení

- využívanie donedávna veľmi drahých špeciálnych a veľmi kvalitných tlačiarní
- CD disky, scanery,

Základným parametrom siete je prenosová rýchlosť toku dát po sieti (b/s). Prenosové rýchlosti sietí LAN sa pohybujú **od 1 Mb/s až po 1 Gb/s**.

### **Ethernet:**

Pre potreby malých sietí sa používa s maximálnou prenosovou rýchlosťou 10 Mb/s až 100 Mb/s (v praxi iba 860 KB/s, pretože časť prenosu tvoria dáta prenosového protokolu a synchronizácia). Je rozšírenejšia, väčšie zaťaženie sietí, pôvodne používaná pre zbernicovú ale aj pre hviezdicovú topológiu.

Počítače v sieťach Ethernet komunikujú medzi sebou tak, že vysielajú na zbernicu a prijímajú dáta z nej môžu všetci súčasne, ale ak dvaja resp. viacerí naraz vyšlú údaje, nastáva kolízia, ktorá sa odstraňuje čakaním, po uplynutí ktorého sa vysielanie opakuje.

### **TOKEN RING:**

Počítače sú spojené do kruhu. Signál (Token) ich postupne prechádza všetky. Ktorý počítač má signál, môže vysielajú, ostane počítače môžu len počúvať. (Jedným smerom po kruhu ide tzv. pešiak, keď ho stanica dostane, môže odoslať správu). Týmto je odstránená možnosť vzniku kolízií pri súčasnom vysielaní niekoľkými stanicami, ako je to známe zo siete typu Ethernet. Dnes sa v týchto sieťach používa prenosová rýchlosť 16Mbit/sec. Nevýhodou tejto Topológie je horšia inštalácia siete a obmedzený počet

### **C. vlastnosti operačného systému**

Operačný systém je súhrnné označenie pre programy, ktoré riadia a dozerajú na spúšťanie a vykonávanie používateľských programov, pridelovanie prostriedkov jednotlivým programom (t.j. čas procesora, operačnú pamäť a prístup k V/V zariadeniam) a riadi tiež spôsob ukladania údajov na externé pamäte.

Operačný systém poskytuje tzv. používateľské rozhranie (grafické alebo textové), vďaka ktorému používatelia a programátori nemusia poznať podrobnú architektúru počítača. Ak napríklad používateľ chce vytlačiť dokument, nemusí poznať port rozhrania tlačiarne a jej pridelené prerušenie, nemusí vedieť na akej adrese v pamäti sa nachádzajú údaje, ktoré chce vytlačiť, všetky potrebné úkony zariadi operačný systém.

### **Vlastnosti moderných operačných systémov:**

viacúlohové - umožňujú zdanlivo vykonávať viacero aplikácií naraz

viacpoužívateľské - umožňujú pracovať viacerým používateľom súčasne  
distribúované - umožňujú rozdeliť spracovávanie úloh na viacero systémov  
Operačné systémy môžeme rozdeliť podľa niekoľkých kritérií:

#### Podľa určenia

- desktopové - určené pre stolné počítače v domácnosti
- serverové - určené pre počítače, ktoré poskytujú sieťové služby
- pre konkrétne zariadenie (netbooky, mobily, PDA)

#### Podľa dostupnosti

- Komerčné - Windows, MAC OS, UNIX
- Slobodné - Ubuntu, Debian, OpenSuse, Fedora, Mandriva, Sabayon, Slackware, Gentoo, Knoppix, FreeBSD, OpenSolaris, FreeDos ...

#### Podľa použitého jadra

- Windows
- Linux - Ubuntu, Debian, OpenSuse, Fedora, Mandriva, Sabayon, Slackware, Gentoo, Knoppix ...
- BSD - Mac OS, PC-BSD, FreeBSD, OpenBSD, NetBSD...
- Solaris - OpenSolaris, Oracle Solaris, FreeDOS

## 5.

**1.úloha: vysvetlite pojmy reálna premenná, aritmetické operácie s premennými**

**2.úloha: b. pojmy vlastnosti súboru, relatívna, absolútna cesta, asociácia prípony súboru**

Vlastnosti sú popisné informácie, ktoré sú užitočné pri vyhľadávaní a usporadúvaní súborov. Vlastnosti nie sú zahrnuté do skutočného obsahu súboru. Slúžia na poskytovanie informácií o súboroch. Okrem značiek, čo sú prispôbené vlastnosti, ktoré môžu obsahovať ľubovoľný vybraný text, môžu súbory obsahovať aj množstvo ďalších vlastností vrátane dátumu úpravy, autora a hodnotenia. Údaje ako texty, obrázky, tabuľky, programy atď. sa ukládajú vo forme **súborov**.

Každý súbor je pomenovaný: **názov.prípona**

**Názov** si volí užívateľ pri ukladaní - mal by charakterizovať obsah súboru. **Prípona (rozšírenie)** určuje typ súboru; je väčšinou 3-znaková a vo väčšine prípadov si ju určuje aplikácia (program), v ktorej súbor vznikol. Napr. **\*.EXE**, **\*.COM** sú spúšťacie súbory, **\*.DOC** je dokument Wordu, **\*.BMP** je obrázok atď. **Formát údajového súboru je uvádzaný vždy za názvom súboru a od názvu je formálne oddelený bodkou.** (napr.: „Textový súbor.doc“)

V systémovom prostredí Windows XP dokážeme rozlíšiť formát súboru dvoma základnými spôsobmi: podľa tvaru ikony, podľa prípony za názvom súboru

Ku každému súboru existuje **jednoznačná adresárová cesta** – vytvoríme ju zapísaním všetkých priečinkov od vrcholu hierarchického stromu až k priečinku v ktorom je súbor uložený napr: C:\ALFA\Data\ARCHIV\FA\_INGJO

Ak túto jednoznačnú adresárovú cestu zakončíme napísaním názvu súboru dostaneme **úplný názov súboru** napr. C:\ALFA\Data\ARCHIV\FA\_INGJO\jun.txt

#### NAJPOUŽÍVANEJŠIE FORMÁTY DÁT:

textové súbory - .doc, .txt, .rtf

súbory tabuľkového kalkulátora - .xls

počítačové prezentácie - .ppt

databázové - .mdb

obrázkové - .jpeg, .tiff, .raw, .bmp, .gif, .tga

zvukové - .wav, .wma, .mp3,

videosúbory - .mpeg, .avi

skomprimované súbory - .zip, .pdf  
dočasné súbory - .tmp

Viac tu: <http://melisko.webnode.sk/news/praca-so-subormi-a-priecinkami/>

### **c. popíšte SPAM a možnosti ochrany pred ním**

Spam je nevyžiadaná a hromadne rozosielaná správa prakticky rovnakého obsahu. Ide o zneužívanie elektronickej komunikácie, najmä e-mailu. Zväčša je používaný ako reklama, hoci za krátku históriu elektronickej komunikácie bol spam použitý aj z iných dôvodov. Existuje veľa rôznych médií, ktoré sú spamermi zneužívané. Môže to byť napríklad spomínaný e-mail, instantné zasielanie správ (napríklad ICQ), skupiny Usenet, krátke textové správy, ...

Rozlišujeme dva druhy spamu - prvý je poslanie správy do množstva diskusných skupín (tzv. fórum) Tieto správy sú väčšinou zamerané na distribúciu pornografického materiálu, resp. na propagáciu určitých produktov, napr. z oblasti farmácie. Správa väčšinou nemá žiaden súvis s témou diskusnej skupiny. Je zameraný na ľudí, ktorí často čítajú správy, ale neposkytujú svoje emailové adresy. Druhý typ je emailový spam, zaslaný konkrétnemu človeku priamo na jeho emailovú adresu. Tieto adresy sa získavajú často prehľadávaním diskusných skupín alebo prehľadávaním webových stránok. Emailové adresy získavajú internetové roboty prehľadávaním stránok. Ďalším spôsobom ako získavajú adresy je vyplnenie formulára ponúkajúceho na prvý pohľad zaujímavú protihodnotu. Existujú špecializované firmy alebo jedinci, ktorí vytvárajú databázy adries a tie následne predávajú.

Pretože množstvo diskusných skupín je dnes moderovaných a prevádzkovateľ limituje prevádzku, pokúšajú sa spammeri dostať cez automatizované nástroje do čo najväčšieho počtu takýchto diskusných skupín. Na zabránenie masového prístupu sa v súčasnosti používa systém CAPTCHA.

**Ochrana proti spamu** je komplikovaná. Problém je v tom, že neexistuje ostrá hranica medzi spamom a užitočnými emailami a neexistujú príznaky, ktoré by jednoznačne indikovali, že ide o spam. Následkom toho nie je možné vytvoriť prostriedok, ktorý by na sto percent likvidoval spam a na sto percent prepúšťal ostatné emaily. Existujú však celkom jednoduché opatrenia, ktoré umožnia minimalizovať množstvo spamov v emailovej schránke.

- **Čierna listina emailových adries** - Email, prichádzajúci z adresy, ktorá je na čiernej listine antispamový program neprepustí. Avšak tvorcovia spamov o možnosti vytvoriť čiernu listinu vedia a tak jednoducho menia adresy, z ktorých svoje spam odosielať.
- **Biele listina emailových adries** - Do bielej listiny sa zapisujú overené adresy. Emaily, prichádzajúce z adries z bielej listiny antispamový program prepustí, aj keby obsahovali „podozrivé“ slová. Jedinou nevýhodou tohto postupu je, že človek je nútený stále dopĺňať bielu listinu o nové adresy.
- **Zoznamy spamov** - Spam je celosvetovým problémom, a preto je samozrejmé, že sa na internetu dajú nájsť zoznamy adries, z ktorých sú posielané spamy. Potom je jednoduché porovnať adresu odosielať správy, ktorú práve posudzujeme s verejným zoznamom spamov a ak adresa patrí do tohto zoznamu, je prakticky isté, že ide o spam. Ako u všetkých zoznamov i tu však je problém s jeho aktualizáciou – na zozname sú predovšetkým staršie adresy, ale chýbajú tam tie najnovšie.
- **Pokročilé antispamové programy** - Primitívne antispamové programy hľadali v správe určitý reťazec znakov a keď ho našli, email prehlásili za spam. Filozofia týchto programov bola založená na predpoklade, že tvorcovia spamov používajú niektoré

frázy a slovné spojenia podľa ktorých sa dá usúdiť, že ide o spam. Ak sa však nadiktujú antispamovému programu určité podozrivé slová zadrží všetky emaily, v ktorých sa tieto slová objavajú.

Sofistikovanejšie antispamové programy využívajú pri rozhodovaní [štatistické](#) vyhodnocovanie. Zjednodušene povedané, vyhodnocujú výskyt „podozrivých“ slov v správe, napríklad nasledujúcim spôsobom:

- ak sú v predmete správy použité všetky veľké písmená pridaj 20 bodov,
- ak sa v správe vyskytuje skratka Dr. pridaj zakaždým 10 bodov,
- za každý vulgarizmus pridaj 40 bodov,
- ak správa obsahuje obrázok pridaj 30 bodov,

Rozhodovacie [kritérium](#) je potom už jednoduché: ak je súčet bodov väčší ako napr. 150, správa je klasifikovaná ako spam. Účinnosť tohto postupu je prekvapujúco dobrá a dokáže zachytiť väčšinu spamov bez toho, aby blokoval užitočné správy.

- **Obrana útokom** Spam síce vymýšľa človek, ale realizujú ho stroje. Teoreticky najúčinnjší spôsob ochrany voči spamu by bol taký, pri ktorom by sa po obdržaní správy žiadalo od odosielateľa vykonávanie určitej „inteligentnej“ činnosti, ako napr. poslať správu ešte raz a do predmetu napísať šifru A42. Ak nešlo o spam, partner na druhej strane takúto správu (ak je preňho dôležitá) pošle. Ak išlo o spam, počítač na druhej strane si nebude vedieť s požiadavkou poradiť. Samozrejme, keby sa tento postup začal uplatňovať masovo, jeho účinnosť by rapídne klesla, lebo ľudia by neboli ochotní robiť dvakrát to isté. Preto hovoríme o teoretickej možnosti, ktorú si v praxi môžu dovoliť len niektorí ľudia.

## 6.

### 1. úloha: vysvetlite pojmy premenná, konštanta, číselná premenná

### 2. úloha: b. konštrukčné zloženie operačných pamätí, prečo sa z nich stratí obsah po odpojení Napájania

Pamäte počítača

Pamäť je jednotka, do ktorej možno údaje vkladať, môžu sa v nej uchovávať a z ktorej možno vyberať čítať. Keďže počítače používajú dvojkovú sústavu, pamäte uchovávajú len 2 stavy – 1 a 0. Spôsob vkladania údajov do pamäte a druh záznamu závisí od technickej realizácie pamäte. Pamäte môžeme delíme podľa rôznych hľadísk.

Základné parametre pamäte:

#### 1. Kapacita pamäte –

vyjadrená je celkovým počtom bytov alebo slov, ktoré môžu byť súčasne uložené v pamäti. Zvyčajne sa vyjadruje v kilobytoch (KB) alebo megabytoch (MB). Najmenšou počítačovou informáciou je 1 bit = jednotka 1 alebo 0. Základnou jednotkou počítačovej pamäti je bajt (byte, skratka B), čo je 8 jednotiek, núl alebo ich kombinácií, napr. 00101110. Ide o pamäťovú jednotku, ktorá je schopná uschovať jeden znak (napr. písmeno). Maximálny rozsah pamäti sa vyjadruje v počte bajtov, ktorý je vždy mocninou dvojky. Používajú sa jednotky kilobajt (kB), megabajt (MB) a gigabajt (GB). Platí:

1

1 kB = 1024 B = 2<sup>10</sup> B

1 MB = 1 048 576 B

1 GB = 1 073 741 824 B

## 2. Rýchlosť pamäte

–je určená:

a)

vybavovacím časom –časový interval, ktorý uplynie od okamihu, keď je vyslaná požiadavka na prenos z pamäte, do okamihu, keď sa požadované údaje objavia na výstupe pamäte.

b)

cyklom pamäte –určený je minimálnym časovým intervalom medzi po sebe nasledujúcimi príkazmi k činnosti.

**3. Rýchlosť toku dát** – počet bitov, ktoré sa dajú za 1s do pamäte zapísať, alebo z nej čítať.

**4. Hustota záznamu** – udáva počet bytov n a jednotku dĺžky alebo plochy pamäťového prostriedku.

**Čo je to RAM?** Je to plošný spoj, na ktorom sú osadené IO (integrované obvody) resp. čipy na ukladanie dát. Veľkosť je rôzna od 80 x 30 mm (staršie) po 130 x 40 mm (novšie typy). Čipy na RAM sú rôzneho typu ako SAMSUNG®, HYNIX® atď. Na konci je slot na celú šírku plošného spoja, ktorý sa zasúva do slotov RAM na matičnej doske. Operačná pamäť slúži na rýchlejšie naťahovanie informácií a dát z pevného disku do procesora, čipovej súpravy, optických zariadení atď., a naopak. Je to akési urýchlenie celého systému kvôli tomu, že keď sa celé dáta len raz nahrajú do operačnej pamäte nemusia sa znovu nahrávať ako by to bolo pri pevnom disku. Operačná pamäť drží dáta, ktoré čerpá samotný systém. Možnosť veľkosti načítaných dát sa ohraničuje kapacitou operačnej pamäte. Kapacitu RAM určujeme v Bytoch, ale pre vyššiu kapacitu jednotka stúpa až na MB. Najznámejšie kapacity poznáme: **32, 64, 128, 256, 512, 1024 MB** moduly operačných pamätí. Tieto sa však dajú rôzne kombinovať podľa matičnej dosky. Existujú však aj 2048 MB moduly, ale tie sa až tak nepoužívajú kvôli ich vysokej cene a nízkemu rozšíreniu. Pri náročných operáciách a procesoch akými sú kódovanie videa, hry a podobne sa RAM pamäte zaberie veľmi veľa. Počítač, ak je brzdený tým, že mu chýba RAM pri náročných procesoch vytvára na pevnom disku tzv. **stránkovací súbor**. Pri stránkovaní sa počítač zachová tak, že vytvára virtuálnu operačnú pamäť na pevnom disku (čo je druhý najpomalší komponent) a tým výrazne stráca rýchlosť na rozdiel od pamäte RAM, ktorá je niekedy mnohonásobne rýchlejšia. Technicky vzaté ide o posúvanie a prípravu dát pre procesor, matičnú dosku... a súčasne platí, že to je zapisovacie a čítacie centrum. Výrobcov je veľa. Najznámejší sú: **Apacer, A-DATA, Samsung, Kingmax...** (priemerné RAM) a k vrcholu sa šplhá **Kingston**.

Typy operačných pamätí, ktoré sa používajú:

1. **SD-RAM** - už staršie menej používané, ale stále majú miesto v starších počítačoch.
2. **DD-RAM (DDR)** - prvá generácia,
2. **3. DD-RAM II. (DDRII)** - druhá generácia, najviac používané.
4. **DIMM / SIMM**
3. **Najnovšie DDR3,**

## c. príklady vplyvu informatizácie spoločnosti na jednotlivca

spoločnosť, ktorá používa moderné informačné a telekomunikačné technológie.

**klady:**

- Sprístupnenie informačných zdrojov a nástrojov pre ich využívanie širokou verejnosťou.
- Rozšírenie a skvalitnenie možností služieb a zábavy.
- Podpora vzdelávania.

- Nové možnosti pre uplatnenie tvorivých schopností ľudí, ako aj uplatnenie handicapovaných ľudí v živote cez „prácu na diaľku“ .
- Väčšie uplatnenie kultúrnych tradícií a identity regiónov.
- Výkonnejšia štátna správa.
- Efektívnejšie riadenie podnikov, zvýšenie konkurencieschopnosti, uľahčenie spojenia medzi výrobcom, poskytovateľom služby a zákazníkom samotným.
- Nové služby v rámci telekomunikácie a nové trhy v oblasti software.
- Efektívnejšia zdravotnícka starostlivosť.

### zápory:

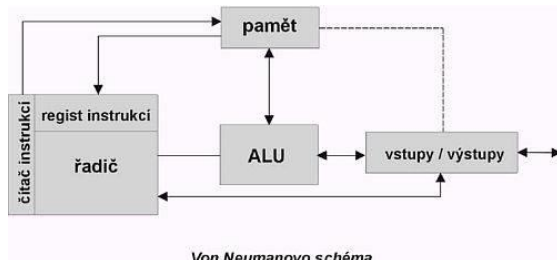
- V mnohých prípadoch nutnosť vysokých počiatočných investícií, či už v oblasti ľudských zdrojov alebo infraštruktúry.
- Kyberpriestor prináša so sebou nebezpečenstvo kyberterorizmu a kyberzločinov.
- Zmena životného štýlu človeka. V princípe, človek je čoraz viac preťažený informáciami a spoločnosť pôsobí uponáhľane, unavuje sa, extrémne množstvá informácií nedokáže spracovať

## 7.

### 1.úloha: vysvetlite pojmy cyklus, vnorený cyklus, vicnásobné vetvenie

### 2.úloha: b popis činnosti jednotlivých častí počítača von Neumanovo typu.

#### c. rozdiel medzi intranetom a extranetom



Von Neumanova schéma bola navrhnutá v roku 1945 americkým matematikom Johnom Von Neumannom ako model samočinného počítača. Tento model s istými výnimkami zostal zachovaný dodnes.

- Operačná pamäť: Služi k uchovaniu spracovávaného programu, dát a výsledkov výpočtov.
- ALU: Arithmetic-logic Unit /aritmeticko logická jednotka/: Jednotka prevádzajúca všetky aritmetické výpočty a logické operácie.
- Radič: Riadiaca jednotka, ktorá riadi činnosť všetkých častí počítača. Toto riadenie je prevádzané pomocou riadiacich signálov, ktoré sú zasielané jednotlivým modulom.
- Vstupné zariadenia: Určené pre vstup programu a dát.
- Výstupné zariadenia: Určené pre výstup výsledkov, ktoré program spracoval.

Von Neumannova koncepcia počítača

Táto koncepcia digitálneho počítača vznikla okolo roku 1946. Základné moduly ním navrhnutého počítača sú: procesor, radič, operačná pamäť, vstupné a výstupné zariadenie a to všetko tvorí základ architektúry súčasných počítačov.

Základné princípy

- dvojková sústava
- programy a dáta v operačnej pamäti (nenačitajú sa z vonkajšej pamäti v priebehu výpočtu, jednotné kódovanie - k programom môžeme pristupovať ako k dátam, to umožnilo univerzálnosť počítača, bezproblémové zavedenie cyklov a podmieneného vetvenia)
- rýchlosť vnútornej pamäti je porovnateľná s rýchlosťou aritmetickej jednotky

- priame adresovanie (prístup) - v ľubovoľnom okamžiku je prístupná ktorákoľvek bunka pamäti
- aritmeticko-logická jednotka - len obvody pre sčítanie čísiel (ostatné operácie sa dajú prevádzať na sčítanie)

Von Neumannov počítač

1946 John von Neumann

1. 5 funkčných jednotiek – riadiaca jednotka, aritmeticko-logická jednotka, pamäť, vstupné zariadenie, výstupné zariadenie
2. Štruktúra je nezávislá na spracovávaných problémoch, na riešení problému sa musí zvonka zaviesť návod na spracovanie (program) a musí sa uložiť do pamäte. Bez tohto programu nie je stroj schopný pracovať
3. Programy, dáta, medzivýsledky a konečné výsledky sa ukladajú do tej istej pamäti
4. Pamäť je rozdelená na rovnako veľké bunky, ktoré sú priebežne očíslované, cez číslo bunky (adresu) sa dá prečítať alebo zmeniť obsah bunky
5. Po sebe idúce inštrukcie programu sa uložia do pamäťových buniek idúcich po sebe, prístup k nasledujúcej inštrukcii sa uskutoční z riadiacej jednotky zvýšením inštrukčnej adresy o 1
6. Inštrukciami skoku sa dá odkloniť od spracovania inštrukcií v uloženom poradí
7. Existujú aspoň - aritmetické inštrukcie (sčítania, násobenia, ukladania konštánt,...), logické inštrukcie (porovnávanie, not, and, or,...), inštrukcie prenosu (z pamäti do riadiacej jednotky a na vstup/výstup), podmienené skoky a ostatné (posunutie, prerušenie, čakanie,...)
8. Všetky dáta (inštrukcie, adresy,...) sú binárne kódované, správne dekodovanie zabezpečuje vhodné logické obvody v riadiacej jednotke logická štruktúra

Princíp činnosti počítača podľa von Neumanna:

Do operačnej pamäti sa pomocou vstupných zariadení cez ALU umiestni program, ktorý bude vykonávať výpočet. Rovnakým spôsobom sa do operačnej pamäti umiestnia dáta, ktoré bude program spracovávať. Prebehne vlastný výpočet, pričom jednotlivé kroky vykonáva ALU. Táto jednotka je v priebehu výpočtu spolu s ostatnými modulmi riadená radičom počítača. Medzivýsledky sú ukladané do operačnej pamäte. Po skončení výpočtov sú výsledky poslané cez ALU na výstupné zariadenie.

Názorne si môžeme popísať Von Neumannovu schému. Predstavme si bežný počítač, aký máme bežne doma.

Každý počítač má matičnú dosku. Na tejto doske je sústredené všetko to, čo v počítači pracuje. Srdcom každého počítača je procesor. Ten vo Von Neumannovej schéme predstavuje ALU.

Ďalšou dôležitou časťou každého počítača je operačná pamäť. V bežnom počítači je to pamäť typu RAM (Random Access Memory).

Radič riadi chod jednotlivých častí počítača.

Vstupné zariadenia – sú to periférne (prídavné) zariadenia pomocou ktorých vkladáme informácie do počítača. Patrí sem napríklad klávesnica, myš, CD-ROM, DVD-ROM a skener. Vstupné zariadenia spracovávajú vstupné informácie zadávané človekom, prevádzajú ich do binárneho kódu a takéto informácie poskytujú programu na spracovanie.

Výstupné zariadenia – sú to periférne zariadenia pomocou ktorých nám vystupujú z počítača spracované informácie. Patria sem monitor, reproduktory, tlačiarne. Výstupné zariadenia spracovávajú už spracované informácie v binárnom kóde a prekladajú ich do výstupného formátu nám zrozumiteľného.

### **c. rozdiel medzi intranetom a extranetom**

#### **Intranet**

Intranet je súkromná počítačová sieť používajúca štandardy a protokoly Internetu (HTTP, FTP, SMTP). Prakticky ide o celofiremný web, na ktorom pracovníci nájdu všetko čo potrebujú pre produktívnu a efektívnu pracovnú činnosť.

Definícií intranetu je viac, tu sú 2 z nich:

1. Sieťové spojenie vybraných príbuzných/podobných klientov (computer) používajúcich štandardný internet protokol, t.j. TCP/IP a HTTP.

2. Sieť zložená z klientov a využívajúca IP adresing, za ktorou stojí jeden alebo viac firewallov zabezpečujúcich bezpečnosť tejto siete z vonku. Takouto sieťou môže byť interná podniková LAN alebo WAN sieť, čo znamená, že interná je táto sieť iba logicky, v skutočnosti môže fyzicky pokrývať celú zemeguľu. Prístup do tejto siete je však obmedzený skupinami klientov, ktorí sú do nej zahrnutí.

Intranet teda predstavuje tzv. "interný internet" s rovnakým transakčným protokolom HTTP (HyperText Transfer Protocol - slúži na distribúciu hypertextových dokumentov), založený na Client/Server architektúre (podobne ako iné TCP/IP aplikácie, napr. FTP, SMTP), pričom:

- Browser = Client
- Web server = Server
- CGI, API, JAVA = MiddleWare

Intranet podobne ako internet používa na základe TCP/IP SMTP (Simple Mail Transport Protocol) na posielanie pošty. Okrem toho však intranet používa tzv. MIME (Multipurpose Internet Mail

#### **Základné funkcie:**

Intranet by mal spĺňať päť základných funkcií:

- e-mail
  - o v rámci firmy existuje aj komunikácia pomocou elektronickej pošty
  - o vo väčšine prípadov sa používajú kontá, ktorými sa komunikuje aj mimo intranetu
  - o komunikácia vo vnútri musí byť navrhnutá tak, aby e-mail určený pre kolegu/kolegyňu neopustil intranet (môže obsahovať dôverné firemné údaje)
- zdieľanie súborov
  - o uloženie dôležitých firemných dokumentov na jednom mieste
  - o jednoduchý prístup zamestnancov k informáciám
  - o istota, že každý má aktuálnu verziu (smerníc, plánov práce,...)
- pracovné adresáre
  - o pokiaľ pracuje viacero ľudí na jednom projekte, úlohe, ...
  - o jednoduchá komunikácia
  - o vždy aktuálna verzia projektu
  - o umožnenie riadenia prístupu
- vyhľadávanie
  - o hľadanie určitej informácie v rámci intranetu je jednoduchšie a rýchlejšie ako prezeranie papierov a dokumentov na počítači (ktoré nemusia byť aktuálne)
  - o prístup k informáciám a kontaktom kolegov
- správa siete
  - o možnosť editácie informácií rýchlo a jednoducho

Okrem týchto základných funkcií sa vo väčšine intranetov nachádzajú aplikácie na evidenciu dochádzky, pracovných aktivít a iné. ež RFC - 1521), aby bolo možné využiť rozmanitejši a rôznorodejší obsah posielaných správ.

#### **Extranet**

Extranet je niekoľko intranetov, ktoré sú medzi sebou prepojené bezpečnými komunikačnými kanálmi technológiou "Virtuálnych privátnych sietí (VPN)". Extranet je koncept rozšírenej



podnikovej komunikácie, ktorá presahuje rámec hraníc podniku. Na vzájomnú komunikáciu medzi spolupracujúcimi podnikmi, ich partnermi, zákazníkmi, dodávateľmi a pod. slúži Internet a jeho komunikačné služby. Každý z účastníkov tvorí individuálny uzol na Internete, či už interný v rámci podnikovej siete, respektíve externý mimo rámca lokálnej siete. Na rozdiel od Internetu extranet nie je široko otvorený. V protiklade k intranetu extranet nie je zase obmedzený na interné použitie. Extranet svojou povahou umožňuje vytvárať virtuálne podnikové spoločnosti, ktoré nemajú charakter bežného podniku, ale sú to spúločenstvá vytvárajúce virtuálne pracovné zoskupenie, kde hlavnú úlohu a cenu má duševné vlastníctvo.

## 8.

**1.úloha: vysvetlite pojmy celočíselný datový typ, aritmetické operácie s celými číslami**

**2.úloha: b popíšte činnosť jednotlivých častí procesora**

Základom každého procesora je kremíková doštička. Na ňu sa špeciálnou technológiou nanášajú ostatné súčiastky, z ktorých sa procesor skladá. Túto doštičku - procesor umiestnime na platňu, na ktorú sa upevnia kontakty, ktorými bude procesor uchytенý v špeciálnej päťici a zároveň pomocou ktorých bude komunikovať s ďalšími komponentmi počítača. Procesor sa skladá z: **a) Jadro, b) Cache c) Radič operačnej pamäte d) GPU**

**Jadro(Core)** interpretuje a vykonáva inštrukcie obsiahnuté v softvéri a vykonáva výpočty. Je zložený z ALU, FPU a RJ. ALU – Arithmetic logic unit, je centrálna časť procesora, ktorá vykonáva základné a logické operácie s celými číslami ako napríklad sčítanie, odčítanie, násobenie, delenie, logický posun, negáciu a ďalšie. RJ – Riadiaca jednotka, je to časť CPU, ktorá ovláda všetky operácie a činnosť procesora. Riadi aj komunikáciu CPU s jeho okolím (RAM, chipset atď.) a zabezpečuje rozdeľovanie úloh celému CPU. PFU – Floating-point unit, je časť procesora, ktorá vykonáva operácie s pohyblivou desatinnou čiarkou. Procesor bez FPU prakticky nevie spracovávať čísla s pohyblivou desatinnou čiarkou (iné ako celé čísla). Dá sa to emulovať a počítať aj pomocou ALU, ale je to zbytočne niekoľko sto krát pomalšie, a vyťažuje to CPU. Bez FPU by sme si nezahrali žiadnu 3D hru. Každé jadro môže obsahovať ešte iné časti v závislosti od jeho určenia a generácie.

**Cache** – nazývaná aj ako vyrovnávacia pamäť, je dočasná veľmi rýchla aj keď malá pamäť procesora, ktorá zabezpečuje neprerušovaný prísun dát na spracovanie CPU a slúži tiež ako odkladací priestor medzi jednotlivými operáciami CPU. Je tvorená veľmi rýchlym typom pamätí – SRAM, ktorý je ale náročný na výrobu a aj na počet tranzistorov

procesora. Pre 1 bit informácie sa použije až šesť tranzistorov. Hlavným dôvodom existencie tejto pamäte je jej rýchlosť, keďže RAM počítača je pre potreby CPU priveľmi pomalá. Cache vyrovnáva časové rozdiely medzi spracovaním dát CPU a ich uložením, či načítaním v RAM. Cache sa v moderných procesoroch skladá zo štyroch úrovní.

L1 (level 1) je najrýchlejšia a najmenšia časť cache. Každé jadro procesora má vždy svoju vlastnú L1 cache. Je teda nezdieľaná.

L2 (level 2) je o niečo pomalšia ako L1, stále je ale veľmi rýchla. Táto úroveň cache sa už nedelí na dátovú a inštrukčnú, celá je dátová.

L3 Cache sa nepoužíva v bežných procesoroch. Používa sa najmä v serverových a 4 a viacjadrových procesoroch (Intel Xeon, AMD Phenom a Opteron).

V serveroch sa používa aj L4 cache, ale výroba procesora s veľkou L4 je cenovo aj technologicky náročná (väčšia náchylnosť na chyby vo väčšom kuse kremíku), a tak sú tieto procesory použité iba v špičkových serveroch, alebo superpočítačoch.

Radič operačnej pamäte integrovaný do CPU nie je bežný a je to vlastne integrovanie časti chipsetu kvôli zrýchleniu práce s RAM. Tento krok podstúpila firma AMD, spolu s prepracovaním komunikácie medzi časťami CPU, RAM a Chipsetom. Miesto FSB použila Hypertransport (HT), ktorý je rýchlejší. V budúcej generácii procesorov – Nehalem, integruje radič pamätí a použije rovnakú komunikáciu aj Intel. Keďže sa použil radič v CPU miesto v chipsete, tak sa nahradila komunikácia cez FSB (front side bus) CPU>Chipset>RAM>Chipset>CPU oveľa efektívnejšou a rýchlejšou komunikáciou cez HT CPU>RAM>CPU.

GPU - Procesor môže obsahovať aj grafické jadro. Takéto procesory sa nepoužívajú v klasických PC, ale vo veľmi lacných počítačoch a špecializovaných zariadeniach. V súčasnosti je taký produkt OLPC notebook pre rozvojové krajiny s procesorom AMD Geode. V budúcnosti sa budú ale používať aj v lacných stolných PC s nízkym výkonom – projekt AMD Fusion a Intel Lincroft, určených najmä pre kancelárske použitie

Procesor (CPU) je tzv. mozog počítača, ktorý vykonáva všetky aritmetické a logické operácie, zadané používateľom. Dáta programu načíta z operačnej pamäte, dekoduje ich a zložitejšiu operáciu rozloží na jednoduché inštrukcie a vypočíta ich. Táto časť počítača sa skladá z miliónov ( miliardy ) tranzistorov vyrobených pomocou nano technológií, ktorá napreduje neuveriteľným krokom.

Dôležitým parametrom procesora je **taktovacia frekvencia**, ktorá sa udáva v GHz(gigahertzoch), čím vyššia, tým je to samozrejme lepšie. Táto frekvencia sa dá dnes už samozrejme od rôznych výrobcou pretaktovať, čo znamená, že tí skúsenejší si doma o niekoľko percent dokážu zvýšiť tu frekvenciu, ale pri pretaktovaní musíte rátať s tým, že budete potrebovať oveľa lepšie chladenie, lebo čím viac ho pretaktujete, tým vyššia pracovná teplota jadra.

Ďalším dôležitým parametrom je **cache pamäť ( L2 prip. L3 )**, to je vyrovnávacia pamäť procesora, zabezpečuje plynulý chod medzi procesorom a zbernicou počítača. Taktiež platí, čím väčšia, tým lepšie. V súčasnej dobe sa jej veľkosť pohybuje od 512kB(intel Atom) a bežne ma hodnoty 3-6MB, špičkové modely od Intelu majú cache 10MB(Intel Core i7), súčasným maximom je až 30MB ( [Intel Xeon E7-4870](#) ).

Samozrejme existujú v dnešnej dobe aj viacjadrové procesory dvoj,štvor,šesť sú bežnou výbavou lepšieho počítača, alebo notebooku ba dokonca už aj osemjadrové a desaťjadrové procesory ( Intel Xeon E7-4870 ). To, že procesor ma viac jadier znamená, že to sú vlastne viaceré procesory uložené na jednom čipe(procesore). Čím viacej jadier má procesor, tým efektívnejšie a rýchlejšie pracuje.

Medzi jeden z parametrov patrí aj **Socket(pätica)** to je akoby nejaký držiak v ktorým je upevnený procesor na základnej doske. Sú rôzne sockety v závislosti od procesora napr. socketAm2, socket 957 atď. Ďalším z parametrov, ktorý je dôležitý najmä pri notebookoch je spotreba. O spotrebe hovorí maximálna hodnota **TDP(Thermal design power)**-konštrukčného tepelného výkonu. Hovorí o tom, koľko W tepla procesor vyprodukuje a teda je potrebné odvieť teplo chladiacim systémom, aby nedošlo k zbatečnému prehrievaniu. Procesory Intel Atom majú veľmi nízke hodnoty čo je len od 2 od 10 W, preto sú často pužívané v tzv. Netbookoch, výkonnejšie procesory okolo tých 30-50 W, a najvýkonnejšie procesory často sú nad 100W.

Rôzne typy procesorov sa líšia aj **výrobnou technológiou**, udávanou v nanometroch. Čím menšia hodnota tým sú menšie tranzistory na čipe. Do rovnako veľkého procesora sa ich zmestí viac, čo znamená vyšší výkon a navyše aj menšiu spotrebu. Súčasne procesory sú vyrobené 45nm alebo 32nm technológiou. Avšak firma Intel vyrába nové procesory už s 22nm techológiu označovanou ako Ivy Bridge. Okolo roku 2015 plánujú 14nm technológiu. Procesory sa tiež líšia šírkou registrov, pamätí, do ktorých sa ukladajú súčasne vykonávané informácie. Toto číslo znamená, koľko bitov spracováva procesor naraz, teda v jednom kroku.

V súčasnosti sú preferované **64-bitové** pred tými 32-bitovými. 64-bitové sú samozrejme rýchlejšie, naopak 32-bitové systémy znamenajú podporu len 32-bitových systémov, ktoré použijú operačné pamäte len do 3GB, čo je pre niektorých používateľov dosť veľký problém.

### **c. vysvetlite pojem INTERNET a princíp jeho fungovania**

Slovo internet je skratkou anglických slov **inter** a **network**. **Internet** je **celosvetový systém navzájom prepojených počítačových sietí**. **Internet** je **decentralizovaná sieť** – nemá žiadny **centrálny uzol, ani vlastníka, ani centrálného správcu**.

V roku **1983** sieť začala používať **komunikačný protokol TCP/IP**, ktorý tvorí **základ komunikácie** do dnes. V tom čase internet spájala iba pár tisíc počítačov. Hromadné rozšírenie prinieslo až uvedenie **operačného systému Windows 95**, ktorý umožnil ľuďom pomerne jednoduché ovládanie počítačov, **bez nutnosti programátorského vzdelania**. Od tohto času už rozvoj internetu nič nedokázalo zastaviť a **dnes je najväčšou celosvetovou sieťou**.

**Internet používa technológiu prepínania paketov** (údaje sa neprenášajú naraz, ale delia sa na menšie časti - pakety).

**Paket je blok používateľských dát spolu s potrebnými informáciami o adresách a administrácii, aby mohla sieť doručiť dáta do správneho cieľa.** Jedno dátové spojenie zvyčajne ponese tok (stream) paketov, ktoré nemusia byť zakaždým smerované tou istou cestou po fyzickej sieti.

### **IP adresa**

Každý počítač na svete má **IP adresu**. (verejnú alebo privátnu). **IP (verzia 4) adresa je 32 bitové číslo**. V **desiatkovej sústave** vyzerá napríklad takto: **192.168.1.1**. Počítač ho však **používa iba v dvojkovej sústave**, kde vyzerá napríklad takto: **11000000.10101000.00000001.00000001**.

**Najnižšia IP adresa je 0.0.0.0, najvyššia 255.255.255.255**. Tento systém umožňuje vytvoriť **4 294 967 296 adries**. Z kapacitných dôvodov sa v súčasnosti používa už aj **systém IPv6** (IP verzie 6, kde adresa bude **128 bitové číslo**). IPv6 umožní toľko adries, aby ešte dlho do budúcnosti mohlo mať IP adresu každé elektrické zariadenie na svete, teda aj chladnička, práčka, holiaci strojček, rýchlovarná kanvica. Systém má obrovskú rezervu preto, že aj verzia 4 vlastne už dávno svoje adresy vyčerpala.

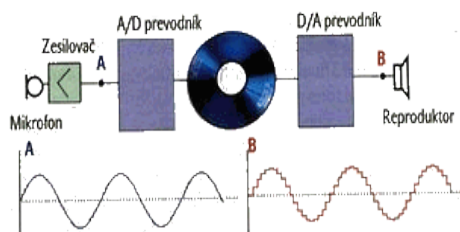
Časť počítačov **nepotrebuje byť napojená na internet priamo, môže teda mať privátnu adresu a komunikáciu s internetom zabezpečuje cez iné zariadenia**, napríklad **router, server, atď.**

## **9.**

**1. úloha: vysvetlite načo sú podmienky v príkaze cyklu, kedy ich vytvárame a ako ich progr. jazyk vyhodnocuje**

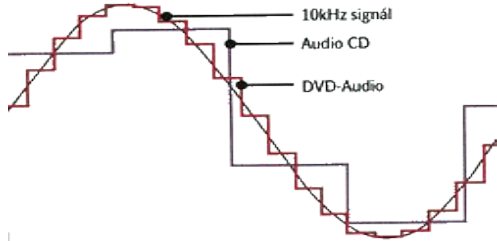
**2. úloha: a, analogová a digitálna zvuková informácia – rozdiel medzi nimi, popíšte proces digitalizácie zvuku**

Zvuk je pozdĺžne mechanické vlnenie s istou vlnovou dĺžkou a teda odpovedajúcou **frekvenciou**, s istou **farbou** a **intenzitou** (hlasitosťou). Zvuk je spojená - **analogová informácia**. Harmonický zvuk (napr. komorné "a") môžeme znázorniť sinusoidou. Počítače však vedú spracovávať len digitálne informácie.



Prevod zvuku z analogovej podoby do digitálnej zabezpečuje **A/D - analogovo - digitálny prevodník**. Najrozšírenejšia forma kódovania je **pulzná kódová modulácia - PCM**. Než sa z analogového signálu stane PCM zvuk, musí najprv prejsť **vzorkovaním, kvantovaním a kódovaním**.

**VZORKOVANIE:** Vzorkovač zaznamenáva aktuálne hodnoty analógového signálu v pravidelných intervaloch s istou frekvenciou napr. pri frekvencii 10kHz sa zaznamená hodnota signálu 10 000 krát za sekundu. Vyniká signál **PAM - pulzná amplitúdová modulácia**.

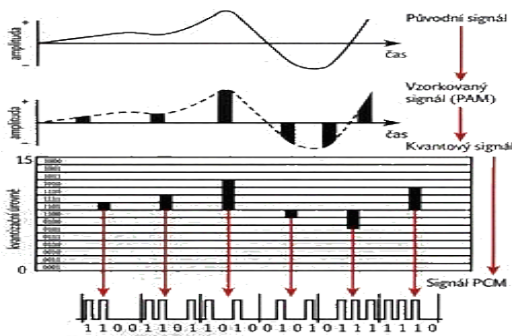


### Vzorkovacia frekvencia (Sampling rate)

Aby sa dal vzorkovaný signál PAM pri reprodukcii plne zrekonštruovať, musí byť splnené tzv. "**Nyquistovo kritérium**": frekvencia, ktorou sa vzorkovanie prevádza, musí byť aspoň 2-krát vyššia ako frekvencia pôvodného signálu. Ak je počuteľný zvuk od 16 - 20 000Hz, tak vzorkovacia frekvencia musí byť aspoň 40kHz. V praxi sa vzorkuje s 10% navýšením, preto sa používa vzorkovacia frekvencia 44,1kHz.

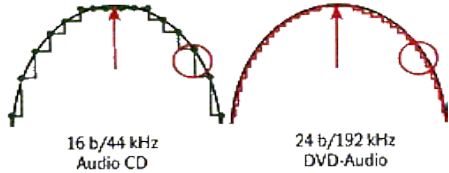
**KVANTOVANIE:** Kvantovaním sa namerané hodnoty "zaokrúhľujú" na najbližšiu úroveň amplitúdy každej vzorky, preto má digitálny signál na rozdiel od analógového schodovitý priebeh.

**KÓDOVANIE:** Pri kódovaní zvuku hudobného CD sa používa 16 bitové kódovanie - t.z že každú vzorku zakódujeme 16 - ticou jednotiek a núl - všetkých možných napät'ových úrovni signálu teda môže byť  $2^{16}=65536$  (hovor v telefóne je kódovaný 8 bitmi - rozlišuje 256 napät'ových úrovni). V prípade stereofónneho signálu sa používajú 2 kanály a výstupom sú 2 prúdy digitálnych hodnôt. Pre porovnanie uvádzame nasledujúcu tabuľku:



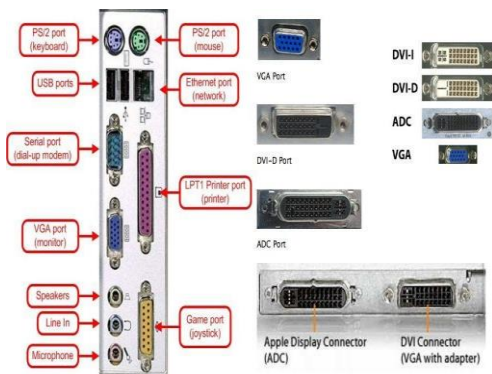
Kvalita digitálneho záznamu	vzorkovacia frekvencia	rozlišenie, kanály
Telefóna kvalita	11 025Hz	8 bit - mono
Rozhlasová kvalita	22 050Hz	8 bit - mono
CD kvalita	44 100Hz	16 bit - stereo
DVD kvalita	192 000Hz	24 bit - 5.1 surround sound

**PRÍKLAD:** Ak chceme zakódovať 1 minútu stereo nahrávky s použitím vzorkovacej frekvencie CD kvality, tak takýto PCM zvuk bude zaberat' 60x2x44100x16 bitov čo je asi 10MB pamäte.



Čím je vyššia vzorkovacia frekvencia, tým kvalitnejší zvuk získame. Doteraz najkvalitnejší zvuk v **CD kvalite** so vzorkovacou frekvenciou 44,1kHz, 16bit stereo je prekonaný formátom **DVD-Audio**, kde vzorkovacia frekvencia je až 192kHz s 24 bitovým rozlíšením.

**B, vstupné, výstupné, vstupnovýstupné zariadenia /uved'te aspoň tri/**



<b>V S T U P N É</b>	tláčidlové ovládače	- <b>klávesnica</b>
	grafické ovládače	- <b>myš</b> a jej modifikácie (trackball, touchpad) - <b>joystick</b> - pakový ovládač - <b>svetelné pero, dotyková obrazovka</b>
	grafické snímače	- <b>scanner</b> - snímač graf. predlohy - <b>tablet</b> (digitizér)  - <b>videokamera,</b> - <b>digitálny fotoaparát</b>
	snímače zvuku	- <b>mikrofón</b>
	sieťové a telekomunikačné dátové vstupy	- <b>modem</b>

<b>V Ý S T U P N É</b>	dočasné zobrazovanie informácií	- <b>CRT monitory</b> - <b>LCD, TFT displaye</b> - <b>LCD projekčné panely</b>
	permanentné zobrazovanie informácií	- <b>tlačiarne</b> ( ihličkové, tepelné, atramentové- tryskové, sublimačné, laserové, LED tlačiarne) - <b>súradnicové zapisovače (plottre)</b> s valcovým posunom, resp. pohyblivým mostom (kresliace stoly)
	počítačom riadené stroje	- <b>vyrezávacie plottre, roboty, NC frézy...</b>
	zvukový výstup	- <b>reprouktory</b> , hudobné a rečové syntetizátory,
	sieťové a telekomunikačné dátové výstupy	- <b>modem</b>

### **C, pojem interaktívna komunikácia** **INTERAKTÍVNA KOMUNIKÁCIA**

-slovo Interaktívny znamená „s možnosťou okamžitej reakcie“. Pri interaktívnej komunikácii je náš partner pripojený na Internet v tom istom čase ako my a môžeme s ním bezprostredne komunikovať, buď textovo, alebo hlasom, prípadne sa môžeme aj navzájom vidieť.

Interaktívnu komunikáciu prostredníctvom Internetu umožňujú špeciálne programy – textové telefóny (prenos textu) alebo video konferenčné programy (prenos zvuku a obrazu).

-jednoduchý nástroj na interaktívnu textovú komunikáciu – rozhovor cez Internet je Talk. Na nadviazanie rozhovoru treba poznať internetovú adresu partnera. Pri rozprávaní sa pomocou tejto služby je obrazovka (okno) rozdelená na dve časti. Do jednej sa zobrazuje to, čo píše náš vzdialený partner, do druhej to, čo píšeme my. Obaja pritom vidíme to isté. Je to ako textový telefón.

-nástroj na interaktívnu komunikáciu prostredníctvom Internetu, ktorý ponúka snáď najviac funkcií, je program ICQ (I seekyou). Tento program funguje na princípe klient – server. Ten, kto ho chce používať, si musí nainštalovať klientsky program a zaregistrovať sa na ICQ serveri. Potom si ho používatelia ICQ, ktorým to dovoľí, môžu zaradiť do svojho zoznamu s menami (prezývkami) vybraných používateľov ICQ a on si môže vytvoriť vlastný zoznam. Keď sa používateľ prihlási do ICQ, všetkým, ktorí ho majú vo svojom zozname sa znázorní jeho prítomnosť.

-ICQ je teda niečo viac ako iba nástroj na interaktívnu komunikáciu, obsahuje viac klientov –e-poštového, Talk, Finger, FTP.

## **10.**

**1.úloha: vysvetlite dokedy platí podmienka v cykle, a ako vplýva na počet opakovaní cyklu**

**2.úloha: b, aktívne, pasívne prvky v poč. sieti**

Aktívne prvky siete:

- sieťová karta
- modem (modulátor a demodulátor)
- Router (brána- zabezpečuje smerovanie, rozhranie medzi vnútornou sieťou a Internetom)
- Hub (rozbočovač – umožňuje pripojenie viacerých počítačov, delí signál medzi pripojené PC)
- Switch (prepínač - umožňuje pripojenie viacerých počítačov, prepína prístup)
- Repeater (opakovač – obnovuje signál, zaraďujeme ho do siete LAN, ak dĺžka kábla presahuje 100 m)

### Pasívne prvky siete:

- káble (medené, optické)
- konektory (BNC – I, T; RJ 45)
- zakončovacie odpory

## **C, význam antivírusovej ochrany v sieťach**

Komplexnú antivírusovú ochranu LAN siete pripojenej k Internetu môžeme rozdeliť na tri základné úrovne:

- 1. Antivírusová ochrana staníc**
- 2. Antivírusová ochrana groupwarových a súborových serverov**
- 3. Antivírusová ochrana vstupných brán do vnútornej siete**

Aktívna ochrana údajov

- Firewall - úlohou firewallu je ochrana počítača pred nežiadúcimi útokmi alebo komunikáciou (hlavne z Internetu), ktorých úlohou je poškodiť alebo zmazať údaje. Firewall chráni používateľa pred útokmi prostredníctvom počítačovej siete. Jeho činnosť sa prejavuje hlásením požiadaviek na prístup z určitej lokality a užívateľ môže prístup povoliť alebo zamietnuť. Každý firewall má svoje nastavenia, kde sa určuje úroveň zabezpečenia a kontroly systému.
- Antivírus a antispyware - úlohou oboch je chrániť pred škodlivými vplyvmi iných programov a aplikácií, ktoré sa môžu nachádzať v e-mailoch, dokumentoch, aplikáciách, hrách, ... Prvotnou úlohou používateľa je používať aplikácie a súbory z overených zdrojov (stránky výrobcov, originálne balenia, ...).

- úlohy antivírusových programov:

- detekcia,
- odstránenie,
- monitoring operačného systému a správania sa spustených aplikácií.
- pozri kapitolu Vírusy

- úlohy antispyware:

čo je spyware: reklamné okná, zbiera osobné údaje, heslá, identifikačné údaje, navštívené webové stránky, mení nastavenie aplikácií alebo systému,...

- detekuje spyware,
- bráni jeho inštalácii,
- odstránenie.

- Update operačného systému a aplikácií - pravidelným update sa dá predísť poškodeniu operačného systému, aplikácií alebo údajov. Prostredníctvom update získa užívateľ najnovšie opravné balíčky, ktoré riešia nedostatky existujúcich verzií. Update sa najčastejšie vzťahuje na operačný systém, internetové prehliadače, aplikácie elektronickej pošty, antivírusové a antispamové aplikácie.

Predchádzanie a prevencia je najúčinnější zbraň v súčasnom svete internetu a počítačových sietí.



Pasívna ochrana údajov

Archivácia operačného systému a údajov:

- uloženie údajov a ich opätovné použitie
  - ukladáme na médium, ktoré nie je súčasťou počítača - napr. disk CD alebo DVD, na kartu USB flash alebo do iného počítača, ak je náš počítač pripojený k sieti,
  - pre získanie ešte väčšieho priestoru, je možné údaje pred uložením skomprimovať.

## 11.

**1.úloha: vysvetlite syntax neúplného podmieneného príkazu, demonštrujte použitie tohto cyklu v bežnej hovorovej reči**

**2.úloha:**

**B, Porovnajzte základné parametre používaných druhov pamätí a stručne ich charakterizujte (kapacita, prístupová doba, prenosová rýchlosť, možnosť zápisu a princíp záznamu).**

Pamäť – slúži na uchovávanie rôznych druhov údajov

vnútorná - RAM (Random Acces Memory) - slúži na uchovávanie údajov a programov počas činnosti počítača. Z RAM pamäti je možné informácie nielen čítať, ale aj zapisovať ich do nej. Je závislá na dodávke elektrického prúdu, po vypnutí počítača sa údaje z nej strácajú. Nazýva sa aj operačná pamäť. V súčasnosti sa za minimum považuje 256 MB, odporúča sa 1 GB a viac.

- ROM (Read Only Memory) – slúži na uchovanie rôznych obslužných programov, ktoré nie sú určené na prepisovanie, napr. BIOS na základnej doske či grafickej karte apod. Dá sa prepísať iba pomocou špeciálnych programov. Kapacita týchto pamätí býva malá, najviac niekoľko MB.

vonkajšia - pevný disk – pamäťové médium, zložené z magnetických kotúčov uložených v hermetickom obale spolu s hlavičkami, ktoré sa používajú na zápis a čítanie. Medzi najdôležitejšie parametre patrí prenosová rýchlosť disku, t.j. koľko MB dát prenesie za sekundu, a prístupová doba, t.j. za akú dobu nájde údaje, ktoré sú v ňom zapísané. Kapacita dnešných diskov sa pohybuje na úrovni 100 – 500 GB.

**C, Vysvetlite výhody a nevýhody elektronického obchodovania.**

Výhody pre zákazníkov

V súčasnosti spotrebiteľ môže nakupovať pohodlne zo svojho domova, práce, jednoducho povedané, všade tam, kde má pripojenie na internet. Nepotrebuje ísť osobne do „kamennej predajne“ v dobe otváracích hodín a komunikovať s predavačom, pričom musí počítať aj s pravdepodobnosťou, že hľadaný tovar alebo služba nebude dostupná. Naopak pri elektronickom obchode si pohodlne zákazník nakupuje neobmedzenom čase tovar nachádzajúci sa momentálne na ktoromkoľvek mieste na svete. Ak nenájde hľadaný tovar alebo službu, prejde na iný elektronický obchod bez nulovej fyzickej námahy. Tento druh nakupovania šetrí čas, ktorý by sme inakšie pri prechádzaní z jednej „kamennej predajne“ do druhej míňali.

Elektronický obchod ponúka na výber širokú škálu tovarov a služieb rozdelených do jednotlivých skupín a podskupín. Pre ľahší výber má k dispozícii zákazník vo väčšine prípadov katalóg s produktami, kde sa dozvie základné informácie o produkte. Pri názve produktu sa môže nachádzať obrázok, video alebo vizualizácia približujúca daný tovar. Prechádzaním medzi elektronickými obchodmi získa rozhľad v ponuke, kvalite produktov. Nakoľko sa jedná o obchod, spotrebiteľ potrebuje mať k dispozícii aj cenník. Porovnávaním cien tovarov a služieb v jednotlivých e-shopoch spotrebiteľ dokáže ušetriť značnú časť peňazí. K tejto cene si musí pripočítať aj náklady spojené s nákupom a dodaním produktu a zároveň odpočítať zľavu, ktorú mu môže poskytnúť spoločnosť pri množstevnom nákupe alebo nákupe nad určitú sumu. Ak prevyšuje výsledná cena cenu produktu v „kamennom obchode“, neoplatí sa realizovať obchod.

### 2.1.2 Výhody pre podniky

Nadšenie pre obchodovanie cez pripojenie na internet uvítali aj spoločnosti a to naraz z viacerých dôvodov. Hlavnými preferenciami bolo rozšírenie svojej pôsobnosti za okraj svojich trhov na neobmedzený rozsah z globálneho pohľadu. Za druhý hlavný dôvod sa považuje značné zníženie nákladov, čo sa týka priestorov a zamestnancov. Spoločnosti využívajúce predaj cez katalógy zásielkového obchodu prešli na online katalógy, čím sa im tiež prudko znížili náklady. Technologizáciou obchodu sa zjednodušili administratívne činnosti. Podnikom klesli náklady na pracovnú silu, ktorá mala evidovať a spracovávať objednávky a vykonávať fakturáciu. Namiesto toho sa používa automatizované sledovanie objednávok, ako aj automatizovaná fakturácia.

Pomocou elektronického obchodovania môžu podnikat' aj fyzické osoby, ktoré by sa za normálnych podmienok nemohli pracovať v tradičnom obchode, nakoľko tento druh obchodovania nevyžaduje fyzickú účasť predajcu. Nemusí sa jednať len o predaj vyrobených tovarov a služieb, ale pod elektronickým obchodom sa rozumie aj nákup produktu z iného e-shopu a následná predaj spotrebiteľovi. Z tohto vyplýva, že môžeme obchodovať s tovarom a službami, ktoré fyzicky podnik nemusí mať k dispozícii. Alebo elektronický obchod má sklady na miestach, kde eviduje najmenšie náklady a z týchto skladov exportuje tovar k spotrebiteľom.

Pri vykonaní transakcie, každý zákazník poskytne spoločnosti o sebe inforácie, ktoré podnik môže využiť na sprofilovanie zákazníkov. Služí to na kategorizáciu spotrebiteľských skupín, čo následne

spoločnosť použije pri zameraní reklamy, výbere skladov, obchodných partnerov a podobne. Rovnako môže sledovať aj históriu prechádzaných produktov, z čoho zistí záujmy a preferencie zákazníkov. Spoločnosť si takýmto jednoduchým spôsobom spraví prehľad o potrebách spotrebiteľov a prispôbi im svoju ponuku.

### 2.2.1 Nevýhody pre zákazníkov

Niektoré druhy tovarov a služieb predávané na internete nemajú veľkú obľubu medzi spotrebiteľmi. Väčšinou váhanie je spôsobené z nedôvery voči kvalite produktu. Napríklad zákazníci preferujú skôr nákup nábytku v kamennej predajni. Pretože takýto druh tovarov je drahý, spotrebiteľ si ho chce osobne prezrieť pred kúpou. Silno pôsobí na spotrebiteľskú mienku aj ich okolie. Ak niekto zo známych mal zlú skúsenosť, v spotrebiteľovi sa vytvorí strach pred elektronickým nakupovaním. Značnou nevýhodou môže byť aj sociálny prístup k nakupovaniu. Môžeme si pod tým predstaviť preferovanie obľúbenej kamennej predajne alebo ak máme obľúbeného zamestnanca obchodu. Niektorí zákazníci spájajú nakupovanie s prechádzkami so svojimi priateľmi, čo elektronické obchodovanie v súčasnosti rozhodne neumožňuje. Pri elektronickom obchodovaní neexistuje plná anonymita. Pri nákupe sa musí spotrebiteľ zaregistrovať, kde udáva svoje osobné údaje. Tu vniká riziko nerešpektovania súkromia, kedy si spoločnosti navzájom vymieňajú alebo predávajú databázy kontaktov a informácií o spotrebiteľoch. Zákazníci s nulovou alebo minimálnou počítačovou gramotnosťou nemajú potrebné vedomosti na tento druh obchodovania. Pre nákup potrebujú mať pri sebe skúsenú osobu, čo pôsobí dosť obmedzujúco na výber produktov.

Značnou nevýhodou pre spotrebiteľov je používanie kreditných kariet, pri ktorých použitie je spravidla spojené s nižšími nákupnými nákladmi. S týmto súvisí aj riziko obchodné. Spotrebiteľ pošle spoločnosti peniaze, ale tovar mu nebude dodaný. Často sa vyskytuje výmena vybraného tovaru za iný tovar, alebo za poškodený tovar. Čo sa týka reklamácií a zákaznickeho servisu, spotrebiteľ sa veľmi na ne nespolieha. Zložitý proces reklamácií so zahraničnými ale aj domácimi spoločnosťami odradí zákazníkov.

### 2.2.2 Nevýhody pre podniky

V súvislosti s ochranou osobných údajov spotrebiteľov a ochranou interných údajov spoločnosti musí podnik vynaložiť najväčšie úsilie, ktoré ho stojí nemalé peňažné sumy. V poslednom období sa zvýšil nárast hackerských nabúrání systémov a databáz, čím spoločnosti strácajú doveryhodnosť a aj zákazníkov. Taktiež sa nemôžu spoliehať, že sa im viackrát nedostanú cez zabezpečenie. Aby spoločnosť mohla obchodovať na internete, potrebuje mať minimálne jedného schopného zamestnanca na správu e-shopu. Sú tu pomerne veľké vstupné náklady na začkolenie zamestnanca, nákup počítačov, operačných systémov, vytvorenie e-shopu a podobne. Zloženie elektronického obchodu vyžaduje jeho rozsiahlu reklamu a propagáciu. Spoločnosť musí využívať čo najviac platobných nástrojov vhodných pre spotrebiteľov. Tiež musí mať vymedzené peňažné prostriedky na nákup tovarov do skladu spoločnosti. Keďže spoločnosť pôsobí v oblasti elektronického obchodovania cez internet, bojuje s vysokou globálnou konkurenciou. Preto musia vyvíjať veľké úsilie na držanie rendu s ponukou produktov, cenami, kvalitou, zľavami, darčekom a podobne.

## 12.

### Úloha č.1: Podmienený príkaz – úplný

Vysvetlite syntax úplného podmieneného príkazu. Uvedením vhodných viet demonštrujte použitie tohto príkazu v bežnej hovorovej reči.

**B, Popíšte činnosť a ukážte na demonštračnom počítači základné komponenty osobného počítača.**

## C, Uved'te rozdiel medzi prepínačom a smerovačom a popíšte ich funkcie.

### Prepínač,(switch)

Vysokorýchlostný multiportový most, ktorého výhodou oproti rozbočovaču je v spôsobe, akým posiela rámce k cieľovému počítaču. Každý dátový rámec je určený pre konkrétny počítač v sieti a keďže rozbočovač nevie na ktorý port má daný dátový rámec poslať, rozošle ho na všetky porty, čím zbytočne zaťažuje komunikáciu v sieti. Prepínač obsahuje vnútornú pamäť, v ktorej si uchováva všetky sieťové adresy MAC pripojených počítačov. Podľa hardvérovej adresy prepínač vie, na ktorý port má vyslať dáta.

### Smerovač,(router)

Zatiaľ čo rozbočovač a prepínač pracujú s dátovými rámcami, router spracováva dátové pakety. Paket neobsahuje len dáta, ale aj cieľovú adresu, na ktorú má byť doručený. Router väčšinou prepája dve alebo viac sietí (LAN, WAN a podobne). Pomocou hlavičky paketu a „forwardovacej tabuľky“ dokáže router určiť najlepšiu cestu pre jeho doručenie. Na komunikáciu medzi dvoma smerovačmi sa používa ICMP protokol (Internet Control Message Protocol).

### Smerovací prepínač (routing,switch)

Pracuje s informáciami tretej vrstvy pri rýchlostiach obvyklých pre druhú vrstvu. Nahrádza pomalšie smerovače pri oddeľovaní broadcastových domén. Takzvaný Layer 3 switching je v podstate obdoba prepínania na druhej vrstve. Tu je však prepínanie na základe tabuľky MAC adries. Prepínanie na tretej vrstve je tiež riešené hardvérovo a rozhodovacie algoritmy sú rozšírené o tabuľku logických adries. Smerovací prepínač tiež umožňuje komunikáciu so štandardnými smerovacími protokolmi.

## 13.

### Úloha č.1: Základy – logické operácie

Vysvetlite pojmy: logická premenná, logické operácie, zložená podmienka

Úloha č.2:

A, Vysvetlite pojmy: www stránka, odkaz, hypertextový dokument.

**Web page** (do slovenčiny prekladané ako **webová stránka**, ľudovo **webstránka/web stránka**, nepresne **internetová stránka**) je dokument obsahujúci hypertext, obrázky a iné multimediálne prvky uložený obyčajne na webovom serveri prístupný prostredníctvom služby World Wide Web v sieti internet. Zobrazuje sa používateľovi pomocou webového prehliadača. Webové stránky sú obyčajne písané v značkovom jazyku HTML alebo XHTML a prenos prebieha pomocou protokolu HTTP. Spočiatku boli webové stránky písané čisto ako dokumenty v HTML (tzv. statické stránky). Postupom času sa začali používať aj technológie na generovanie dynamického obsahu prostredníctvom skriptovacích jazykov (napr. ASP alebo PHP), ktoré inštruujú webový server ako zostrojiť webovú stránku. Ako zdroj dát môže slúžiť  databáza. V súčasnosti sa tešia veľkej obľube rôzne redakčné systémy (CMS).

## Odkazy na zdroje v elektronickej forme

Pri využívaní informácií z internetu a elektronickej dokumentov z iných zdrojov treba dodržiavať tieto zásady:

- uprednostňujeme autorizované súbory solídnych služieb a systémov,
- zaznamenáme dostatok informácií o súbore tak, aby bolo opäť možné vyhľadať informácie,
- urobíme si kópiu použitého prameňa v elektronickej alebo papierovej forme,
- za verifikovateľnosť informácií zodpovedá autor, ktorý sa na ne odvoláva.
- **Odkaz na informáciu z www:**
- Treba zaznamenať meno autora a názov dokumentu, na ktorý sa autor v písomnej práci odvoláva, a potom lokáciu URL.
- *Príklad:*  
Boutell, Thomas. 1996. What can I do to protect my legal rights on the web?  
<http://nswt.tuwien.ac.at:8000/htdocs/boutell/>

**Hypertext** je z hľadiska [hypertextológie](#) konečný tvar činnosti autora pri tvorbe [hyperdokumentu](#). Hypertext je nelineárne (nesekvenčné) usporiadanie [textu](#), v ktorom sú jednotlivé časti navzájom prepojené [hypertextovými väzbami](#) reprezentujúcimi najčastejšie [krížové odkazy](#).<sup>[1]</sup> Hypertext je:

- Text písaný a čítaný nesequenčným spôsobom. <sup>[2]</sup>
- Spôsob reprezentácie a správy nepravidelne štruktúrovaných informácií, ktoré sú organizované podľa významu do modulov. V jednom module býva obyčajne tlmočená ucelená myšlienka, vysvetlený určitý pojem, prípadne vyjadrená hierarchia pojmov. Moduly sú prepojené spojeniami, takže vytvárajú sieť. ([Miloš Kravčík](#))
- Okná, zobrazené na monitore sú prepojené s objektmi v databáze a sú vytvorené linky medzi týmito objektmi.
- Vo všeobecnosti hypertextový systém má tri úrovne:
  1. databázu informácií,
  2. hypertextový abstraktný stroj (HAM)
  3. prezentačná úroveň. Aplikované na [Internet](#) to znamená, že www je hypertextom, obsahy dostupné cez Internet vytvárajú webovú databázu, HAM v súčasnosti predstavuje [HTML](#), [HTTP](#), a [FTP](#), prezentačnú úroveň najčastejšie reprezentuje [webový prehliadač](#)

**V adresári *Stránka1* sa nachádzajú štyri webové stránky. Hlavná stránka *index.html* a tri podstránky *sucet.html*, *sucin.html* a *negacia.html*. Prepojte hlavnú stránku s podstránkami pomocou odkazov.**

**B, Vysvetlite pojem informačná spoločnosť a vymenujte jej základné znaky.**

**INFORMAČNÁ SPOLOČNOSŤ** - je taká spoločnosť, kde práca s informáciami je každodennou záležitosťou. Žijeme uprostred informácií. Všade okolo nás sú zariadenia, ktoré nám ich pomáhajú prenášať, uchovávať, spracúvať. Pre jednoduchosť im hovoríme počítače, hoci mnohé z nich vyzerajú na prvý pohľad úplne inak, napr. elektronickej

zápisníky, mobily a pod. Zariadenia na spracovanie informácií sa dnes ukrývajú aj v televízore, aute, práčke, alebo fotoaparáte

→ **informačná éra** – rozmach počítačov, počítačových sietí a komunikácie (od 20.storočia)

Medzi **základné znaky** informačnej spoločnosti patria :

- **Informácia je najcennejším tovarom** – ten , kto získa novšie informácie a vie ich efektívne využiť, môže získať ekonomický náskok, predvídať budúcnosť, vyvíjať nové technológie.
- **Informatizácia výrobných procesov** – odbremeňuje človeka od monotónnej fyzickej práce prostredníctvom rýchlejších, výkonnejších a spoľahlivejších elektrických zariadení.
- **Informačný sektor zamestnáva viac pracovníkov.**
- V oblasti hospodárstva sa **rozvíja oblasť služieb**. Významným obchodným artiklom sú informácie, ktoré je možné prostredníctvom počítačových sietí prístupovať a predávať okamžite a to na ktoromkoľvek mieste na svete.

**INFORMATIZÁCIA** - proces zberu, spracovania, prenosu a uchovania informácií.

**INFORMAČNÉ A KOMUNIKAČNÉ TECHNOLOGIE** – pod týmto pojmom rozumieme hardware, software a komunikačné technológie, pomocou ktorých dochádza k prenosu dát medzi počítačmi, ktoré sú navzájom poprepájané v sieti. Žijeme v informačnej spoločnosti, využívame výhody informačných technológií, ktoré však so sebou prinášajú aj rôzne riziká a nedostatky.

→

**C, Popíšte nové formy komunikácie (sociálne siete, blogy, diskusné fóra, webové konferencie, ..), ich výhody a nevýhody.**

**14.**

## **Úloha č.1: Cyklus - cyklus s pevným počtom opakovaní**

**Vysvetlite pojmy: cyklus s pevným počtom opakovaní, riadiaca premenná cyklu**  
**Príklad**

Napíšte program, ktorý vytvorí pravouhlý trojuholník v jednoduchom číselnom rade tak, aby vrchol začínal číslom „1“. Výšku trojuholníka (počet riadkov) zadáte na vstupe z klávesnice do premennej V. Výsledok vyzerá napr. takto (pre V=6):

```
1
2 3
4 5 6
```

7 8 9 10  
11 12 13 14 15  
16 17 18 19 20 21

Úloha č.2:

**a, V adresári Stránka2 sa nachádza súbor *druhy\_regulacie.html*. Upravte súbor podľa obrazovej prílohy.**

**Úlohy:**

- **Pod nadpis *Druhy regulácie* vložte čiaru.**
- **Vytvorte pre delenie druhov regulácie (podľa toho, kto vykonáva funkciu regulátora a podľa druhu signálu) viacúrovňové odrážky.**
- **Vložte do súboru obrázok, ktorý sa nachádza v adresári *pics*.**
- 

**B, Vysvetlite pojmy: súborový systém, fyzický a logický disk. Aké súborové systémy možno použiť na optických diskoch, v operačných systémoch Windows a Linux.**

Súborový systém vytvára hierarchickú štruktúru zväzkov a zložiek, do ktorých sú ukladané jednotlivé súbory. Súčasne súborový systém usporiada dáta tak, aby ich bol schopný vyhľadať na ktorejkoľvek časti disku. Súborový systém je zvyčajne integrovanou súčasťou operačného systému, pričom mnohé najnovšie operačné systémy podporujú niekoľko rôznych súborových systémov.

#### 2.1 Štruktúra súborového systému FAT

File Allocation Table (FAT, alokačná tabuľka súborov) je podľa [2] špeciálna oblasť na sformátovanom údajovom médiu, obsahuje informácie o uložených súboroch a jeho adresárovej štruktúre. FAT popisuje aj kompletný popis rozloženia súborov na médiu. Alokačná tabuľka nie je voľne prístupná používateľovi a upravuje sa automaticky pri manipuláciách so súborami a adresármi na disku. Z dôvodov bezpečnosti sú na väčšine médií vytvárané a udržiavané dva rovnaké kópie tej istej tabuľky. Jednotlivé varianty FAT sa líšia najmä počtom čísiel, ktoré je možné použiť k vyjadreniu čísla clusteru v alokačnej tabuľke.

FAT12 – pri logických jednotkách s kapacitou do 16 MB (diskety)

FAT16 – pri logických jednotkách s kapacitou od 16 MB do 2 GB (OS Windows NT/2000/XP umožňujú použiť tento variant pre logické jednotky až do 4 GB) Obmedzenia

FAT16:

#### Súborový systém Windows NT (NTFS)

Podľa [1] ho používajú OS Windows NT/2000/XP. Meno súboru môže byť dlhé maximálne 256 znakov a maximálna veľkosť diskového oddielu je obmedzená na 16 exabytov. Súčasťou súborového systému NTFS sú rozšírené atribúty súborov a ich zabezpečenie, neexistujúce v systémoch FAT a FAT32. Podporuje väčšie súbory a oddiely, ponúka vyšší výkon, stabilitu a bezpečnosť, ale nie je spätne kompatibilný.

Systémové súbory súborového systému NTFS:

- *hlavná tabuľka súborov* (MFT) – obsahuje záznam pre každý súbor
- *druhá kópia MFT*
- *súbor so zoznamom chybných clusterov*
- *raster alokácie clusterov* – ukazuje využitie jednotlivých clusterov

- *spúšťačí súbor* – ak ide o aktívny disk, obsahuje spúšťačí kód

7/7

- *tabuľka definície atribútov* – obsahuje definíciu všetkých systémových a používateľsky definovaných atribútov, použitých na logickej jednotke
- *súbor protokolov* – zaznamenáva jednotlivé operácie so súbormi (využíva sa pri obnove údajov)
- *tabuľka diskových kvót* – tabuľka v ktorej je uvedená kvóta diskového priestoru každého používateľa (NTFS 5)
- *tabuľka pre prevod malých znakov na veľké*
- *zväzok* – obsahuje základné informácie o logickej jednotke

Poznámka: 1 exabyte =  $2^{60}$  bytov = 1024 petabajtov. Údajová kapacita zatiaľ jednotlivým pamäťovými médiami nedosiahnuteľná, nič menej sa s ňou už občas stretneme [

ako najviac rozšírené úložisko údajov používajú **pevné disky**. Jedná sa o **mechanické zariadenie**, kde sú na jednej pevnej osi namontované viaceré kovové platne. Pomedzi ne sa pohybujú čítaco – **záznamové hlavy**. Tieto sú schopné zmagnetizovať povrch disku a tak zaznamenať naň údaje. V jednom počítači býva zväčša namontovaný aspoň jeden fyzický pevný disk o určitej kapacite a rýchlosti otáčania. Na každom disku je výrobcom dodaný štítok, ktorý obsahuje základné informácie o disku, ako sú:

- Počet čítacov – záznamových hláv
- Počet cylindrov
- Počet sektorov

Mnoho užívateľov má jeden **fyzický disk**, ktorý je **zároveň aj logickým diskom** (zvyčajne sa označuje **c:\**). Sú však aj takí užívatelia, ktorí chcú mať svoj disk rozdelený na viacej logických častí. Potom sa môžeme stretnúť s tým, že sa nám v počítači zobrazujú viaceré diskové jednotky ako: **c:\**, **d:\**, **e:\** atď.

Pri inštalácii operačného systému sa nás inštalátor pýta či si prajeme rozdeliť disk. Ak áno vytvorí sa viacej **partícií** – presne toľko, koľko chceme mať vytvorených diskov. Potom prebieha formátovanie jednotlivých logických diskov. Po naformátovaní sa inštaluje operačný systém. Ten si sám počas inštalácie vytvára rôzne adresáre, do ktorých ukladá svoje súbory.

Spôsob rozdelenia disku na dve partície je veľmi rozšírený najmä na počítačoch využívaných v práci. Operačný systém nám disk **obsahujúci programy** bude zobrazovať ako disk **c:\** a disk **obsahujúci údaje** ako **d:\**. Informáciu o rozdelení jedného fyzického disku na logické má operačný systém zaznamenané v **MBR = Master Boot Record** v **PaT = Partition Table**. PaT sa nachádza v 0 – nultom sektore disku, čo je vlastne hlavný spúšťačí záznam. Tu sú základné informácie o disku v počítači. Ak máme disk rozdelený na viac logických diskov, potom sa nám jeden fyzický disk začína javiť ako viac diskov. Každý logický disk môže obsahovať iný operačný systém. Bežne môžu byť v MBR 4 záznamy, ale v prípade väčšieho počtu diskových oddielov na jednom médiu je možné odkázať na takzvaný **rozšírený**



**diskový oddiel = extended partition**. Aby sme však zachovali na disku poriadok a aby naše programy a údaje boli tematicky usporiadané, je potrebné vytvárať si na disku rôzne **tematicky zamerané adresáre (pričinky)**. Tak ako si ich vytváral operačný systém už počas inštalácie.

**Koreňovým adresárom** označujeme adresár **c:\**. Jemu sú podriadené všetky ostatné adresáre. Samozrejme môžeme adresáre ďalej vetviť a tak sa naše adresáre začínajú podobať stromu a jeho konárom. Tu by som mohol koreňový adresár prirovnať ku hlavnému kmeňu. Podadresáre ku hlavným konárom vyrastajúcim zo stromu a adresáre nižšej úrovne ku jednotlivým konárom a konárikom.

Platí však zásada, že na jednej úrovni môže byť len **jeden adresár určitého mena**. Toto si stráži sám operačný systém a neumožní nám vytvoriť na jednej hierarchickej úrovni adresár s menom už existujúceho adresár, organizácia súborov je veľmi dôležitá a nevyhnutná pre **zachovanie prehľadnosti, väzieb a nadväzností súborov**. Každý program inštalovaný do počítača si na disku vytvára vlastné a samostatné adresáre, tak aby mohol ukladať svoje vlastné súbory do tých oblastí disku, ktoré má priamo nadefinované v programe, alebo, ktoré mu zadáva užívateľ programu. Cestou v počítačovej terminológii rozumieme presný popis umiestnenia súboru od koreňového adresára až po samotný súbor. Môže to vyzeráť nasledovne:

**C:\hry\logické\piskvorky\piskvorky.exe**, kde

- C:\ je koreňový adresár
- C:\hry je adresár v ktorom sú všetky hry
- C:\hry\logické je adresár v ktorom sú uložené logické hry
- C:\hry\logické\piskvorky je adresár v ktorom je uložená hra piškvorky
- C:\hry\logické\piskvorky\piskvorky.exe je kompletná cesta aj s názvom programu

### **C, Aké výhody a riziká prináša využívanie Internet bankingu? Ako sa možno pred jednotlivými rizikami brániť?**

Rozvoj celosvetovej počítačovej siete prináša do vzťahu medzi klientom a bankou novú kvalitu a komfort. Internet banking je vlastne istá obdoba homebankingu, avšak na kvalitatívne vyššej úrovni, pretože pri použití odpadá potreba špeciálneho softvéru, ktorý zabezpečoval komunikáciu s bankou, a stačí obyčajný internetový prehliadač (internet explorer alebo Netscape Navigator). Avšak prvej návšteve banky sa klient ani tu nevyhne, pretože aj o túto službu musí požiadať banku, ktorá mu poskytne identifikačný kód.

Po otvorení webovej stránky v danej banke stačí, ak si klient vyberie voľbu internet banking a po správnej identifikácii má možnosť uskutočňovať pasívne, resp. aj aktívne operácie so svojim účtom v závislosti od jednotlivých podmienok každej banky.

Princíp internet bankingu.

Na prekonanie rizika verejnej siete sa používajú rôzne druhy kryptovania a šifrovania dát. Jedným z možných riešení je elektronický kľúč, ktorý v spojení so softvérom ponúka nasledovné bezpečnostné funkcie:

- autentizácia klienta
- autentizácia servera
- šifrovanie komunikácie
- generovanie certifikačných dát.

Pred každým prístupom na zmluvné služby Internet bankingu sa klient musí identifikovať a autentifikovať pomocou identifikačného čísla, PIN a hesla. Identifikačné číslo je uvedené v Zmluve o Elektronických a distribučných cestách. PIN a heslo si musí klient prevziať osobne v banke, s ktorou uzatvoril Zmluvu o Elektronických distribučných cestách.

## **15. Úloha č.1: Cyklus – porovnanie príkazov cyklu**

**Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?**

**Príklad**

**Napište program, pomocou ktorého vypočítate N faktoriál (N!), kde N je na vstupe z klávesnice. Použite cyklus s pevným počtom opakovaní. Vytvorte na papieri krokovaciu tabuľku pre N=6 a vysvetlite pomocou nej ako cyklus funguje.**

**Ako by ste pretvorili program, keby ste mali použiť cyklus s podmienkou?**

## **Úloha č.2:**

**A,V adresári Maturita sa nachádza 20 súborov rôzneho typu. Použite kompresiu dát, aby ste prenášali iba jeden súbor na USB kľúč. Súbor uložte na USB pod názvom maturita.zip (ukážte aj dekompresiu dát).**

Archivácia a zálohovanie

Počítač nie je zariadenie s neobmedzenou spoľahlivosťou! Je mnoho spôsobov, ako prísť o pracne vytvorené súbory (texty, údaje ap.). K tým najhorším patrí havária pevného disku, vírusová nákaza, omylom vyvolané formátovanie disku.

Zálohovanie údajov je záznam údajov na pamäťové médium za účelom neskoršieho použitia v prípade straty údajov v dôsledku havárie pevného disku alebo poškodenia údajov ľudským zásahom (omylom vymazané súbory, formátovanie disku a pod.). O zálohovaní hovoríme vtedy, ak je naším zámerom obnoviť stratené údaje.

Ak údaje zaznamenávame preto, že všetky sa nezmestia na HDD alebo ich nepotrebujeme, ale nechceme o ne prísť, tak hovoríme o archivácii.

Na zálohovanie je potrebné zaviesť spoľahlivý zálohovací systém.

Metódy zálohovania/archivácie

bežným kopírovaním

s použitím špeciálnych zálohovacích programov (napr. nástroje Windows)

s použitím komprimačných programov (WinRAR, WinZIP, ...)

Záznamové médiá: interný alebo externý pevný disk - výhodou je vysoká rýchlosť prenosu údajov a ľahká obsluha, výmenné médiá:

- magnetické (pásy)
- optické (CD, DVD, BR DVD)
- USB Flash (kľúč, karty)

## Komprimácia

Aké súbory sa oplatí komprimovať?

Výhodou archivačných formátov je, že dokážu do jedného balíka vložiť množstvo súborov čo je výhoda, ktorú určite oceníme napríklad pri mailovaní.

Ak však archivujeme súbory len za účelom zníženia ich veľkosti je dobre si uvedomiť, že komprimovať sa oplatí len súbory, na ktorých ešte nie je aplikovaná žiadna kompresia. Aj preto majú napríklad obrázky vo formáte TIFF alebo BMP po pridaní do archívu veľmi malú veľkosť, zatiaľ čo obrázky v JPEG nezmenšíme takmer vôbec. To isté platí aj o hudobných súboroch. Veľmi efektívne je komprimovať textové dokumenty, databázy, súbory s poštou, objemné savy z hier, jednoducho súbory kde je predpoklad výskytu redundancie.

## Komprimačné programy

Komprimačné (pakovacie) programy boli vyvinuté za účelom zmenšenia veľkosti súborov, pri zachovaní ich pôvodného obsahu. Ich cieľom je zmenšenie (skomprimovanie) súboru, ktorý potom zaberá podstatne menej miesta, ako jeho pôvodný tvar. So skomprimovaným súborom však nie je možné vykonávať bežné operácie, t.j. čítať ho, zapisovať a pod. Na to, aby ho bolo možné meniť, je potrebné jeho dekomprimovanie, čiže rozpakovanie - vrátenie do pôvodného stavu.

Princíp fungovania: Predstavte si text, v ktorom sa 30 krát vyskytuje slovo archivácia. Znamená to, že takýto text zaberá  $30 \times 10 = 300$  znakov. Keď namiesto slova archivácia použijeme nejakú značku, napr. #, ušetrí sa tým  $30 \times 9 = 270$  znakov. Pri komprimácii sa teda každé slovo archivácia nahradí značkou # a do tabuľky kódov sa zaznačí: # = archivácia. Tým sa výrazným spôsobom zmenší veľkosť textového súboru. Komprimačné programy pritom pracujú na ešte nižšej úrovni. Namiesto opakovania znakov sledujú opakovanie bitov (0 alebo 1).

Dekomprimácia je opačný proces ako komprimácia. V tomto prípade sa namiesto znakov tabuľky musia vložiť slová, ktoré zastupujú. Napr. namiesto značky # sa v komprimovanom texte vloží slovo archivácia.

K najznámejším komprimačným programom patria: ZIP, ARJ, ARC a RAR, PowerArchiver, FilZip, Windows Commander.

Komprimačný program môže vytvoriť niekoľko druhov výstupných súborov:

bežný archívny súbor - kapacita obmedzená len voľným miestom na cieľovom disku  
delený archív - vznikne rozdelením jedného veľkého archívu na niekoľko častí s danou maximálnou veľkosťou, napríklad pre uchovanie archívu na niekoľkých disketách (vhodné pre kapacitu niekoľko desiatok MB)

samorozbalovací archívny súbor (\*.exe)- SFX (Self-eXtracting) archív je archív s pripojeným spustiteľným modulom, ktorý po spustení dokáže súbory z archívu rozbaľiť. K rozbaleniu SFX archívu nie je teda treba žiadny externý program - sám vykoná dekompresiu do aktuálneho adresára

Základné operácie komprimačných programov:

ADD - pripojenie súborov do archívu, nový archív  
EXTRACT - dekompresia (obnovenie súborov z archívu)  
MOVE - presun súborov do archívu  
DELETE - vymazanie súborov z archívu  
UPDATE, FRESH - aktualizácia archívu  
TEST - kontrola celistvosti archívu

WinRAR - popis základných činností v programe

WinZip - popis základných činností v programe

Windows Commander - rýchla a prehľadná kompresia a dekompresia

## **B, Porovnajete základné parametre USB kľúča a DVD disku.**

- DVD (Digital Versatile Disc) – podobne ako CD je to optický dátový nosič. Rozmery disku sú rovnaké ako pri CD – 12 cm. Existuje však viac formátov – DVD-ROM, DVD-R, DVD-RW, DVD+R, DVD+RW, DVD-RAM. Základná kapacita je 4,7 GB. DVD však umožňuje používať dvojvrstvový zápis, vtedy je kapacita 7,96 GB. A tiež umožňuje ukladať dáta na obe strany média, takže celková kapacita DVD média je  $7,96 + 7,96 = 16$  GB.

- USB kľúč – elektronické médium, dáta sú uložené v čipe, ktorý však na ich udržanie nevyžaduje elektrinu ako napr. pamäť RAM. Výhodou je malá veľkosť a tiež pomerne vysoká odolnosť. Kapacita sa v súčasnosti pohybuje od 128 MB až do 8 GB.

**C, Charakterizujte jednotlivé typy softvéru z hľadiska právnej ochrany (freeware, shareware, ...) a vysvetlite ako sa dajú používať. Nájdite na počítači alebo na Internete príklady takýchto typov programov.**

Softvérová licencia

V prípade zaobstarávania softvérového vybavenia nejde o kúpu softvéru, ale len o získanie LICENCIE. Súčasťou každej dodávky originálneho produktu sú:

inštalčné médiá  
manuál

hlavným predmetom kúpy je vždy nehmotný majetok - programové vybavenie, duševné vlastníctvo autora. Prostredníctvom svojich obchodných partnerov potom autor môže poskytnúť právo používať svoje duševné vlastníctvo ďalším subjektom - používateľom softvéru. Cena za softvér je teda de facto poplatkom za poskytnutú licenciu - právo na používanie softvérového produktu.

Tento výslovný súhlas obvykle autor udeľuje generálne prostredníctvom licenčnej dohody EULA (End User License Agreement), ktorá je súčasťou každej dodávky softvéru.

V počítačovom priemysle softvér, ktorý je chránený autorskými právami a licencovaný pod softvérovou licenciou, sa delí do dvoch hlavných kategórií licenčných schém:

open source / slobodný softvér - musí mať dostupný zdrojový kód; koncoví používatelia musia mať právo modifikovať a šíriť softvér, ako aj právo baliť a predávať produkt. Tieto podmienky spĺňa softvér so zdrojovým kódom uvoľneným ako public domain aj pod GNU General Public License (GPL). Open-source licencie môžu obsahovať ešte ďalšie obmedzujúce podmienky, ako napr. uvedenie mena autora a uvedenie autorských práv.

closed source / proprietárna licenčná schéma - má obmedzenia na jeho používanie a kopírovanie, obvykle vynucované vlastníkom (proprietárom). Používaniu, kopírovaniu, alebo úprave je možno zabrániť právnymi alebo technickými prostriedkami. Medzi právne prostriedky patrí licencovanie softvéru, autorské a patentové práva.

Označovanie programov z pohľadu licenčnej politiky:

OEM (Original Equipment Manufacture) - takto označované produkty sú predinštalované výrobcami počítačov v nových počítačoch a dodávané za cenovo veľmi zaujímavých podmienok.

Licencia nadobudnutá formou OEM zostáva po celú dobu používania produktu viazaná na počítač, na ktorý bola prvotne nainštalovaná. S týmto počítačom môže byť ďalej predaná alebo prevedená na ďalšieho používateľa. Licenciu OEM nie je možné preniesť na iný počítač - v prípade likvidácie počítača alebo jeho vyradenia dochádza zároveň k zániku licencie.

Softvér v krabici (FPP - Full Package Product) - je určený pre koncový predaj zákazníkom, ktorí zakupujú len jednu alebo dve licencie. Softvér zakúpený ako FPP je možné ďalej predat' alebo previesť na iný subjekt.

Shareware - je šírený zadarmo napríklad cestou internetu, alebo na rôznych CD. Takýto program je možné používať na základe pravidiel uvedených v licenčnej zmluve. Pre plné využitie je potrebné zaplatiť nejakú, väčšinou symbolickú cenu, za ktorú užívateľ obdrží aktivačný kľúč, alebo heslo.

Freeware - je šírený zadarmo, je možné ho získať napríklad stiahnutím z internetu, alebo z rôznych CD. Program je možné používať zadarmo po neobmedzenú dobu, je možné ho šíriť ďalej.

Demoverzia - softvér uvoľnený do užívania zadarmo, niektoré funkcie programu sú obmedzené

Trialware - je časovo obmedzený program, po uplynutí určenej doby ho musíte buď zakúpiť, alebo odinštalovať, inak porušujete autorské práva toho produktu.

GNU - GPL - tzv. verejná licencia GNU (GNU Public Licence) umožňuje voľne šíriť softvér z pohľadu ceny i zdrojového kódu. Najväčšia výhoda takéhoto SW vychádza od samotných používateľov - ak potrebujú nejakú novú funkciu, môžu ju pridať do programu a potom ju poskytnúť ďalej (stále však pod licenciou GPL!).

## 16.

### Úloha č.1: Podmienený príkaz - úplné a neúplné vetvenie

Vysvetlite pojem podmienka, vysvetlite rozdiel medzi úplným a neúplným vetvením.

#### Príklad

Na vstupe sú 3 navzájom rôzne celé čísla A, B, C, ktoré zadá užívateľ z klávesnice. Program navzájom porovná všetky 3 čísla a vypíše ich vzostupne.

Úlohu môžete vyriešiť tak, že pomocou neúplného podmieneného príkazu vypíšete všetkých 6 existujúcich možností. Ako by ste program pretvorili tak, aby ste využili úplné vetvenie cez vnorené podmienené príkazy?

### Úloha č.2:

a) Vysvetlite rozdiel medzi 2D a 3D grafikou.

**Grafika 2D** Existujú 2 základné prístupy ke 2D grafice: [vektorová](#) a [rastrová](#). **Vektorová grafika** ukláda presná geometrická data, napr. súradnice bodů, propojení mezi body (úsečky a křivky) a vyplnění tvarů. Většina vektorových graf. systémů umožňuje použít standardní tvary jako kružnice, čtverce atd.

Základem **rastrové grafiky** je pravidelná síť pixelů, organizovaná jako dvourozměrná matice bodů. Každý pixel nese specifické informace, napr. o jasnosti, barvě, průhlednosti bodu, nebo kombinaci těchto hodnot. Obrázek v rastrové grafice má omezené rozlišení, které se udává počtem řádků a sloupců. Často se kombinuje rastrová a vektorová grafika v souborových formátech jako PDF či SWF. (4)

### Grafika 3D

3D grafika je příbuzná [vektorové](#) 2D grafice. Také pracuje se souřadnicemi bodů a informacemi o úsečkách, křivkách a plochách, ale data jsou uložena ve trojrozměrném souřadnicovém systému. Z těchto trojrozměrných dat reprezentujících tělesa je potom [renderován](#) » 2D obrázek.

Různými technikami se dají ve 3D grafice vytvořit velmi realisticky vypadající obrázky díky věrné simulaci světelných a optických jevů jako jsou stíny, odrazy, lom světla či kaustika. Pokročilé vývojové nástroje umožňují i realistické animace včetně pohybů oděvu, vlasů, vodní hladiny a simulace fyzikálních jevů jako je gravitace a odrazy. (4)

### Důležité pojmy

#### Príklad:

Nakeslite priestorové geometrické telesá: guľa, kváder, štvorboký ihlan, telesá vnorte do seba, na kreslenie použite program Google Sketchup alebo ekvivalent).

## **b. Charakterizujte štandardné vstupno-výstupné zariadenia (klávesnica, monitor) a vysvetlite princíp komunikácie procesora s nimi**

**Klávesnica** tvorí základné vstupné zariadenie počítača a slúži na bežnú komunikáciu používateľa s počítačom. Typická klávesnica má 102 až 108 klávesov, ale mnoho klávesníc býva doplnených o ďalšie klávesy (napr. na ovládanie multimédií a internetu).

Principiálne rozoznávame klávesy:

**aktívne** – ich stlačenie vyvoláva na obrazovke odozvu tj. výpis znaku alebo pohyb kurzora. **pasívne** – ich samotné stlačenie nevyvolá odozvu, ale používajú sa pri dvojtrojhmatoch, napr. CTRL, SHIFT...

*Alfanumerická časť* – tvorí základ klávesnice. Obsahuje písmená, číslice a niektoré špeciálne klávesy napr. ENTER, BACKSPACE, TAB...

*Časť riadiacich klávesov* – obsahuje klávesy na pohyb kurzora, buď o jednu pozíciu (šípky), alebo o niekoľko pozícií (*home, end, page down, page up*), tiež klávesy *insert* na prepínanie režimov a *delete* na mazanie, špeciálne klávesy: *pause, scroll lock, print screen*, ktorých funkciou je zastavenie výpisu na obrazovku, rolovanie obrazovky a skopírovanie obrazovky na tlačiareň.

*Numerická časť* – slúži na zadávanie čísel a numerických operátorov. Číselné klávesy fungujú iba ak je aktívny číselný mód, inak sú aktívne klávesy na pohyb kurzora.

*Časť ESC* – kláves, ktorý väčšina aplikácií používa na ukončenie činnosti.

*Funkčné klávesy F1 – F12* – klávesy, ktoré v programoch zastávajú isté funkcie. Napr. F10 sa aktivuje hlavná ponuka programu, F1 je pomocník, atď.

Pripojenie: PS/2, bezdrôtové, USB

## Monitor

- výstupné zariadenie na zobrazenie obrazu generovaného počítačom a spracovaného grafickou kartou na obrazovku. Pri výbere monitora rozhoduje jeho **rozlíšenie, obnovovacia frekvencia, farebná hĺbka, druh obrazovky a dĺžka jej uhlopriečky**.

**Rozlíšenie** – určuje počet grafických bodov, ktoré je schopný monitor zobraziť. Udáva sa ako dve čísla, jedno určuje počet stĺpcov grafických bodov (pixel) a druhé počet riadkov. Napr. 1024x768.

Rozlíšenie súvisí aj s *dĺžkou uhlopriečky* displeja, ktorá sa udáva v palcoch. Meria sa dĺžka v diagonále a medzi CRT a LCD tu vzniká rozdiel. Ak si kúpite 17-palcový displej CRT, jeho skutočná viditeľná plocha sa pohybuje niekde nad úrovňou 15 palcov. Do plnej „sedemnástky“ sa započítava aj plocha v okolí displeja. Pri LCD sa udáva viditeľná zobrazovacia plocha displeja.

**Farebná hĺbka** - počet zobraziteľných farieb, udáva sa ako číslo určujúce počet bitov vyhradených pre informáciu o farbe jedného bodu. (napr. farebná hĺbka 16 bitov znamená, že monitor je schopný zobraziť  $2^{16}$  farieb). Štandardom je 32 bitov, keďže ľudské oko nie je schopné rozpoznať viac farieb a predpokladá sa, že sa už nebude zvyšovať.

Farby sa vytvárajú z troch základných farieb RGB (red, green, blue) - skladaním rôznych kombinácií intenzity základných farieb môžeme vygenerovať viac ako 16 miliónov farieb. Tzv. "tru color" typ je 24 bitový režim, kde farba jedného bodu je popísaná tromi 8bitovými číslami, udávajúcimi jas R,G,B zložiek, pričom každá zložka môže mať až 256 úrovní. Skladanie farieb môžeme znázorniť tzv. **RGB modelom**.

**Obnovovacia frekvencia** - určuje koľkokrát za sekundu je každý bod obnovený.

**Druh obrazovky** - CRT a LCD. CRT sú lacnejšie, ale zaberajú viac miesta. LCD nie sú vhodné na profesionálnu grafiku.

Základom CRT obrazovky sú tri elektrónové delá vyžarujúce lúče elektrónov, ktoré pri dopade na tienidlo na malú chvíľu spôsobia rozsvietenie grafického bodu. Ako náhle lúč opustí bod, bod začne tmavnúť, preto je potrebné, aby elektrónové delá niekoľkokrát za sekundu obnovili každý bod.

LCD obrazovky sú tvorené pevnou mriežkou zloženou z buniek tekutých kryštálov. To, či nejaký bod svieti, určujú elektródy v bunkách. Najdôležitejším parametrom pre LCD je **doba odozvy**, t.j. **minimálny čas, ktorý potrebuje jeden prvok na rozsvietenie a zhasnutie**. Udáva sa v ms. (je to obdoba parametra obnovovacej frekvencie u CRT).

### c) **Vysvetlite princíp paketovej komunikácie a spôsob doručovania paketov do cieľovej siete prostredníctvom smerovačov**

Súbory údajov sa v sieťach neprenášajú v celkoch, ale delia sa na menšie časti – pakety. Paket je ucelený blok údajov, ktorý sa v sieti prenáša a spracúva ako celok. Má svoju definovanú štruktúru. Obsahuje hlavičku a samotné údaje. V hlavičke sú uložené informácie potrebné na prepravu a správne doručenie paketu. Okrem iného obsahuje adresy odosielateľa a prijímateľa, dĺžku paketu, poradové číslo paketu a kontrolný súčet údajov. Kontrolný súčet údajov je informácia, podľa ktorej je možné overiť, či sa údaje pri prenose nepoškodili. Každý paket môže byť prepravený inou cestou a do cieľa môžu doraziť v rôznom poradí. TCP (Transmission Control Protokol) protokol sa postará o to, aby do cieľa dorazili všetky pakety a v cieľi sa zoradili v správnom poradí.

Najdôležitejším uzlom v sieti je smerovač, ktorého hlavnou funkciou je spájať sieťové prvky, i celé siete a zabezpečovať ich spoľahlivú diaľkovú komunikáciu. Funguje ako smerovač údajov, ktorý vyberá najvhodnejšiu a najrýchlejšiu cestu pre údaje.

Na zistenie cesty, ktorou prejde paket od nášho počítača k počítaču so zadanou IP adresou slúži príkaz tracer, napr. tracer 158.193.138.40.

### **17. Úloha č.1: Cyklus s pevným počtom opakovaní – prerušenie cyklu**

Vysvetlite pojmy: cyklus, preskočenie krokov v cykle (skok), predčasné ukončenie cyklu (prerušenie).

*Príklad*

Napište program, ktorý bude simulovať vkladanie PIN kódu do bankomatu. Počítač bude pomocou príkazu cyklu s pevným počtom opakovaní načítavať od užívateľa celočíselný 4-ciferný PIN kód (ak nezadá 4-ciferné celé číslo, musí zadať znova). Ak zadá správny kód, cyklus sa predčasne ukončí. Po treťom nesprávnom zadaní cyklus skončí.

### **Úloha č.2:**



- a) Vysvetlite pojmy: informácia, údaj, uveďte príklady. Vymenujte a vysvetlite 4 základné etapy manipulácie s informáciami okolo nás v ľudskej spoločnosti. Uveďte základnú jednotku informácie a ďalšie jednotky, definujte prevodové vzťahy medzi nimi.  
 Vykonajte nasledovné prevody jednotiek:  $3 \text{ kiB} = ? \text{ B}$ ,  $20 \text{ B} = ? \text{ b}$ ,  $2 \text{ MiB} = ? \text{ B}$ ,  $32 \text{ b} = ? \text{ B}$ .

#### Informácia

zahrňuje v sebe správu spolu s jej významom pre príjemcu; je to správa, ktorá vyjadruje istý stav, slúži nejakému cieľu alebo vyvoláva nejakú akciu.

#### Informatika

vedná disciplína, ktorá sa zoberá počítačmi a metódami spracovania najrôznejších informácií pomocou nich.

#### Informačný systém

slúži na uchovávanie, znovuzískavanie, spájanie a vyhodnocovanie informácií.

Pozostáva zo zariadenia na spracovanie dát, systému banky dát a vyhodnocovacích programov.

#### Jednotky informácie

- 1 bit; 1 b - základná jednotka
- 1 bajt (anglicky byte);  $1 \text{ B} = 8 \text{ bitov}$ ;  $1 \text{ B} = 2^8$ , t. j. 256 rôznych stavov
- Násobky bajtu:

Kilobajt (angl. kilobyte), je jednotka, ktorá pôvodne predstavovala 1024 bajtov, teda  $2^{10}$  bajtov. Pretože počítače pracujú v dvojkovej sústave, bol zvolený tento násobok namiesto obvyklého 1000 a práve kvôli svojej "neokrúhlosti" sa zaviedlo označenie KB (a nie kB).

V decembri 1998 IEC (International Electrotechnical Commission) odporučila pre počítačové jednotky nový systém označovania násobkov, v ktorom bola pre pôvodný 1 kilobyte = 1024 B navrhnuté označenie kibibyte (KiB) a zavedená jednotka 1 kilobyte = 1000 B so skratkou 1 kB, tak ako je obvyklé v sústave SI. Podobne sú na tom aj ostatné odvodené jednotky:

predpona SI			binárna predpona (IEC 60027-2)		
Meno	Symbol	Násobok	Meno	Symbol	Násobok
kilobajt	kB	$10^3$	kibibajt	KiB	$2^{10}$
megabajt	MB	$10^6$	mebibajt	MiB	$2^{20}$
gigabajt	GB	$10^9$	gibibajt	GiB	$2^{30}$
terabajt	TB	$10^{12}$	tebibajt	TiB	$2^{40}$
petabajt	PB	$10^{15}$	pebibajt	PiB	$2^{50}$
exabajt	EB	$10^{18}$	exbibajt	EiB	$2^{60}$
zettabajt	ZB	$10^{21}$	zebibajt	ZiB	$2^{70}$
yottabajt	YB	$10^{24}$	yobibajt	YiB	$2^{80}$

- b. Vymenujte funkcie operačného systému.

---

Operačný systém je nenahraditeľné rozhranie pri **komunikácii hardvéru a softvéru**.

Bez operačného systému nie je možné počítač používať, pretože všetky príkazy užívateľa musí najprv spracovať OS. Zjednodušene OS je súbor programov, ktorý riadi a kontroluje činnosť hardwaru, ale aj softwaru v počítači, stará sa o efektívne využitie operačnej pamäte a procesora, o optimálnu komunikáciu medzi všetkými používanými technickými aj programovými prostriedkami.

**OS v širšom ponímaní** - súhrnné označenie pre technické a programové prostriedky počítača (spôsob, akým je celý systém tvorený).

**OS v užšom zmysle** - je to skupina programových produktov (programov a údajov) – softvér, zabezpečujúci čiastočne alebo úplne tieto funkcie:

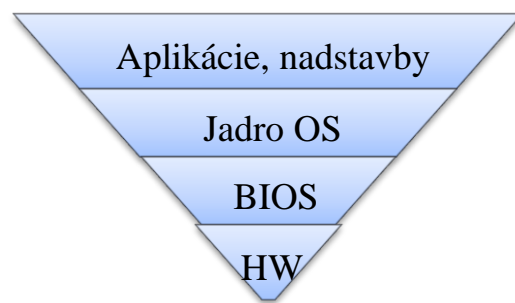
- **komunikácia s užívateľom** prostredníctvom periférnych zariadení
- **pridelovanie prostriedkov systému** užívateľom alebo aplikáciám
- **pridelovanie času** aplikáciám (multitasking)
- **organizácia programov a súborov údajov** na vonkajších pamäťových médiách (príkazový systém umožňujúci vytváranie, editovanie, prezeranie, kopírovanie, premenovávanie, mazanie, zálohovanie, komprimovanie a ďalšie operácie so súbormi)
- **diagnostické funkcie** - autokontrola systému, možnosť analýzy a odstraňovania chýb
- **zabezpečovacie funkcie** - ochraňuje systém proti strate údajov pri výpadku napätia, proti neoprávnenej alebo neodbornej manipulácii, ktorá by mohla poškodiť programové produkty
- **komunikácia s inými zariadeniami** (počítače, tlačiarne) **v sieti** - pri použití sieťových operačných systémov.

### Súčasti operačného systému:

1. **Jadro** (exekutíva = výkonná časť) operačného systému - táto časť je rezidentne umiestnená v pamäti; podľa potreby sa inicializuje (spúšťa) alebo nahráva do pamäte ostatné dôležité časti operačného systému,
2. **Monitor** operačného systému (od slova monitorovať = sledovať) - tiež nazývaný interpreter príkazov; zabezpečuje komunikáciu systému s užívateľom. Prijíma a analyzuje impulzy z klávesnice, zisťuje význam systémových príkazov, vypisuje príslušné odozvy na zobrazovacie (výstupné) zariadenie,...
3. **Ovládače** (drivers) - obslužné programy vstupno/výstupných zariadení.

## Vrstvy operačného systému

1. **Najnižšiu - fyzickú** tvorí **hardware** počítača,
2. nad ním operujú programy označované ako „**firmware**“ - **BIOSu** (Basic Input Output System), na nich sú postavené vyššie „logické vrstvy“ operačného systému.
3. Pomocou služieb a prostriedkov operačného systému sú vytvorené **používateľské prostredia, v ktorých prebiehajú aplikácie.**



- b) Spustíte na počítači internetový prehliadač a zistíte vlastníka autorských práv a vlastníka licencie programu.

## 18. Úloha č.1: Cyklus – porovnanie cyklov

Vysvetlite význam a použitie cyklu **for** a cyklu **while**.

### Príklad

Napište program, ktorý bude načítavať ceny maximálne 20 položiek (minimálna hodnota jednej položky bude 1 Euro) a na konci vypíše celkovú sumu za nákup. Zadávanie možno predčasne ukončiť zadaním 0. Nastavte program tak, aby po prekročení sumy 100 Eúr, odrátal za každú ďalšiu položku prímiovú zľavu 1 Euro. Použite príkaz **while**.

Ako by ste pretvorili program, keby ste mali použiť cyklus **for**?

## Úloha č.2:

- a) Upravte pôvodnú zvukovú nahrávku tak, aby v upravenej nahrávke bol použitý efekt ozveny / na úpravu použite program AUDACITY a priložený súbor
- b) Popíšte spôsob jednoznačnej adresácie počítačov a sietí v Internete pomocou IP adries. Pomocou ľubovoľného nástroja alebo príkazov operačného systému zistite IP adresu vášho počítača v lokálnej sieti.

Adresovanie paketov v sieti internet sa riadi podľa komunikačného protokolu s názvom Internet Protocol (IP). Ide o hlavný komunikačný protokol zo sady internetových protokolov. Jeho funkcia smerovania umožňuje vzájomné prepojenie sietí, čím v podstate vytvára internet.

Komunikačný protokol IP má za úlohu dodávať pakety od zdrojového hostiteľa do cieľového hostiteľa iba na základe adresy IP. Za týmto účelom IP definuje štruktúru paketov obsahujúcich zapuzdrené údaje, ktoré majú byť dodané. Definuje tiež metódy adresovania, ktoré sa používajú na ich označenie zdrojovou a cieľovou adresou. Paket je teda niečo ako poštový balík na ktorom je napísaná zdrojová a cieľová adresa.

V súčasnosti sa používajú dve verzie komunikačného protokolu - Internet Protocol verzie 4 (IPv4), ktorý je dominantný a jeho nástupca Internet Protocol verzie 6 (IPv6).

## Jednoznačná IP adresa

Aby bolo možné doručovať pakety, je potrebné aby každá stanica, ba dokonca každé pripojené rozhranie stanice malo svoju jednoznačnú adresu. Každé miesto v sieti, ktoré má svoju vlastnú adresu sa nazýva **hostiteľ** (host).

### IPv4

Protokol IP verzie 4 definovaný štandardom [RFC 791](#) používa 32 bitovú adresu, ktorá vytvára adresný priestor s  $4294967296$  ( $2^{32}$ ) adresami. Adresy sa najčastejšie zapisujú ako štyri desiatkové čísla v rozsahu 0-255 a oddeľujú sa bodkami. Napr. 192.168.1.150

### IPv6

Protokol IP verzie 6 definovaný štandardom [RFC 2460](#) používa 128 bitovú adresu, ktorá vytvára adresný priestor so  $3,4 \times 10^{38}$  adresami. Adresy sa zapisujú ako osem skupín štvorciferných šesťnástkových čísel oddelených dvojbodkami, napríklad 2001:0db8:85a3:0042:1000:8a2e:0370:7334.

## Adresa siete

Aj napriek tomu, že adresný priestor IPv4 bol vyčerpaný už 3. februára 2011, Protokol IPv4 stále používa väčšina počítačových sietí. Používanie týchto adries je však stále možné vďaka zavedeniu tried adries, vnútrodoménovému smerovaniu a prekladu sieťových adries.

### IPv4

**Triedy adries** rozdelili adresy na podľa veľkosti sietí, pre ktoré sú určené. Každá adresa bola rozdelená na dve časti, prvá časť identifikuje sieť a druhá identifikuje hostiteľa v sieti. Trieda A má menšiu časť vyhradenú na adresu siete a väčšiu časť pre adresu hostiteľa - je teda určená pre veľké siete, ktorých nie je veľa. Naopak trieda C má väčšiu časť vyhradenú pre adresu siete a menšiu pre adresu hostiteľa. Popis všetkých tried je v nasledujúcej tabuľke:

trieda	počiatočné bity	počet bitov pre adresu siete	počet bitov pre adresu stanice	počet sietí	počet adries v sieti	počiatočná adresa	koncová adresa
A	0	7	24	128 ( $2^7$ )	16,777,216 ( $2^{24}$ )	0.0.0.0	127.255.255.255
B	10	14	16	16,384 ( $2^{14}$ )	65,536 ( $2^{16}$ )	128.0.0.0	191.255.255.255
C	110	21	8	2,097,152 ( $2^{21}$ )	256 ( $2^8$ )	192.0.0.0	223.255.255.255
D (viacsmerové vysielanie)	1110	nedefinované	nedefinované	nedefinované	nedefinované	224.0.0.0	239.255.255.255
E (rezervovaná)	1111	nedefinované	nedefinované	nedefinované	nedefinované	240.0.0.0	255.255.255.255

Neskôr bolo vytvorené **Beztriedové vnútrodoménové smerovanie** známe pod skratkou **CIDR**, ktoré definuje štandard [RFC 4632](#). Tento spôsob adresovania odstraňuje striktné delenie tried a umožňuje nastaviť, koľko bitov má byť použitých

pre adresu siete a koľko pre adresu hostiteľa.

Zápis CIDR pozostáva z IP adresy a ku nej pridruženému smerovaciemu prefixu. Ten sa určí tak že sa za adresu IP napíše lomka a za ňu desiatkové číslo určujúce počet bitov tvoriacich smerovací prefix, Napr. zápis 192.168.2.0/24 hovorí, že ide o sieť ktorej adresa je určená prvými 24 bitmi - to znamená, že adresa siete je 192.168.2 a pre adresu hostiteľov v rámci siete zostáva zvyšných 8 bitov.

c) Vysvetlite pojem počítačová bezpečnosť.

**Prečo chrániť počítač** Pod pojmom počítačová bezpečnosť rozumieme ochranu údajov v našom PC, ale aj ochranu údajov pri elektronickej komunikácii v sieťach a na Internete. Pri komunikácii v sieti a na Internete (napr. Internet banking, platba cez Internet a pod.) platí pravidlo, že nepoužívame cudzie a neznáme počítače, na závažnejšie operácie ani "Internetové kaviarne"! Po ukončení práce zavrieme všetky okná a odhlásime sa z počítača, aby nebolo možné zneužiť naše údaje.

---

Denne sa v počítačoch ukladá obrovské množstvo informácií (v podnikoch dokumenty na zabezpečenie podnikových procesov, v domácnosti fotografie, videá, prednášky, seminárne práce,...)

Všetky tieto údaje treba chrániť a zároveň archivovať. Ich poškodenie alebo strata môžu spôsobiť nevyčísliteľné škody a opätovne zostavené zmazané alebo poškodené dokumenty už nikdy nebudú identické.

### **Najčastešie formy poškodenia alebo straty údajov**

#### Externé prostredie

- počítačové vírusy,
- nebezpečné e-maily, ktoré využívajú bezpečnostné chyby informačného systému a aplikácií,
- útoky tretích strán s cieľom pripojiť sa k počítaču.

#### Interné prostredie

- poškodenie údajov aplikáciou,
- otvorenosť systému používanou aplikáciou k externému prostrediu,
- poškodenie alebo zmazanie údajov nedbanlivosťou alebo nevedomosťou neúmyselného charakteru,
- cieľavedomé zmazanie údajov (už ich nebudem potrebovať),
- poškodenie počítačového hardvéru, ktoré má za následok stratu alebo poškodenie údajov.

Rozdelenie zabezpečenia:

- aktívna ochrana údajov - neustála kontrola a monitorovanie systému,
- pasívna ochrana údajov - archivácia údajov, vytváranie log súborov, ktoré kontrolujú a monitorujú systém.

### **Aktívna ochrana údajov**

- Firewall - úlohou firewallu je ochrana počítača pred nežiadúcimi útokmi alebo komunikáciou (hlavne z Internetu), ktorých úlohou je poškodiť alebo zmazať údaje. Firewall chráni používateľa pred útokmi prostredníctvom počítačovej siete. Jeho činnosť sa prejavuje hlásením požiadaviek na prístup z určitej lokality a užívateľ môže prístup povoliť alebo zamietnuť. Každý firewall má svoje nastavenia, kde sa určuje úroveň zabezpečenia a kontroly systému.
- Antivírus a antispyware - úlohou oboch je chrániť pred škodlivými vplyvmi iných programov a aplikácií, ktoré sa môžu nachádzať v e-mailoch, dokumentoch, aplikáciách, hrách, ... Prvotnou úlohou používateľa je používať aplikácie a súbory z overených zdrojov (stránky výrobcov, originálne balenia, ...).
  - úlohy antivírusových programov:
    - detekcia,
    - odstránenie,
    - monitoring operačného systému a správania sa spustených aplikácií.
    - pozri kapitolu [Vírusy](#)
  - úlohy antispyware:

čo je spyware: reklamné okná, zbiera osobné údaje, heslá, identifikačné údaje, navštívené webové stránky, mení nastavenie aplikácií alebo systému,...

    - detekuje spyware,
    - bráni jeho inštalácii,
    - odstránenie.
- Update operačného systému a aplikácií - pravidelným update sa dá predísť poškodeniu operačného systému, aplikácií alebo údajov. Prostredníctvom update získa užívateľ najnovšie opravné balíčky, ktoré riešia nedostatky existujúcich verzií. Update sa najčastejšie vzťahuje na operačný systém, internetové prehliadače, aplikácie elektronickej pošty, antivírusové a antispamové aplikácie.

Predchádzanie a prevencia je najúčinnější zbraň v súčasnom svete internetu a počítačových sietí.

### **Pasívna ochrana údajov**

Archivácia operačného systému a údajov:

- uloženie údajov a ich opätovné použitie
  - ukladáme na médium, ktoré nie je súčasťou počítača - napr. disk CD alebo DVD, na kartu USB flash alebo do iného počítača, ak je náš počítač pripojený k sieti,

- o pre získanie ešte väčšieho priestoru, je možné údaje pred uložením skomprimovať.

## 19. Úloha č.1: Polia – jednorozmerné pole reálnych čísel

Vysvetlite pojmy: jednorozmerné pole, prvok poľa, index prvku. Akým spôsobom deklarujeme pole reálnych čísel?

*Príklad*

Napíšte jednoduchý program, ktorý ilustruje definíciu a použitie jednorozmerného poľa reálnych čísel. Na vstupe z klávesnice sú dve polia po 4 reálne čísla:

- pole A (obsahujúce ceny výrobkov v Eurách)
- pole B (obsahujúce množstvo v kg).

Vypíšte na obrazovku obe polia do 2 samostatných riadkov, tak aby súvisiace hodnoty boli pod sebou. Vypočítajte celkovú cenu výrobkov a vypíšte ju na obrazovku.

### Úloha č.2:

- a) Vysvetlite pojmy: číselná sústava, základ číselnej sústavy. Vymenujte základné číselné sústavy používané v informatike a vysvetlite zápis čísla v týchto sústavách. Vysvetlite a ukážte na konkrétnom príklade algoritmus prevodu čísla z dekadického do binárnej sústavy a naopak:

- prevedte z desiatkovej sústavy číslo 133 do dvojkovej sústavy
- prevedte z dvojkovej sústavy číslo 10101010 do desiatkovej sústavy

### Rozdelenie číselných sústav

Číselné sústavy môžeme rozdeliť do dvoch kategórií - pozičné a nepozičné. Medzi nepozičné sústavy patria napríklad rímske čísla. Pozičné číselné sústavy sú sústavy pozostávajúce z určitého počtu cifier (v desiatkovej ich je desať 0 - 9), ktoré sú usporiadané do pozícií, ktoré sa tiež nazývajú rády. Tá istá cifra umiestnená na inom ráde má inú hodnotu. Rády sú číslované sprava doľava od 0 vyššie, pretože prvá cifra sprava má najmenšiu hodnotu. V desiatkovej sústave posunutím cifry o jedno miesto (rád) doľava, cifra nadobudne desaťkrát väčšiu hodnotu. V dvojkovej sústave posunutím cifry o jeden rád doľava, cifra bude mať dvakrát väčšiu hodnotu. Číslo 10 preto bude mať hodnotu základu číselnej sústavy (v desiatkovej sústave má zápis 10 hodnotu desať, v dvojkovej má hodnotu dva, v osmičkovej osem a šestnástkovej šestnásť).

### Prevod čísel do desiatkovej sústavy.

Číslo v ľubovoľnej sústave sa do desiatkovej sústavy prevedie nasledovne:

1. Očíslujeme si rády čísla, t.j. pod prvú cifru sprava napíšeme 0 pod druhú 1 pod tretiu 2  
...
2. výsledné číslo v desiatkovej sústave potom dostaneme ako súčet cifier vynásobený základom sústavy na príslušný rád.

Tento postup funguje pre všetky sústavy vrátane desiatkovej (prevod do seba sama). Napríklad číslo 10294 sa dá popísaným postupom napísať takto:

$$10\ 294 = 1 \cdot 10^4 + 0 \cdot 10^3 + 2 \cdot 10^2 + 9 \cdot 10^1 + 4 \cdot 10^0$$

Prirodzenou sústavou počítača je dvojková - **binárna** sústava (kvôli jednotke informácie 1 bitu, ktorý môže nadobúdať dve hodnoty 0 a 1). V jednej pamäťovej bunke počítača je však 8 bitov, preto vždy dopĺňujeme cifry tak, aby bol ich počet deliteľný číslom 8 (8, 16, 24, 32 ...).  
 $(01101100)_b = 0.2^7 + 1.2^6 + 1.2^5 + 0.2^4 + 1.2^3 + 1.2^2 + 0.2^1 + 0.2^0 = 108$

Pri sústavách so základom vyšším ako 10 (napríklad šesťnástkovej) pre vyššie cifry nemáme prislúchajúce symboly, preto ich nahrádzame písmenami (v šesťnástkovej sústave sú to písmená ABCDEF, kde A=10, B=11, C=12, D=13, E=14, F=15).

Obrovskou výhodou šesťnástkovej - **hexadecimálnej** sústavy je to, že hodnotu, ktorú v dvojkovej sústave je potrebné zapísať pomocou 8 cifier, v šesťnástkovej zapíšeme presne pomocou dvoch cifier. Napríklad najvyššia možná hodnota v dvojkovej sústave, ktorá sa dá zapísať pomocou 8 cifier, v šesťnástkovej sústave zapíšeme  $(11111111)_b = (FF)_H$ . Príklad prevodu do desiatkovej sústavy:

$$(E4)_H = E \cdot 16^1 + 4 \cdot 16^0 = 14 \cdot 16^1 + 4 \cdot 16^0 = 228$$

Převěd'te číslo F1AE z hexadecimální (šestnáctkové) soustavy do desítkové.

$$E \cdot 160 = 14 \cdot 160 = 2240$$

$$A \cdot 161 = 10 \cdot 161 = 1610$$

$$1 \cdot 162 = 162$$

$$F \cdot 163 = 15 \cdot 163 = 2445$$

Výsledek je 61870

### Prevod čísel z desiatkovej do inej sústavy.

Ak chceme previesť číslo z desiatkovej sústavy do inej sústavy, musíme číslo deliť základom sústavy, do ktorej číslo chceme previesť, až pokiaľ nedostaneme číslo nula. Po každom delení si zapíšeme zvyšok, pričom zvyšok po prvom delení je cifra najnižšieho (nultého) rádu, zvyšok po druhom delení udáva cifru prvého rádu ... Takto napríklad prevedieme číslo 215 do dvojkovej sústavy:

$$215 : 2 = 107; 107 : 2 = 53; 53 : 2 = 26; 26 : 2 = 13; 13 : 2 = 6; 6 : 2 = 3; 3 : 2 = 1; 1 : 2 = 0$$

$$\text{zvyšok } 1 \quad \text{zvyšok } 1 \quad \text{zvyšok } 1 \quad \text{zvyšok } 0 \quad \text{zvyšok } 1 \quad \text{zvyšok } 0 \quad \text{zvyšok } 1 \quad \text{zvyšok } 1$$

Výsledok teda je **11010111** (zvyšky zapísané v opačnom poradí).

A takto rovnaké číslo 215 prevedieme do šesťnástkovej sústavy.

$$215 : 16 = 13; 13 : 16 = 0;$$

$$\text{zvyšok } 7 \quad \text{zvyšok } 13 \text{ (to zodpovedá symbolu D)}$$

Výsledok je teda **D7**.

Rovnako funguje i prevod z desiatkovej sústavy do seba sama.

$$215 : 10 = 21; 21 : 10 = 2; 2 : 10 = 0;$$

$$\text{zvyšok } 5 \quad \text{zvyšok } 1 \quad \text{zvyšok } 2$$

Dostali sme teda skutočne číslo **215**.

### Prevod čísel z dvojkovej do šesťnástkovej sústavy.

Sila šesťnástkovej sústavy, ako som písal vyššie, spočíva v jednoduchšom zápise čísla z dvojkovej sústavy. Jednoduchší zápis by však bol na nič, keby bol prevod z dvojkovej sústavy do šesťnástkovej zložitý. Našťastie je veľmi jednoduchý. Stačí číslo dvojkovej sústavy rozdeliť na štvorcíslika začínajúc od najnižšieho rádu (teda sprava). Každé štvorcíslika môže nadobúdať hodnoty od 0 do 15, čo je práve jedna cifra čísla šesťnástkovej sústavy. Napríklad číslo 11010111 prevedieme nasledovným spôsobom:

$$11010111 = 1101 \ 0111; (1101)_b = (13)_d = (D)_H; (0111)_b = (7)_d = (7)_H$$

Takže výsledok je **D7**.

### Úloha 1

*Pri prevode údajov z jedného systému do iného potrebujeme niektoré čísla v osmičkovej sústave prepísať do šesťnástkovej sústavy. Aký zápis v šesťnástkovej sústave má číslo, ktoré má*



v osmičkovéj sústave zápis 1234567?

(A) 5397F (B) 53977 (C) 539F (D) 5397

### Riešenie

Osmičková sústava má cifry 0 - 7. Každá hodnota takejto sústavy sa dá vyjadriť pomocou troch cifier dvojkovej sústavy. 0 - 000; 1 - 001; 2 - 010; 3 - 011; 4 - 100; 5 - 101; 6 - 110; 7 - 111. Takže najskôr číslo prevedieme do dvojkovej sústavy, a potom takto získané číslo prevedieme do šestnástkovej sústavy.

$1234567_{10} = 001\ 010\ 011\ 100\ 101\ 110\ 111_{2}$

Teraz toto číslo v dvojkovej sústave preskupíme do štvoric a prevedieme do šestnástkovej sústavy.

$0101\ 0011\ 1001\ 0111\ 0111_{2} = 53977_{16}$

Správna odpoveď je (B).

### Úloha 2

Zápis čísla 11C v číselnej pozičnej sústave s najmenším možným základom predstavuje v desiatkovej sústave číslo:

(A) 155. (B) 194. (C) 195. (D) 284.

### Riešenie

V čísle sa vyskytuje cifra "C", ktorá má v desiatkovej sústave hodnotu 12. Najmenší možný základ tohto čísla môže byť teda 13.

$11C_{13} = (1 \cdot 13^2 + 1 \cdot 13^1 + C \cdot 13^0)_{10} = (1 \cdot 12^2 + 1 \cdot 12^1 + 12 \cdot 12^0)_{10} = (169 + 13 + 12)_{10} = 194_{10}$

Správna odpoveď je (B).

### b. Popíšte a vysvetlite činnosť osobného počítača po zapnutí.

Pri startu počítača se jako první provede **reset** (signál propojující všechny obvody). Dostane se tak do výchozího stavu.

Všechny procesory jsou poté v 16bitovém režimu (z historických důvodů, první procesory v [IBM PC](#) byly 16bitové). To znamená, že vidí jen 1 MB paměti a mají omezené schopnosti. Je úkolem SW, který se spustí, aby ho přepnul do režimu 32bit (nebo 64bit).

[Procesory](#) jsou navrženy tak, že po resetu okamžitě začnou provádět instrukce na adrese 0xFFFF0. Na tuto adresu se umísťuje [BIOS](#). Vlastní BIOS mají i některá doplňující zařízení (např. grafická karta, síťová karta, řadiče pro disky atd.). BIOS je najde a spustí jejich inicializační kód (jejich umístění a struktura je pevně daná).

### Činnost BIOSu při startu:

- detekce a inicializace HW – uvede HW do výchozího stavu, spustí se [POST](#)
- aktivace [operačního systému](#)

## Zavedení OS

Operační systém je možné zavést z různých zdrojů:

- [pevný disk](#)

- [floppy disk](#)
- [CD/DVD](#)
- [LAN](#)
- [USB mass storage](#) (flash disk, externí pevný disk...)

## Postup startu OS z diskety

Start OS z diskety je podobný jako z pevného disku, rozdíl je v tom, že pevný disk může mít více oddílů. Ke každému oddílu se pak přistupuje stejně jako k disketě.

### Postup:

- zavedení boot sektoru z diskety do RAM – prvních 512 bytů na disketě
- skočí se na začátek, kde je kód zavaděče OS (specifický pro různé OS)
- nakopírování jádra OS do paměti
- skočí se na začátek jádra

### Postup startu jádra OS:

- převzetí kontroly nad PC, přepnutí procesoru do 32/64bitového režimu
- aktivace [privilegovaného režimu](#), který si pro sebe zabere jádro (nikdo jiný ho nemůže používat)
- detekce a inicializace I/O zařízení
- připojení [souborového systému](#)
- spuštění prvního programu ([Unixové systémy](#) – [Init](#))
- provedení startovací sekvence (startovací skripty, provádí je již init)

## Postup startu z disku

Již existoval funkční start z diskety, při přechodu na pevné disky přišla myšlenka využití "několik disket na disku", proto byla přidána další vrstva ([MBR](#) + [oddíly](#)). Do BIOSu stačilo doprogramovat jen malý dodatek.

Podmínka: BIOS musí umět číst z pevného disku → každý BIOS umí číst z [PATA](#) disku (na doplňujících kartách [SCSI](#), [SATA](#) je rozšíření BIOSu)

Popis logických oddílů disku je uložen v tabulce MBR. Vnitřek oddílu disku má stejnou strukturu jako disketa. Místo MBR se může používat např. [GRUB](#), [LILO](#), [GPT](#)...

### Postup:

- BIOS načte MBR z disku do paměti a skočí na začátek
- kód MBR vybere aktivní oddíl, ke kterému se dále přistupuje jako k disketě

c.Uved'te príklady zlepšenia kvality života prostredníctvom informačných technológií u ľudí s rôznymi (zdravotnými a sociálnymi) znevýhodneniami

## 20. Úloha č.1: Polia – dvojrozmerné celočíselné pole

Vysvetlite pojmy: dvojrozmerné pole, prvok poľa, indexy prvku. Akým spôsobom deklarujeme celočíselné pole?

*Príklad*

Definuj pole o 10 x 10 prvkoch celočíselného dátového typu. Naplň toto pole vo vnorenom cykle výsledkami násobenia celých čísel od 1 po 10. Vypíš toto pole na obrazovke ako tabuľku malej násobilky od 1 do 10 (10 riadkov x 10 stĺpcov).

### Úloha č.2:

- a) Otvorte súbory *hruska.bmp* a *jablka.wmf* v prehliadači obrázkov a vysvetlite k akým typom grafických formátov patria. Pomocou dostupných nástrojov programu demonštrujte rozdiely medzi týmito grafickými formátmi.

*Príklad:*

Minimalizujte veľkosť súboru s obrázkom *hruska.bmp* rôznymi spôsobmi pomocou dostupného softvérového vybavenia. Výsledok každého pokusu uložte do samostatného súboru. Rozoberte výhody a nevýhody jednotlivých spôsobov.

- b) Ukážte možnosti vzdelávania sa cez Internet. Vysvetlite výhody a nevýhody e-learningu a dištančného vzdelávania.

Multimediálne vzdelávacie programy, ktoré sa označujú ako e-learning sa považujú za rýchlejšie, lacnejšie a efektívnejšie. E-learning obvykle znamená možnosť poskytovať viac výcvikov a vzdelávania väčšiemu množstvu ľudí bez rastu dodatočných nákladov a bez problémov s organizovaním tradičných foriem vzdelávania. Pre e-learning sa najviac hodia tri typy tréningov. Okrem výcviku nových zamestnancov je tento druh výučby vhodný, ak je potrebné rozšíriť znalosti veľkej skupiny ľudí, ktorá je rozptýlená. Aj to je dôvod, prečo je e-learning alebo učenie on-line príťažlivé pre globálne pôsobiace firmy. Ako potvrdzujú skúsenosti podnikov v Európe a v USA, na elektronické učenie sa musí firma náležite pripraviť. E-learning je totiž efektívny, ak sa využíva vo firme dlhodobo, vtedy je táto forma štúdia lacná a viac dostupná. Pre niektoré firmy znamená e-learning aj vstupné technologické investície - počítače vybavené zvukovými kartami s dostatočnou výkonnosťou, slúchadlami pre každého účastníka.

Výhodou e-learningu je aj jeho prispôbenie sa prevádzkovým podmienkam firmy a pracovným úlohám jednotlivých pracovníkov. Umožňuje tiež lacnú realizáciu opakovaných školení a retrainingov, zvyšuje rýchlosť a efektívnosť zavádzaných procesov a podporuje tvorbu firemnej kultúry. Pri využívaní e-learningu sa pracovník bude čoraz viac meniť z prijímateľa informácií a vzdelávacích produktov na aktívneho vyhľadávača. Multimediálne vzdelávacie kurzy môžu byť dostupné cez komunikačné siete, čo umožňuje študentom získavať aktuálne vzdelávacie obsahy z viacerých zdrojov, v akomkoľvek čase a z akéhokoľvek miesta. Tým sa dá meniť čiastočne alebo v niektorých prípadoch aj celkom charakter poskytovania služby vzdelávania na službu e-vzdelávania. Výhody elektronického vzdelávania zo strany študujúceho, vyučujúceho a vzdelávacej inštitúcie:

- dostupnosť – štúdium je možné nezávisle na vzdialenosti medzi miestom bydliska a umiestnením vzdelávacej inštitúcie,

- časová a priestorová asynchrónnosť – študent sa venuje štúdiu v čase, keď mu to najlepšie vyhovuje a na ktoromkoľvek mieste. Prevažnú časť štúdia prispôsobuje svojim potrebám, riadi si svoj vlastný proces učenia sa. V anglickej literatúre je tento typ učenia označovaný ako self-directed learning. Študenti sú zodpovední sami za seba, elektronické vzdelávanie je vhodným typom pri vzdelávaní dospelých ľudí popri zamestnaní, pretože nie sú viazaní časom a ani konkrétnym miestom,
- dobre rozvinutý servis – poradenstvo, vydávanie, zasielanie alebo požičiavanie študijných materiálov organizovaných tak, aby čo najlepšie umožňovali samostatné štúdium,
- nízke, takmer nulové požiadavky na ubytovanie a stravu alebo cestovné,
- nízke nároky na počty učební a s tým súvisiace prevádzkové náklady,
- nízke pracovné úväzky tútorov,
- väčšia zaangažovanosť študenta do vzdelávacieho procesu,
- ľahšia dostupnosť pre hendikepovaných študentov.

Nevýhody elektronického vzdelávania zo strany študujúceho, vyučujúceho a vzdelávacej inštitúcie:

- založenie a financovanie organizačného pracoviska
  - nutnosť zaškolenia pedagogických pracovníkov pre novú technológiu
  - na tvorbu kvalitných študijných materiálov sú kladené vysoké nároky
  - preverovanie vedomostí študentov – študent môže dávať úlohy vypracovávať niekomu inému
  - náročnosť na technické zabezpečenie
  - nedostatočná dôveryhodnosť všetkých materiálov na internete
  - vysoké náklady na technické vybavenie inštitúcie a prípravu alebo zakúpenie softwarových programov
- Obsah vyučovacieho predmetu ponúkaného v e-vzdelávaní je špecifický a vychádza z požiadaviek a potrieb študentov. Text určený pre e-vzdelávanie nemôže obsiahnuť celý obsah predmetu, preto je sústredený iba na nevyhnutný obsah.

Elektronický text má svoje špecifiká. Takéto študijné programy by mali byť koncipované ako flexibilné (pružné prispôsobenie sa formy a v určitej miere aj obsahu výučby tak, aby stanovený obsah, kvalita i ciele výučby boli dosiahnuté čo najefektívnejším spôsobom), otvorené (dostupnosť pre rôzne cieľové skupiny, v rôznom priestore a čase, pre rôzne možnosti integrácie s inými študijnými programami a pod.), modulárne (je snaha vytvárať relatívne malé moduly, s minimálnymi závislosťami, s jasne vymedzeným vstupom aj výstupom, pri tvorbe modulov je použitá jednotná metodika i štruktúra, zahŕňajúca základné parametre modulu – názov, kód, anotáciu, učebné ciele,...), on-line (využitie možností nových informačných a komunikačných technológií, najmä internetu a www) a dištančné (študijné materiály realizované metodikou dištančného vzdelávania, ktorá umožní ich univerzálne a flexibilné využitie nielen pre predpokladanú cieľovú skupinu, ale aj efektívnu adaptáciu pre iné cieľové skupiny).

Metodika prípravy učebných materiálov pre dištančné vzdelávanie

- vkladany učebný materiál musí byť čo najviac členený, odporúča sa 1 kapitola a 2 úrovne podkapitol,
- vhodná implementácia mimotextových zložiek, predovšetkým orientačného aparátu,
- množstvo textu na stránke/obrazovke menšie než v tlačenej študijnej materiále,

- pri tvorbe materiálu je potrebné dodržiavať dohodnuté značky,
- text musí byť vyjadrený jasnými formuláciami v jednoduchých vetách,
- dobrá „čitateľnosť“,

Čo sa týka obsahu:

- v texte je vhodné umiestniť odkazy na materiály na inom mieste, resp. iných www serveroch odkazy na index kľúčových slov,
- jasne stanovené dlhodobé i špecifické ciele a ich dôsledná korelácia s obsahom,
- využívanie a nadväzovanie na predchádzajúce vedomosti frekventantov,
- zohľadnenie rôznych potrieb frekventantov,
- animácia a simulácia dejov,
- dostatok praktických zadaní na precvičenie a upevnenie osvojeného,
- dostatok vysvetľujúcich príkladov,
- otázky a cvičenia na sebahodnotenie,
- zaujímavosť a prítťaživosť materiálu (obsah i forma).

Dôvody na implementáciu e-vzdelávania do vyučovacieho procesu:

- nárast množstva informácií, v ktorých je potrebné sa zorientovať,
- potreba rozsiahlejšieho vzdelávania, aké je možné získať v školských laviciach,
- aktívna úloha študenta vo vzdelávacom procese,
- rýchlejšie a kvalitnejšie získavanie informácií,
- individuálny časový harmonogram výučby,
- možnosť zvýšenia počtu študentov bez nárokov na rozširovanie výučbových priestorov,
- zníženie potreby zabezpečovania ubytovania, zníženie cestovných nákladov,
- konkurenčný tlak zahraničných univerzít, najmä univerzít v USA

**c) Definujte pojem malvér a popíšte typy softvéru, ktoré do tejto kategórie patria.**

**Malware** (skratka z [anglického](#) *malicious software*) alebo **malvér** <sup>[1]</sup> je všeobecné označenie škodlivého [softvéru](#).

- patria sem napríklad [vírusy](#), [trójske kone](#), [spyware](#) a [adware](#). Malware sa do počítača v dnešnej dobe dostáva zvyčajne cez [Internet](#), hlavne pri prezeraní škodlivých stránok s nie dobre zabezpečeným systémom. Vírus - zvyčajne má dve funkcie, prvou z nich je reprodukcia (rozmnožovanie), na čo využíva spustiteľné časti programov (zväčša súbory .exe, .com, .sys, .dll, ...), prípadne dokumenty.

Okrem šírenia sa vírus zväčša aj prezentuje nejakou formou, napríklad môže vypisovať rôzne hlášky, vytvárať na obrazovke grafické efekty alebo aj zahrať melódiu.

Vírusy mávajú aj deštruktívnu formu prejavu - napadnutie systémových súborov operačného systému čoho dôsledkom môže byť čiastočné alebo celkové znefunkčnenie daného systému, zmazanie súborov alebo adresárov, menenie obsahu súborov,

šifrovanie dát, prípadne i poškodenie hardwaru (prepísaním BIOSu na základnej doske).

Ďalším typom vírusu je vírus označovaný ako stealth, čo v praxi znamená, že sa maskuje pred možným vypátraním, rezidentná časť takéhoto vírusu neustále sleduje činnosť počítača a na každú požiadavku o zistenie hodnôt, ktoré by mohli viesť k jeho odhaleniu reaguje vrátením pôvodných hodnôt, ktoré sú bez náznaku infiltrácie.

Ďalším druhom vírusov sú vírusy polymorfné, ich základnou filozofiou je, že žiadna kópia tela vírusu nie je totožná s inou kópiou, čím vírus zabezpečí horšie podmienky pre jeho detekciu užívateľom/antivírusovým programom.

- Trójsky kôň – sú to programy šíriace sa ako vírusy, na rozdiel od vírusu trójsky kôň nepotrebuje hostiteľa a väčšinou si ho aktivuje sám užívateľ svojou nevedomosťou (trójske kone sa zväčša prezentujú ako užitočné utility). Trójsky kôň môže vykonávať funkciu ako daný program, nie je to však pravidlom. Na pozadí vykonáva činnosť nesúvisiacu s aplikáciou – vymazávanie súborov, formátovanie diskov, zasielanie informácií o užívateľovi prostredníctvom internetu. Dosť rozšírená je tzv. backdoor forma trójskeho koňa, ktorá pridáva k vlastnostiam trójskeho koňa ešte niečo navyše – možnosť autora programu týmto spôsobom ovládať vzdialený počítač, robiť rôzne úkony so súbormi a aplikáciami.
- Červ - typ vírusu ktorý sa šíri pomocou elektronickej pošty, na rozdiel od klasických súborových a boot vírusov sa väčšina červov nereprodukuje v takom množstve, veľká časť sa dokonca nereprodukuje vôbec a na infiltrácie využíva pôvodnú vzorku. Na svoje šírenie využíva všetky možné komunikačné kanály, hlavne elektronickú poštu a intranetovú sieť.
- Makrovírus – sú to vlastne makrá, alebo súhrn makier, ktoré sú vytvorené pomocou VBA (Visual Basic for Applications) obsiahnutom v Microsoft Office. Telo vírusu je integrované do makra, ktoré slúžia na tvorbu interaktívnych formulárov pod textovými editormi a tabuľkovými procesormi. Makrovírusy sú schopné poškodiť systém alebo jeho súčasť vďaka tomu, že VBA je úzko späté s operačným systémom. Hlavným znakom toho, že je počítač napadnutý makrovírusom je nemožnosť ukladať súbory alebo spúšťať jednotlivé aplikácie, ktoré sú súčasťou Microsoft Office.

## 21. 1: Základy – príkaz výstupu

Vysvetlite pojmy: príkaz výstupu, textová obrazovka. Ako dokážeme výpis presunúť o pozíciu tabulátora doprava? Ako dokážeme výpis presunúť do nového riadku.

### *Príklad*

Zostavte program, ktorý načíta od užívateľa číslo N a vypíše na obrazovku počítača v textovom režime N-krát slovíčko AHOJ a to tak, aby každé nasledujúce slovíčko, bolo posunuté o jeden riadok dolu a o jednu medzeru vpravo (podľa obr.)

Pozmeňte program tak, aby sa každé ďalšie slovíčko posúvalo o X znakov vpravo a Y riadkov dolu, kde X a Y zadáte na vstupe.

### Úloha č.2:

- a) Definujte pojmy: rastrová a vektorová grafika a popíšte ich výhody a nevýhody. Vymenujte známe grafické editory a aspoň 3 základné formáty rastrových obrázkov, stručne ich charakterizujte.

*Príklad:*

Máme rastrový obrázok s parametrami: 800x600, 256 farieb. Vypočítajte, aký veľký bude neskomprimovaný súbor obrázka?

## počítačová grafika

Podľa spôsobu vzniku a záznamu **grafickej informácie** - obrázka v digitálnom tvare, poznáme **rastrovú a vektorovú grafiku** a rozdeľujeme grafické editory na **rastrové** (bitmapové) a **vektorové** (objektové).

### Rastrová grafika

Rastrová grafika vychádza z vyjadrenia obrázka ako zoskupenia bodov, ktoré označujeme **pixely** (picture element - prvok obrázka, skratka px; nesprávne bod obrázka). Vzhľad každého pixelu je zakódovaný a obrázok je teda vyjadrený ako sada zakódovaných pixelov. Takúto sadu zakódovaných pixelov nazývame **bitová mapa** (bmp) alebo jednoducho rastrový obrázok.

Zložitosť kódovania pixelov je daná ich farebnou hĺbkou. Pri čiernobielym obrázku je možné každý pixel opísať (zakódovať) jediným bitom (podľa poznatku, že  $n$  bitov umožňuje zakódovať  $2^n$  rôznych hodnôt, 1bit umožňuje zakódovať  $2^1 = 2$  farby). Pri čiernobielych fotografiách sa používa 256 rôznych odtieňov šedej farby, teda farba každého pixelu je zakódovaná vo ôsmich bitoch (1B/pixel). V prípade farebných obrázkov sa používa ešte zložitejšie kódovanie pixelov (pozri nižšie kódovanie farieb, farebné modely).

Súbor obsahujúci rastrový obrázok obsahuje teda zakódované informácie o farbách jednotlivých elementov tvoriacich daný obrázok.

### Vektorová grafika

Pri vektorovej grafike je **obrázok zložený z objektov skladajúcich sa z kriviek**, kde jednotlivé objekty sú popísané matematicky. Objekt je popísaný počiatočným bodom, smerom, dĺžkou,... a jeho vlastnosťami ako farbou obrysovej čiary, farbou výplne, priehľadnosťou, tieňovaním, poradím, v akom bude vykreslený,...

Krivky spájajú jednotlivé kotvové body a môžu mať definovanú výplň (farebnú plochu alebo farebný prechod). Tieto čiary sa nazývajú Bézierove krivky.

Francúzsky matematik Pierre Bézier vyvinul metódu, pomocou ktorej sme schopní popísať pomocou štyroch bodov ľubovoľný úsek krivky. Krivka je popísaná pomocou dvoch krajných bodov (kotvové body) a dvoch bodov, ktoré určujú tvar krivky (kontrolné body). Spojnica medzi kontrolným bodom a kotvovým bodom je dotyčnicou k popisovanej krivke. Základom vektorovej grafiky je matematika. Súbor obsahujúci vektorový obrázok sa skladá z definície súradnicového systému,

z rovníc (s konkrétnymi hodnotami parametrov) opisujúcich jednotlivé objekty a z vlastností objektov – obrysovej čiary, výplne, tieňa, priehľadnosti, poradia vykreslenia,...



### Rastrová grafika

Na obrázok sa pozerá ako na sieť (raster) veľmi malých štvorcov - pixelov, uložených v pevných riadkoch a stĺpcoch. Rozmer každého obrázka je pre počítač počet pixelov na šírku krát počet pixelov na výšku. Pre každý pixel (štvorček) je nutné okrem polohy (riadok a stĺpec) zakódovať aj farbu, resp. ďalšie parametre napr. priehľadnosť. Výhodou tejto metódy je

možnosť relatívne presného opisu aj veľmi nesúrodého a zložitého objektu, nevýhodou je výsledná veľkosť súboru. Čím je vyššia presnosť zachytenia vernosti originálu, tým je potrebné väčšie rozlíšenie a tým je aj väčší výsledný súbor. Pre reprodukciu rastrovej grafiky musí byť vstupné rozlíšenie obrázka ekvivalentné výstupnému rozlíšeniu reprodukčného zariadenia (napr. tlačiarne), inak dochádza ku strate kvality výstupu. Príkladmi formátov rastrovej grafiky sú BMP, GIF, JPEG, PNG, TIFF. Vektorová grafika

Na obrázok sa pozerá ako na zoskupenie objektov alebo ich častí, ktoré sa dajú popísať pomocou matematických vzorcov a funkcií. Tieto objekty majú svoje vlastnosti napr. - polohu na obrázku, veľkosť, farbu, priehľadnosť povrchu, lesklosť povrchu a pod. Tieto vlastnosti sú vstupnými parametrami (vektormi) matematických vzorcov a funkcií, pomocou ktorých sa objekty opíšu (vykreslia). Vstupné rozlíšenie je dané len zložitostou a detailnosťou matematického opisu, výstupné rozlíšenie nie je tak závislé na vstupnom rozlíšení, ako je tomu u rastrovej grafiky. Pre jednoduché objekty - kružnica, štvorec, trojuholník, krivka alebo pre zloženiny týchto objektov a pod. je možný jednoznačný a stopercentný zápis pomocou matematických rovníc. V takomto prípade je možné z vektorovej grafiky vytvoriť ľubovoľnú zväčšeninu bez straty kvality výstupu. Príkladom sú formáty CDR (CorelDraw), ZMF (Zoner Callisto), DXF (AutoCAD),... Väčšina vektorových grafických editorov umožňuje export vektorových obrázkov aj do rastrových formátov BMP, JPG, GIF,...

#### **Porovnanie rastrovej a vektorovej grafiky:**

Vektorová grafika má proti rastrovej grafike niektoré výhody:

- je možné ľubovoľne zmenšovať alebo zväčšovať obrázok bez straty kvality
- je možné pracovať s každým objektom v obrázku oddelene
- výsledná pamäťová náročnosť obrázka je obvykle omnoho menšia ako pri rastrovej grafike
- Nevýhody vektorovej grafiky
- oproti rastrovej grafike spravidla zložitejšie vytvorenie obrázka; v rastrovej grafike možno obrázok ľahko získať pomocou fotografie
- ak prekročí zložitost' grafického objektu určitú hranicu, začne byť vektorová grafika náročnejšia na pamäť, procesor, veľkosť disku ako grafika bitmapová.

## Použitie

Vektorová grafika sa používa najmä na tvorbu ilustrácií, diagramov a počítačových animácií.

Na prácu s vektorovou grafikou sa používajú vektorové editory Corel Draw, Zoner Callisto, Adobe Illustrator,... Použitie rastrovej grafiky je uvedené pri grafických formátoch BMP, GIF, JPG, PNG,...

## **B. Porovnajete základné vlastnosti dvoch operačných systémov (napr. Linux- Windows) z hľadiska používateľa.**

Windows verus Linux – téma, ktorej sa venovali a stále venujú mnohí autori v tisíckach svojich článkov. Pre túto kontroverznú tému nemôžu mnohí počítačoví nadšenci spávať, neustále premýšľajú a stále sa pýtajú tú istú otázku: „Ktorý z uvedených operačných systémov (OS) je kvalitnejší, rýchlejší, lepší?“

Bohužiaľ, odpoveď na túto otázku neexistuje. Autori článkov hodnotiacich rozdiely, výhody a nevýhody jednotlivých OS hodnotia a porovnávajú neporovnateľné. V tomto článku si



vysvetlíme prečo nemožno Linux porovnávať s Windowsom. Urobíme tak z pohľadu Linuxu ako vážneho konkurenta Windowsu. Našou zásadou bude, že Linux ani Windows nebudeme nijako vyzdvihovať ani podhodnocovať.

### **Nemožno porovnávať neporovnateľné**

Základné dôvody prečo nemôžeme jednoducho porovnávať Linux s Windowsom sú nasledujúce:

1. OS Windows je vydávaný (released) vo forme verzií, ktoré sú ďalej členené podľa tzv. zostavení (builds). V Linuxe hovoríme o tzv. verzionovaných distribúciach obsahujúcich príslušnú verziu jadra (kernel). Ktorú konkrétnu distribúciu a ktoré konkrétne jadro máme vybrať aby sme mohli Linux objektívne porovnať s Windowsom?
2. Linux je rovnako ako Windows neustále (niekedy aj denne) upravovaný (update) a doplňaný (upgrade) o rôzne nástroje (utilities), záplaty (patches), opravné balíky, nové verzie jadra a nadstavbových aplikácií. Aký konkrétny stav oboch systémov máme vybrať aby sme ich mohli porovnať?
3. Linux a Windows majú úplne odlišné jadrá, ktoré pristupujú k hardvéru (HW) počítača úplne iným spôsobom. Ako môžeme zaručiť, že pri porovnaní týchto OS bude hodnotený proces využívať HW zdroje úplne rovnako?
4. Linux rovnako ako Windows potrebujú pre svoju činnosť ovládače HW zariadení. Tie pripravujú výrobcovia konkrétneho HW osobitne pre každý OS a jeho konkrétnu verziu. Máme istotu, že ten istý ovládač pracuje úplne rovnako v jednom aj druhom OS?
5. Linux aj Windows dokážu svoju prácu buď automaticky, alebo používateľsky optimalizovať tak, aby bol ich výkon a stabilita lepšia v porovnaní s neoptimalizovanou, resp. štandardnou (default) konfiguráciou. Aj nepatrná zmena niektorého z optimalizačných parametrov môže mať výrazný vplyv na výsledný výkon systému. Môžeme objektívne vyhlásiť, že takto upravený systém je lepší ako konkurenčný?

Je nepopierateľné, že popisované OS majú tisíce výhod aj tisíce nevýhod. Obidva systémy sú porovnateľne stabilné, pričom miera ich stability je úmerná viacerým menším i väčším chybám, ktorým sa nevyhne žiaden operačný systém.

Nie je objektívne argumentovať, že Linux je zadarmo. Nie je totiž pravda, že všetko čo sa týka Linuxu je úplne zadarmo. Používateľ Linuxu, ktorý ho využíva doma na prípravu bakalárskej práce sa v tomto prípade výrazne odlišuje od operátora pracujúceho v jadrovej elektrárni, ktorý si v žiadnom prípade nemôže dovoliť používať Linux bez akejkoľvek podpory (zväčša platenej).

Aj keď Linux nemôžeme porovnávať s Windowsom, stále existuje určitý priestor na popísanie odlišností, ktoré ale v žiadnom prípade nepoukazujú na výhody či nevýhody používania konkrétneho systému. Takisto je relevantné uviesť, čo možno od jednotlivých OS očakávať v súvislosti s popísanými odlišnosťami.

### **Hardvér**

Začneme úplne od začiatku a to obstaraním (kúpou) hardvéru. Linux možno v súčasnosti úplne rovnako ako Windows nainštalovať na takmer každý, resp. na všetky moderné počítače. V minulosti to nebolo až také jednoznačné. Problémom mohli byť ovládače niektorých bežne nepoužívaných zariadení. Konkrétnym príkladom bol interný modem nemenovaného výrobcu, ktorého ovládač určený pre Linux vôbec neexistoval. Nie je vylúčené, že niektoré špecializované zariadenia môžu mať podobné problémy aj v súčasnosti. Vždy je teda dobré si pred samotnou kúpou skontrolovať, či je daný HW kompatibilný s OS Linux a či je možné

získať potrebné ovládače. To platí nielen pre kategóriu osobných počítačov, ale aj pre servery, u ktorých je však podpora Linuxu zväčša bezproblémová.

V minulosti sa vyskytovali problémy s podporou určitého HW konkrétnymi Linuxovými distribúciami. Moderné distribúcie však takéto problémy nepoznajú. Používateľ už nemusí riešiť problémy so zlou podporou HW a konkrétnu distribúciu si vyberá zväčša na základe kvalít jej používateľského rozhrania.

## **Obstaranie a inštalácia**

### *1. Obstaranie*

Prvým krokom k používaniu Linuxu je jeho obstaranie a inštalácia. Tu sa prejavuje prvý rozdiel oproti Windowsu a síce, že Linux je zadarmo. Znamená to, že ktokoľvek si môže bezplatne stiahnuť kompletne inštalčné médium z internetu, resp. toto médium dokáže získať prostredníctvom pošty, alebo kúpou v knihkupectvách, kde je často prílohou používateľskej príručky. Nezriedka je kompletná inštalčka prílohou niektorého z IT časopisov (vrátane PCRevue).

### *2. Inštalácia*

Samotná inštalácia sa už od inštalácie Windowsu výrazne neodlišuje. Inštaláciu moderného Linuxu zvládne aj úplný začiatočník bez akýchkoľvek IT vedomostí. Dávno nie je pravdou, že na inštaláciu Linuxu potrebujeme vysoké odborné vedomosti či priam účasť linuxového profesionála.

Inštalácia prebieha v grafickom prostredí kde používateľ jednoducho kliká myškou a väčšinu činností necháva na inštalátor. Linux teda dokáže nainštalovať ktokoľvek kto vloží inštalčné médium do mechaniky a prekliká sa na koniec inštalácie pomocou tlačidla „ďalej“.

Výhodou Linuxu je, že súčasne s inštaláciou OS si používateľ môže na svoj počítač nainštalovať aj množstvo doplnkových aplikácií. Príkladom môže byť kompletný kancelársky balík, nesmierne množstvo vývojových (development) nástrojov, grafických, zvukových, multimediálnych, sieťových a systémových aplikácií. Za zmienku stoja aj najrôznejšie aplikačné, súborové, tlačové, poštové a iné servery. Pritom nesmieme zabudnúť na fakt, že všetky tieto aplikácie sú súčasťou inštalčného média a sú zadarmo.

## **Ovládače**

Po inštalácii OS nás zaujíma stav ovládačov HW zariadení. V tomto smere sa Linux za posledné roky posunul míľovými krokmi vpred. V súčasnosti nie je žiadnou výnimkou, že po inštalácii OS sú všetky ovládače (moduly jadra) správne nainštalované a Linux dokáže spravovať všetky HW komponenty počítača.

Z osobných skúseností môžem povedať, že moderný Linux (konkrétne distribúcie Ubuntu a Fedora) dokáže ihneď po inštalácii bez problémov pracovať na moderných základných doskách osadených modernými CPU, používať RAM, obhospodarovať HDD, FDD, USB zariadenia, grafické karty, zvukové karty, sieťové karty a pracovať s mnohými ďalšími bežne vyrábanými HW zariadeniami.

Linux by tu mohol mať voči Windowsu navrch, avšak nie až tak jednoznačne. Konkrétnym príkladom môže byť ovládač grafickej karty. Po inštalácii OS je používateľovi ponúknuté iba priemerné využitie jeho grafických kapacít. Pre získanie požadovaného grafického výkonu je nutné dodatočne nainštalovať ovládač grafickej karty vydaný jej výrobcom.

## **Inštalácia aplikácií**

V tomto smere môžeme povedať, že Windows má voči Linuxu istú prednosť. Vec sa má tak, že pokiaľ chceme vo Windowse nainštalovať konkrétnu aplikáciu, jej inštalčný súbor získame

(napr. z internetu) vo forme binárneho súboru (s príponou .exe), ktorého spustením sa aplikácia bez ďalších zbytočných otázok nainštaluje. Je pritom jedno či pri inštalácii máme prístup na internet, alebo nie.

V Linuxe je situácia odlišná. Áno, skúsenejší používatelia okamžite navrhnú použiť správcu balíčkov (package manager), ktorý urobí všetku prácu za nás. Čo však v prípade, ak nemáme prístup na internet? Problém je v tom, že Linux majoritne nepozná inštalčné súbory, ktoré by boli reprezentované jediným spustiteľným súborom. Inštalčné súbory, tzv. balíčky (packages) spravidla neobsahujú všetko potrebné na kompletnú inštaláciu danej aplikácie. Balíček predpokladá, že potrebné komponenty (utility, knižnice...) sa v našom systéme nachádzajú, resp. že ich manažér balíčkov doplní v priebehu inštalácie. Faktom však zostáva, že pokiaľ nemáme priame pripojenie na internet z počítača, na ktorom chceme balíček nainštalovať, môžeme sa dostať do zložitej situácie.

Linux pri inštalácii preveruje tzv. závislosti (dependencies) balíčkov. Manažér balíčkov tieto závislosti rieši automaticky, avšak musí mať prístup na internet, resp. do tzv. repozitáru (repository). Repozitár s potrebnými balíčkami síce môžeme byť uložený aj na off-line médiu, avšak takýto prístup nie je obvyklý. Navyše nás môže prekvapiť vznik tzv. zacyklenia. Ide o prípady kedy balíček A požaduje prítomnosť balíčka B a zároveň balíček B požaduje pred jeho inštaláciou existenciu balíčka A. V takom prípade jeden z balíčkov nainštalujeme tzv. na tvrdo (force) a druhý sa už nainštaluje bez ďalších pripomienok. V takýchto prípadoch sa však bez širších vedomostí o práci v Linuxe nezaobídeme.

### **Používanie prostredia**

Už v úvode sme si povedali, že moderný Linux má moderné grafické používateľské rozhranie (GUI – Graphical User Interface). Jeho kvalitu oproti Windowsu ťažko komentovať. Faktom však je, že ani prvé ani druhé rozhranie nie je lepšie ani horšie. V Linuxe dokážeme takmer všetky činnosti vykonať pomocou klikania myškou. Dávno neplatí tvrdenie, že ten, kto chce pracovať v Linuxe, musí pracovať v prastarom príkazovom riadku (command prompt). GUI Linuxu je plne konfigurovateľné, obsahuje rovnaké vizualizačné nástroje ako GUI Windowsu. Je rovnako „pekné“ aj „škaredé“. Nie je pravda, že v ňom nedokážeme robiť veci tak ako vo Windowse. Takisto nie je pravda, že v ňom nedokážeme urobiť to, čo dokážeme vo Windowse.

### *Konzola*

Mnoho používateľov Windowsu zavrhuje príkazový riadok (konzole, terminal) Linuxu a to pre jeho úzke naviazanie na príkazy zadávané prostredníctvom klávesnice v tzv. textovom móde. Vo Windowse nebolo používanie príkazového riadku zaužívané a tento spôsob práce mal vždy akúsi „menejcennú“ pozíciu voči GUI prostrediu. Konzola Linuxu však „vyrastala“ spolu s Linuxom a plní dôležitú úlohu aj napriek tomu, že takmer všetky úlohy možno vykonať priamo v GUI.

Prečo je konzola tak často používaná vedia snáď iba skalní priaznivci Linuxu. Moje osobné skúsenosti s jej používaním sú však takisto kladné. Aj keď máme k dispozícii plnohodnotné GUI, pomocou konzoly predsa len dokážeme urobiť veľa vecí jednoduchšie. Možno sa teraz mnohí z nás čudujú ako je to možné. Odpoveďou môže byť konštatovanie, že pri práci v konzole skutočne „cítíme“, že svoj OS a teda aj svoj počítač, ovládame. Príkazy zadávané v konzole sú bez otázok splnené, ich výstup je možné okamžite vidieť a reagovať naň. V Linuxe nie je bežné, že po spustení aplikácie čakáme na zobrazenie výstupu, na otvorenie okna, resp. systém na nás vôbec nereaguje. Spustený príkaz okamžite informuje o svojej činnosti buď prostredníctvom GUI, alebo minimálne textovým výpisom v konzole.

Linux poskytuje stopercentnú kontrolu spustených procesov. Ktorýkoľvek proces môžeme pozastaviť alebo zrušiť. Je pritom jedno či daný proces prebieha, čaká, neodpovedá, alebo je v

inom nedefinovanom stave. V priebehu mojich 10-tich rokov používania Linuxu mi OS tzv. „zamrzol“ iba 2 či 3 krát tak, že som musel počítač vypnúť (reštartovať) hardvérovým tlačidlom. Vo všetkých ostatných prípadoch som bežiacie procesy vždy kontroloval iba pomocou konzoly a jej príkazov.

### **Základné aplikácie**

Rozsah základných aplikácií ponúkaných Linuxom jednoznačne stavia Linux pred Windows. Síce by sa dalo namietat', že ide o open source aplikácie, ktoré nedosahujú takú kvalitu ako proprietárny softvér, pre bežného používateľa sú však ich drobné nedokonalosti nepodstatné. Nebudeme menovať konkrétne aplikácie, avšak môžeme uviesť, že v Linuxe nájdeme najrôznejšie aplikácie na úpravu obrázkov, zvuku, na prácu s multimediálnymi súbormi, s CD/DVD/USB médiami a podobne. Linux disponuje takmer nekonečným zoznamom vývojových nástrojov, systémových utilít a nástrojov pre prácu v sieťovom prostredí. Výnimkou nie sú rôzne emulátory, virtualizačné nástroje, skôr spomínané softvérové servery a mnoho ďalších menších i väčších aplikácií.

Práca v uvedených aplikáciách je úplne rovnaká ako vo Windowse. Je pravda, že prostredie aplikácií je odlišné a jednotlivé úlohy sa vykonávajú iným spôsobom, výsledný efekt je však rovnaký. To, či je práca v aplikáciách Linuxu komfortnejšia, alebo problematickejšia ako vo Windowse, nebudeme rozoberať. Každý z nás si musí urobiť vlastný názor na základe vlastných skúseností. Niekomu totiž vyhovuje skôr jedno, inému naopak druhé riešenie. Všeobecne môžeme povedať, že výsledky práce s obrázkami, zvukom a multimédiami sú v oboch OS úplne rovnaké. Tie isté súbory môžeme spracovávať v oboch systémoch bez akejkoľvek audiovizuálnej zmeny v ich podaní.

### **Kooperácia**

S najviac viditeľnými odlišnosťami medzi Linuxom a Windowsom sa môžeme stretnúť v reálnom živote pri ich vzájomnej kooperácii. Práve pri súčasnom využívaní oboch OS sa prejavujú ich špecifické vlastnosti a samozrejme aj ich obmedzenia. Najhorším prípadom je ten, pri ktorom je náš Linuxový systém „opustený sám v mori Windowsových strojov“, s ktorými musí nejakým spôsobom komunikovať.

### *Linux doma*

Pokiaľ sa nachádzame v domácom prostredí, našim problémom by mohli byť počítačové hry, vrátane tých, ktoré sa mienime hrať s našimi známymi prostredníctvom siete. Experti budú iste namietat' prečo sa pri hodnotení rozdielov medzi Linuxom a Windowsom zaoberáme takou banalitou akou sú počítačové hry. Omyl páni, počítačové hry dnes hýbu softvérovým priemyslom rovnako ako vysokocenené návrhárske, alebo vedecko-výskumné nástroje. Finančný podiel herného priemyslu nemožno zanedbať. A práve tu bohužiaľ musíme skonštatovať, že Linux nemôže priamo konkurovať Windowsu. Mnohé moderné hry potrebujú na svoju inštaláciu a beh OS Windows a jeho technológiu DirectX. Aj keď je viacero moderných hier platformovo nezávislých, Linux v tomto smere stále zaostáva. Nie je to však vina Linuxu, ale skôr vývojárov hier, ktorí vidia ich väčšie uplatnenie práve vo svete OS Windows. Hry sú tvrdým biznisom a žiadna firma si nenechá ujsť zisk z ich predaja.

### *Linux v práci*

V prípade používania Linuxu v pracovnom prostredí, v ktorom určite nemyslíme na počítačové hry, nás istotne zaujímajú možnosti kooperácie v pracovnom zmysle slova. V tomto smere je situácia nasledujúca:

1. Prítomnosť Linuxového počítača v čisto Windowsovom prostredí musí byť týmto prostredím

akceptovaná.

Týka sa to najmä možnosti prístupovať k sieťovým zdieľaným zdrojom. V minulosti toto nepredstavovalo žiaden problém, súčasnosť však prináša isté obmedzenia a síce v implementácii nových foriem bezpečnosti. V prípade ak máme dosah na aplikované bezpečnostné politiky v našej sieti, môžeme si ich upraviť tak, aby bol náš počítač akceptovaný. V opačnom prípade, kedy nám správca neposkytne dostatok informácií o štruktúre siete a aplikovaných politikách a zároveň tieto politiky priamo vylučujú používanie iného systému ako je OS Windows, nemáme šancu v takomto prostredí prevádzkovať OS Linux.

2. Windowsové prostredie, ktoré akceptuje Linuxový počítač, mu musí umožniť prístupovať k ponúkaným zdrojom.

Pokiaľ môžeme pracovať vo Windowsovom prostredí a toto nám umožní napr. prístup k e-mailovému serveru, v takom prípade pre nás nie je žiaden problém využívať ľubovoľného Linuxového poštového klienta na spracovanie e-mailov.

Prístup na zdieľané zdroje (diskový priestor, tlačiarne, skenery, multifunkčné zariadenia...) bude rovnako bezproblémový. Drobným problémom môže byť identifikácia daného zariadenia v sieti. Jeho meno (často iba DNS záznam), ktoré nám prezradí správca, alebo náš kolega pracujúci vo Windowse, nie vždy postačuje na nadviazanie spojenia. Linux síce obsahuje niekoľko nástrojov, pomocou ktorých sme schopní získať ďalšie nevyhnutné informácie, ich použitie si však vyžaduje hlbšie vedomosti v danej problematike.

3. Spracovanie údajov nie je v oboch prostrediach stopercentne rovnaké.

Pokiaľ doma vytvoríme textový dokument, vytlačíme ho, alebo pošleme kamarátovi elektronickou poštou, nebude nás trápiť jeho mierne odlišný vzhľad oproti úplne rovnakému dokumentu, ktorý by sme vytvorili vo Windowse. Firemné prostredie však často vyžaduje od svojich zamestnancov, aby títo dodržiavali stanovené formáty, formuláre, šablóny a podobne. Práve striktné dodržiavanie firemných štandardov je najväčším kameňom úrazu multiplatformového spracovania textových, tabuľkových, prezentačných a iných „kancelárskych“ údajov. V tomto smere je kandidátom na víťaza formát PDF. PDF dokumenty môžeme bez akýchkoľvek rozdielov vytvárať a následne prezerat' v ktoromkoľvek OS. Bohužiaľ ostatné kancelárske (office) formáty takmer vždy obsahujú rozdiely. Nebudeme hovoriť o problémoch, ale o akceptovateľných rozdieloch. Základným faktom je, že ten istý dokument spracovaný v Microsoft Office nebude úplne totožný s dokumentom spracovaným v OpenOffice. To sa týka všetkých formátov, ktoré v kancelárskom balíku spracúvame. Zamestnanec, ktorý je povinný dodržiavať firemné štandardy je nútený rozdiely identifikovať a tomu prispôsobiť svoju prácu. Jedným z možných riešení je inštalácia virtualizačného nástroja, pomocou ktorého do Linuxu nainštalujeme virtuálny OS Windows a dotknuté dokumenty budeme spracovávať akoby sme pracovali priamo na fyzickom Windowse. Takýto prístup však nenaplní naše požiadavky na prácu výhradne s open source aplikáciami. Preto nám nezostáva nič iné iba čo najvernejšie prispôsobiť naše výstupy tak, aby spĺňali požadované kritériá, alebo sa im aspoň v maximálnej miere priblížili.

### **Prenos údajov**

Prenos údajov medzi Linuxom a Windowsom nepredstavuje žiaden platformový rozdiel. Linux síce používa niektoré špeciálne formáty súborov, všetky však dokáže konvertovať do podoby čitateľnej Windowsom. To sa týka nie len kancelárskych, grafických, zvukových a multimediálnych formátov, ale aj formátov využívaných pri tvorbe archívov, sieťových prenosoch, alebo iných špecializovaných formátov.

V prípade ak predsa len narazíme na taký súborový formát, ktorý v jednom či druhom OS nedokážeme štandardnými nástrojmi spracovať, na internete určite nájdeme konvertor, ktorý nám to umožní. V bežnej praxi neexistuje taký formát údajov, pri ktorom by sme boli obmedzovaní použitím konkrétneho OS. Zároveň možno konštatovať, že súborové systémy oboch OS sú schopné vzájomne spolupracovať – čítať a prenášať údaje.

Linux aj Windows sú v súčasnosti dva operačné systémy nachádzajúce sa v pomyselnom vývojovom rebríčku na úplne rovnakom stupni. Je iba na nás, ktorý OS si vyberieme. Finančné hľadisko v prípade Linuxu nemusí byť vždy rozhodujúcim faktorom. Dôležitejším je akceptácia okolia, schopnosť bezproblémovej práce v tíme, potreba využívať niektoré špecializované aplikácie existujúce výhradne pre jeden či druhý systém. Konečné rozhodnutie však musí urobiť používateľ, ktorý až po určitom čase strávenom používaním konkrétneho OS dokáže zhodnotiť jeho výhody či nevýhody

- b) **Pomocou nainštalovaného antivírusového programu urobte kontrolu C disku počítača na prítomnosť vírusov (program ESET ANTIVIRUS)**

## 22. Úloha č.1: Aritmetika - náhodné čísla

Vysvetlite pojmy: pole, prvok poľa, náhodné číslo, generátor náhodných čísel.

*Príklad*

V kasíne si chceli overiť „správnosť“ softvérového generátora náhodných čísel nastaveného na hracom automate simulujúcom hru v kocky. Zaujímalo ich či výherné číslo 6 nenabieha príliš často. Napíšte program, ktorý im v tom pomôže. Program na začiatku vygeneruje a vypíše na obrazovke monitora vedľa seba 40 celých čísiel z intervalu <1,6>.

Vypočítajte a vypíšte koľkokrát padlo číslo 6. Vypíšte na jedno desatinné miesto aj percentuálne zastúpenie čísla 6 vzhľadom k celkovému počtu čísel.

### Úloha č.2:

- a) Popíšte ako sa kódujú rastrové obrázky. Koľko farieb možno zakódovať 4, 8, 16, 32 bitmi ? Zakódujte obrázky z prílohy pomocou postupnosti 0 a 1.

Rastrové obrázky sú rozdelené na sieť myslených štvorcíkov - **pixelov**. Rozmer každého obrázka je pre počítač počet pixelov na šírku x počet pixelov na výšku. Pre každý pixel (štvorček) je nutné okrem polohy (riadok a stĺpec) zakódovať aj farbu, resp. ďalšie parametre.

Ak je obrázok monochromatický (čierna a biela farba), kódovanie je jednoduché (1 - rozsvietený (biely) bod, 0 - nerozsvietený (čierny) bod), napr.

```
100111001
011010110
101101101
110111011
111010111
111101111
```

A teda zápis v riadku vyzerá : 011000110 100101001 010010010 001000100  
000101000 000010000, čo je 54 bitov.

Pri niektorých obrázkoch je však potrebné zaznamenať i odtiene, preto dve farby - čierna a biela - nestačia. Výhodné je zakódovať odtiene sivej farby pomocou ôsmich

bitov tak, aby informácia o každom bode zaberala 1 B, t.j. jedno pamäťové miesto počítača. Čím bude bod svetlejší, tým väčšia hodnota sa do pamäte uloží. Preto bod čiernej farby bude uložený ako 0 a bod bielej farby ako maximálna možná hodnota - 255. Obrázok teda bude kódovaný nasledovne:

```
192 0 0 255 255 192 0 0 255
0 255 192 0 192 0 255 192 0
192 0 255 192 0 255 192 0 255
255 192 0 255 255 192 0 255 255
    255 5 192 0 192 0 255 255 255
255 255 255 192 0 255 255 255 255
```

Obrázok bude v pamäti počítača zaberat' 54 Bajtov (pamäťových miest).

**b. Definujte kroky pre bezpečné používanie operačného systému, čo je potrebné dodržiavať, nainštalovať..., aby práca s operačným systémom bola bezpečná.**

- Update operačného systému a aplikácií - pravidelným update sa dá predísť poškodeniu operačného systému, aplikácií alebo údajov. Prostredníctvom update získa užívateľ najnovšie opravné balíčky, ktoré riešia nedostatky existujúcich verzií. Update sa najčastejšie vzťahuje na operačný systém, internetové prehliadače, aplikácie elektronickej pošty, antivírusové a antispamové aplikácie.

Predchádzanie a prevencia je najúčinnější zbraň v súčasnom svete internetu a počítačových sietí.

### **Pasívna ochrana údajov**

Archivácia operačného systému a údajov:

- uloženie údajov a ich opätovné použitie
  - ukladáme na médium, ktoré nie je súčasťou počítača - napr. disk CD alebo DVD, na kartu USB flash alebo do iného počítača, ak je náš počítač pripojený k sieti,
  - pre získanie ešte väčšieho priestoru, je možné údaje pred uložením skomprimovať.

očítačová ochrana je často více zaměřena na techniku a matematiku, než některé jiné oblasti kybernetiky a informatiky. Spočívá ve třech krocích:

1. *prevence* – ochrana před hrozbami
2. *detekce* – odhalení neoprávněných (skrytých, nezamýšlených) činností a slabých míst v systému
3. *náprava* – odstranění slabého místa v systému

Zlepšení počítačové bezpečnosti může zahrnovat následující kroky:

- omezení fyzického přístupu k počítači pouze pro ty, kteří budou dodržovat bezpečnost při práci s počítačem a daty

- použití [hardwarových](#) zařízení, která vynucují bezpečnostní opatření, což snižuje závislost počítačové bezpečnosti na [software](#) (počítačových programech)
- využití mechanismů [operačního systému](#), která vynucují pravidla chování programů, aby byl omezen rozsah programů, kterým je nutné důvěřovat
- využití záznamů o změnách v programu ([verzování](#)), které je možné využít pro sledování jejich vývoje

Pokud budeme mluvit o počítačové bezpečnosti, nesmíme zapomenout na bezpečnost [operačního systému](#), bezpečnostní projekt a bezpečné [šifrování](#) ([kryptografie](#)).

## Bezpečnostní projekt

Aby byla ochrana počítačového systému efektivní je potřebné vypracovat bezpečnostní projekt. Cílem tohoto projektu je docílení takového stavu, aby úsilí, riziko odhalení a finanční prostředky potřebné na narušení bezpečnostního systému byly adekvátní v porovnání s hodnotou, která je bezpečnostním systémem chráněna.

### Části bezpečnostního projektu

#### Zabezpečení fyzického přístupu

Zabezpečení fyzického přístupu spočívá v zabránění přístupu nepovolaných osob k částem počítačového systému. Na toto zabezpečení se používají bezpečnostní prvky jako přidělení rozdílných práv zaměstnancům, elektronické zámky, poplašné zařazení, kamerové systémy, autorizační systémy chráněné hesly, čipovými kartami, autentizační systémy na snímání otisků prstů, dlaně, oční duhovky, rozpoznání hlasu, auditovací systémy na sledování a zaznamenávání určitých akcí zaměstnanců (vstup zaměstnanců do místnosti, přihlášení se do systému, kopírování údajů atd.).

## Zabezpečení počítačového systému

Zabezpečení počítačového systému spočívá v zabezpečení systému před útokem [crackerů](#), škodlivých programů ([viry](#), [červy](#), [trojské koně](#), [spyware](#), [adware](#), ...). do této části patří i zaškolení zaměstnanců, aby se chovali v souladu s počítačovou bezpečností a dodržovali zásady bezpečného chování na síti.

## Zabezpečení informací

Zabezpečení informací spočívá v bezpečném [zálohování dat](#). Záloha dat by měla být vytvořena tak, aby ji neohrozil útočník ani přírodní živelní pohroma (požár, záplavy, pád letadla, ...). Zálohovaná data je také potřeba chránit proti neoprávněné manipulaci použitím vhodného [šifrovacího](#) systému.

## Ekonomické a právní zabezpečení

Ekonomické a právní zabezpečení spočívá ve správné motivaci a postihu zaměstnanců.



•

**c. Definujte pojem otvorený softvér. Uved'te výhody a nevýhody**

Definícia otvoreného softvéru (Open Source Definition)

Definícia otvoreného softvéru[13] je rozsiahlejšia ako definícia slobodného softvéru, skladá sa z 10 bodov:

1. Licencia musí dovoliť redistribúciu za nulový poplatok alebo za finančnú čiastku.
2. Program musí obsahovať zdrojový kód.
3. Odvodené dielo musí byť redistribuované pod rovnakými podmienkami ako pôvodné dielo.
4. Licencia môže obmedziť distribúciu zdrojového kódu v zmenenej forme iba vtedy, ak licencia dovoľuje distribúciu opravných súborov.
5. Licencia nesmie diskriminovať jednotlivcov alebo skupiny.
6. Licencia nesmie obmedzovať spôsob použitia.
7. Licencia musí zaručiť také isté práva pre nového vlastníka kópie zdrojového kódu, akými disponoval pôvodný vlastník.
8. Licencia musí byť aplikovateľná na daný program a nemôže byť závislá od iných programov.
9. Licencia nesmie obmedzovať iný softvér, ktorý je distribuovaný s licencovaným soft-verom.
10. Licencia musí byť technologicky neutrálna

**výhody**

1. väčšia možnosť pri výbere dodávateľov programu
2. nižšie náklady
3. jednoduchšie prispôbenie programu pre vlastné potreby
4. vyššia bezpečnosť

5. Za najväčšiu výhodu bola vyhodnotená voľba c.1. Ako príklad poslúži nasledovná situácia. Ak spoločnosť Microsoft vydá novú verziu operačného systému Windows, po určitom čase zanikne podpora starších typov operačných systémov Windows. Keďže neexistuje iná podpora ako podpora od spoločnosti Microsoft, používateľ je donútený upgradovať svoj systém na novší - resp. ešte na momentálne podporovaný. Ak používateľ tento krok neuskutocní, stratí nárok na akékoľvek aktualizácie a záplaty a to aj vysoko kritického charakteru. V Open Source je to diametrálne odlišné. Staršie verzie produktov (už nepodporovaných od pôvodného výrobcu) udržiava komunita alebo rôzne tretie strany, preto si používateľ môže vybrať, či produkt zaktualizuje alebo nie.

6. Existuje obrovské množstvo Open Source aplikácií, ktoré sú voľne stiahnuteľné z Internetu. Monitoruje ich Open Source portál freshmeat.net. Ak potrebujeme nejakú aplikáciu na riešenie nášho problému, môžeme si vybrať z

viacerých variánt. Rozdiely sú v kvalite a úrovni spracovania problému. Inštalácia závisí od zručnosti používateľa, nie vždy je to jednoduché. Na Internete sa vyskytuje neuveriteľné množstvo návodov (HOWTO), postupných krokov ako program nakonfigurovať podľa želania. Napríklad, stačí použiť v prehliadači frázu "howto + jabber", kliknúť na linku a máme pravdepodobne popis inštalácie. Ak sme našli chybu alebo prišli na zlepšenie v programe (vo všeobecnosti bug), môžeme to zaznamenať v bugtrackingovom systéme<sup>2</sup> programu. Zaznamenanie a vyriešenie vedie k tomu, že bug bude odstránený a už sa nebude vyskytovať v najbližšej vydanej verzii.

7. Kvalitné produkty majú kvalitne spracovanú dokumentáciu, ktorá je udržiavaná v elektronickej podobe. Sú to súbory vo formáte pdf alebo Wiki stránky, existujú aj videá s nahovorenými krokmi inštalácie. Dokumentácie sú voľne stiahnuteľné z Internetu.

8. Poskytovanie profesionálnych služieb je ohodnotené finančnými prostriedkami. Každý poskytovateľ môže ohodnotiť svoju službu na ľubovoľnú finančnú čiastku. Ako som vyššie uviedol, na Internete sú voľne dostupné materiály, ako službu sprístupniť. Produkt nemusí poskytovať len vlastník produktu, môže to byť dokonca aj koncový používateľ. Rozdiel je ale v kvalite služby. Preto si spotrebiteľ môže vybrať medzi dodávateľmi podľa vlastných kritérií.

## Nevýhody Open Source

9. Proprietárny program je dodávaný s licenciou, ktorá často stojí oveľa viac finančných prostriedkov ako program. Licencia slobodného programu je zadarmo. To, čo môže byť finančne ohodnotené, sú poskytované služby k programu, ktoré sa môžu pohybovať v inej (oveľa vyššej) platobnej sfére ako proprietárna licencia.

10. Proprietárny produkt je navrhnutý profesionálnymi vývojármi, programovanie je riadené a je pod dohľadom spoločnosti, na trh sa dostávajú celistvé verzie programu v určitých časových intervaloch. Je to tzv. katedrálový vývoj. Vývoj Open Source programu poznačuje cieľ tzv. bazárový štýl vývoja[8]. Nové verzie vychádzajú s pravidelnou nepravidelnosťou, najčastejšie tak skoro, ako je to možné. Každá verzia prebieha cyklom neustálych modifikácií, pokiaľ sa neuzatvorí vývojová časť. Potom nasleduje obdobie testovania a konečné vydanie programu. Verzie počas záverečného testovania sú pomenované ako RC (release candidate), finálna verzia má už nemenný názov. Avšak vývojový cyklus neskončí, pokračuje stále ďalej. Nasadenie nestabilnej verzie do produkčného prostredia si vyžaduje každodenné sledovanie nových úprav a záplat, údržba je náročná na čas. Stabilná verzia už nevyžaduje toľkú pozornosť, spôsob údržby je porovnateľný s proprietárnym programom. Problém prichádza pri jeho ďalšej aktualizácii. Môže nastať situácia, že sa produkt úplne prestane vyvíjať. Stratí sa podpora a budúcnosť nasadeného riešenia založeného na Open Source produkte je veľmi malá. Pokracovanie vo vývoji programu závisí od schopnosti adaptovať nový zdrojový kód. Lenže to vôbec nie je jednoduché. Dokumentácia nie je tak dobre spravovaná ako pri proprietárnom programe, programátori nie sú nútení dokumentovať svoju prácu. A nadviazať na neznámy kód

---

<sup>2</sup> systém, ktorý monitoruje kolobeh bugov v programe. Správca kategórie bugu (napr. Lokalizácia) priradí bug programátorovi, ktorý po úprave v programe informuje správcu. Ak je bug vyriešený, bug sa uzavrie a skončí sa jeho životný cyklus.

pre priemerného programátora je ako hľadať ihlu v kope sena. Je veľa nedokončených projektov, ktoré majú dobrú myšlienku, ale zlú implementáciu. Sú vytvárané neskúsenými programátormi, ktorí nenesú zodpovednosť za finálny projekt.

11. Preniknutie do Open Source prostredia<sup>3</sup> je umožnené prostredníctvom rôznych mailing listov, diskusií a aktívnych komunikácií (napr. IRC), ale existuje veľmi málo takých, ktoré sú venované úplným zaciatočníkom. Typický znak fóra (ale aj ďalších prostriedkov) pre zaciatočníkov je, že ak sa používateľ dožaduje nejakej informácie, ktorá už bola raz zodpovedaná, dostane odpoveď bez vedľajších poznámok. Pri spýtaní sa rovnakej otázky v inom fóre už nie pre zaciatočníkov, používateľovi nie je zodpovedané veľmi ústretovo alebo vôbec. Odpovede sú často typu RTFM a UTFG<sup>4</sup> a odrádza ho to od ďalšieho používania.

12. Otvorený operacný systém je veľmi interaktívny a dáva to najavo prostredníctvom rôznych výpisov do súborov, na konzolu alebo priamo do aktívneho okna na obrazovke. Pre zaciatočníka je to zbytočne mäťúci fakt. Nepotrebuje vedieť, či program vykonal nepovolenú operáciu alebo bola chybná segmentácia pamäte. Je jednoduché povedať používateľovi „cítaj, čo poť cítať c píše“. Nie je na to zvyknutý. Sú situácie, keď zlyhá grafické prostredie a používateľ sa ocitne v textovom konzolovom prostredí. Na obrazovke chýba jasný jednoduchý popis, čo treba prekonfigurovať. Je to pre neho slepá ulička a signál, ktorý informuje o tom, že takto moderný systém nemôže vyzerat

## 23. Úloha č.1: Reťazce – kódovanie znakov

Vysvetlite pojmy: znak, znaková premenná, ASCII tabuľka

*Príklad*

Vytvorte jednoduchý šifrovací program. Program načítava nezašifrovaný text zložený z veľkých písmen po znakoch zo súboru *vstup.txt*, posunie každé písmeno na jeho nasledujúce v abecede (A->B,... Z->A) a výsledný zašifrovaný text zapíše do nového textového súboru *vystup.txt*. Všetky ostatné nepísmenové znaky ponechá presne tak ako sú. Načítavanie údajov z textového súboru a zápis do nového súboru môže prebiehať naraz v tom istom cykle.

### Úloha č.2:

a) Demonštrujte na príkladoch aritmetické operácie v binárnej sústave.

Príklady:

$$1100 + 1010 = ?$$

$$11001 - 101 = ?$$

$$101 \cdot 101 = ?$$

$$10101 : 011 = ?$$

Binárna aritmetika

<sup>3</sup> prostredie, kde sú základom Open Source programy

<sup>4</sup> read the f\* manual; use the f\* google



číslicou. Touto číslicou vynásobíme súčiniteľ a výsledok zapíšeme tak, aby jeho posledná číslica bola zapísaná pod číslicou, ktorou sme násobili. Ďalej pokračujeme v násobení sprava doľava. Pre zjednodušenie, ak je číslica, ktorou násobíme súčiniteľ jednotky, odpíšeme súčiniteľ. Ak je to nula, zapíšeme nulu. Jednotlivé čiastkové výsledky zapisujeme pod seba (sprava nám od druhého riadku v každom ďalšom pribúda jedna medzera). Tieto čiastkové výsledky nakoniec binárne sčítame. Odporúča sa sčítavať tieto čísla po dvojiciach, aby sme tak predišli možným chybám.

**Úloha:** Binárne násobte dané čísla.

$$5 \cdot 5 = 25$$

$$\begin{array}{r} 101 \\ \cdot 101 \\ \hline 101 \\ 000 \\ \hline 101 \\ \hline 11001 = 25D \end{array}$$

$$26 \cdot 7 = 182$$

$$\begin{array}{r} 11010 \\ \cdot 111 \\ \hline 11010 \\ 11010 \\ 11010 \\ \hline 10110110 = 182D \end{array}$$

Binárne celočíselné delenie Binárne delenie nahradíme posunom a binárnym odčítaním. Postup pri delení je podobný ako pri delení v dekadickej sústave. Pri delení si najskôr všimame prvé čísla delenca a celého deliteľa. Hľadáme totiž, v akom čísle sa deliteľ nachádza raz. Dobrým zvykom je doplniť delenca zľava nulami tak, aby sa počet jeho bitov rovnal s počtom bitov s ktorými sme museli rátať v prvom kroku. Je to dobré hlavne z hľadiska dvojkového doplnku, ktorý z daného deliteľa (doplneného nulami) bude potrebné vytvoriť. Ako vidíme v príklade, číslo 11B sa raz nachádza až v čísle 101B, preto sme ho doplnili na 011B. Keď sme našli číslo v ktorom sa raz nachádza deliteľ, od tohto čísla binárne deliteľa odčítame (pripočítame k nemu dvojkový doplnok = 101B). Nezabudneme zapísať prvú číslicu výsledku. Po binárnom odčítaní pokračujeme v počítaní. Posúvame sa o jeden bit delenca vpravo. Tento bit sprava pripíšeme k výsledku odčítania. Zistíme, či sa v tomto čísle nachádza raz deliteľ. Ak áno, pokračujeme ako v predchádzajúcom. Ak nie, do výsledku zapíšeme 0 ako ďalšiu číslicu a k výsledku odčítania pripíšeme ďalší bit sprava a pokračujeme v počítaní. Takýmto spôsobom počítame, kým sme neprešli všetkými bitmi delenca a výsledok binárneho odčítania nie je menší ako deliteľ. Ak výsledok odčítania je väčší ako 0, dostali sme zvyšok po celočíselnom delení.

**Úloha:** Binárne delte dané čísla.

$$21 : 3 = 7$$

$$\begin{array}{r} 10101 : 011 = 111 = 7 \qquad 011 \rightarrow \text{dvojkový doplnok} = 101 \\ \hline +101 \\ \hline 1010 \\ \hline 100 \\ \hline +101 \\ \hline 1001 \end{array}$$

$$\begin{array}{r} 11 \\ +101 \\ \hline 1000 \end{array}$$

Prezentacia

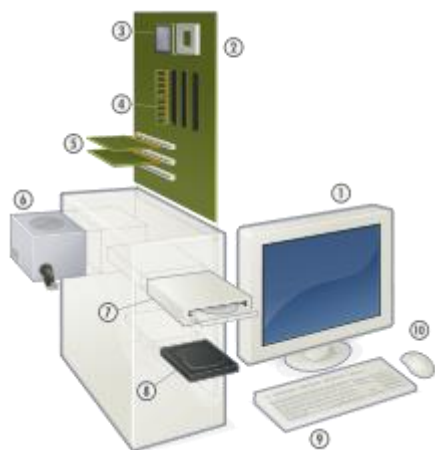
[spseke.sk/web/copko/download/4\\_rocnik/pos.../binarna\\_aritmetika.pps](http://spseke.sk/web/copko/download/4_rocnik/pos.../binarna_aritmetika.pps)

b. Vysvetlite pojmy: počítač, hardvér, softvér. Popíšte rozdiely medzi systémovým a aplikačným softvérom a ďalej ich charakterizujte. Nájdite na počítači príp. cez internet aspoň dvoch predstaviteľov z každého druhu softvéru.

Osobný počítač dneška (2013) je postavený na báze procesorov architektúry x86, s výkonnou grafickou kartou umožňujúcou prehrávať multimediálny obsah, s optickou mechanikou umožňujúcou zápis na veľkokapacitné médiá (DVD, Blu-Ray), s operačnou pamäťou rádovo ~ jednotkách GB, s vysokokapacitným pevným diskom ~ stovky GB, s možnosťou pripojenia do počítačovej siete a k internetu. PC je tiež vybavené portami na pripojenie periférnych zariadení (vstupné a výstupné periférie, ako aj zariadenia schopné komunikovať s počítačom – MP3 prehrávač, mobilný telefón, a pod.). Ako výstupné zariadenie sa používa LCD monitor s uhlopriečkou 17"-30", tlačiareň (monochromatická obvykle laserová alebo farebná obvykle atramentová prípadne vo forme multifunkčného zariadenia) ako vstupné zariadenie sa používa klávesnica a optická prípadne laserová myš.

V oblasti programového vybavenia je najrozšírenejším operačným systémom Windows od firmy Microsoft (verzia XP, Vista, 7 alebo 8), s rozšírením až po úroveň monopolu. Druhou najväčšou skupinou je OS X firmy Apple. Alternatívou - aj keď relatívne málo rozšírenou sú Open Source operačné systémy s jadrom Linux.

### Zloženie osobného počítača



- (1) monitor
- (2) matičná doska
- (3) Procesor
- (4) RAM
- (5) Rozširujúca karta
- (6) Zdroj

- (7) Optická mechanika
- (8) Pevný disk
- (9) Klávesnica
- (10) Myš

Ak hovoríme o dieloch osobného počítača, musíme oddeliť dve základné roviny – softvér a hardvér. Hardvér zjednodušene je „železo“, to čo sa dá chytiť do ruky, to čo „vdýchne ducha“, čo mu umožní pracovať je jeho softvérové vybavenie.

## Hardvér

Osobný počítač sa skladá zo základnej jednotky (systemová jednotka), vstupných a výstupných zariadení.

ení.

**Systemová jednotka** Typický osobný počítač (jeho systemová jednotka) je uložený v plechovej (resp. plastovej) skrinke v ktorej sú uložené všetky podstatné časti počítača (prípadne spolu s monitorom a V/V jednotkami integrovaný v jednom tele – notebook).

- **Počítačová skriňa** (case) je dnes obvykle vo forme tower (anglicky *veža*), čiže nastojato orientovaná skrinka. Tento formát nahradil predtým najrozšírenejšie usporiadanie desktop (skrinka naležato). Usporiadanie *tower* zaberá menej miesta, a je aj vhodnejšie z hľadiska chladenia komponentov. Najbežnejšie používanou formou je *minitower*, čo je skrinka o rozmeroch cca 520 x 520 x 210 mm. Obvykle je vyrobená z plechu. Má odnímateľné veko (alebo bočné steny), ktoré po odstránení odhalia samotné šasi. Šasi je kostra skrine s vytvorenými upevňovacími plochami a otvormi na a do ktorých sa pripievňujú všetky interné mechanické a elektrické diely počítača.
- **Počítačový zdroj** (power supply) je zariadenie modifikujúce napájacie napätie elektrickej rozvodnej siete, na napätia potrebné pre jednotlivé komponenty počítača (matičná doska, mechaniky a pod.). Dnešné zdroje majú príkon (výkon) 300-500VA.
- **Matičná doska** (základná doska, motherboard) je v doska plošného spoja obsahujúca konektory a prípojné miesta všetkých vnútorných periférií. Obsahuje soket na upevnenie procesora, pamäťového modulu, zbernice na pripojenie rozširujúcich kariet, hodiny reálneho času, čipset (obvody riadiace komunikáciu medzi komponentmi a vonkajším okolím), porty (konektory pre pripojenie vstupných a výstupných zariadení – klávesnice, myši a pod. Obvyklé je aj integrovanie rozširujúcich kariet priamo na základnú dosku (zvuková karta, sieťová karta, modem, grafická karta a pod.).
- **Rozširujúce karty**. Sú to externé moduly počítača vo forme karty (doska plošného spoja veľkosti pohľadnice), ktorá má na spodnej časti zásuvné kontakty. Kontakty sa zasúvajú do zbernice matičnej dosky. Použitá karta musí zodpovedať typu zbernice na matičnej doske. Najčastejšie používanou rozširujúcou kartou je grafická karta, pomocou ktorej počítač zobrazuje na monitore. Ďalšími príkladmi sú sieťová karta (pripojenie do počítačovej siete), zvuková karta (karta vytvárajúca zvuky), TV karta (karta pre príjem TV signálu) a pod. Viacero rozširujúcich „kariet“ – modulov sa pripája externe pomocou zbernice USB.
- **Procesor** (CPU) alebo tiež centrálny procesor je „mozgom“ počítača. Pre procesor je na matičnej doske pripravená päťica – konektor do ktorého sa procesor zasunie.

Processor vyvíja veľké množstvo tepla a preto musí byť pri prevádzke neustále chladený. Procesor vykonáva inštrukcie programového vybavenia a umožňuje chod celého počítača.

- **Pamäť** (RAM) je modul slúžiaci na uloženie dát počas chodu počítača. Je to malá doska plošného spoja s pamäťovými čipmi. Umiestňuje sa do soketu (konektora) na matičnej doske.
- **Mechanika**. Mechanik je viacero typov. Najčastejšie sa používa **optická mechanika** (číta a zapisuje na optické disky (CD a DVD) a **disketová mechanika** (číta a zapisuje na magnetické diskety). Mechanika je plechová skrinka s plastovým čelným panelom. Mechanicky sa upevní do skrine a elektricky káblami prepojí s matičnou doskou. Optická mechanika môže byť štandardu **IDE** (plochý kábel), alebo **SATA** (úzky kábel). Napája sa zo **zdroja**. Zariadenie slúži na čítanie a zápis optických médií.
- **Pevný disk**. Je zariadenie na ukladanie dát. Je to kovová skrinka veľkosti 3.5“. Disk sa upevňuje do skrine a elektricky (plochým IDE káblom, alebo SATA káblom) prepojí s matičnou doskou a napájacím zdrojom. Je na ňom uložené celé **softvérové** vybavenie počítača.

## Vstupné zariadenie

Vstupné zariadenie (vstupná jednotka) je periférne zariadenie (**hardvér**) osobného počítača, ktorá umožňuje vstup **údajov** alebo **signálov** do počítača za účelom ich ďalšieho spracovania alebo ich využitia pre riadenie počítača a k nemu pripojených zariadení. Vstupné zariadenie transformuje informácie z “vonkajšieho” sveta do formy použiteľnej pre počítač. Vstupné zariadenia často slúžia užívateľovi – **človeku** ktorý ich používa pre zadanie **príkazov**, alebo ďalších **informácií** spracovávaných počítačom. Počítač odpovedá užívateľovi pomocou výstupných zariadení počítača. Niektoré zariadenia sú kombinované vstupno-výstupné zariadenia. Typickým príkladom vstupného zariadenia je klávesnica, myš, mikrofón, skener, webová kamera.

**Výstupné zariadenie** Výstupné zariadenie (výstupná jednotka) je periférne zariadenie (**hardvér**) počítača, ktoré umožňuje výstup údajov alebo signálov z počítača. Výstupné zariadenie transformuje informácie z počítača do formy použiteľnej pre človeka (alebo pre iné zariadenie resp. ďalší počítač). najpoužívanejším výstupným zariadením je monitor (displej), a počítačová tlačiareň.

## Iné zariadenia

Osobný počítač sa využíva aj ako zariadenie spracúvajúce, alebo ukladajúce údaje pre iné zariadenia, ktoré nie sú typickými perifériami počítača. Príkladom je pripojenie **mobilného telefónu** (ukladanie kontaktov, ich triedenie a odoslanie späť do telefónu), spracovanie obrazu a zvuku pre multimedialne prehrávače (stiahnutie filmov, prevody multimedialnych formátov, úpravy fotografií a ich tlač a pod.). Z tohto hľadiska je počítač jedným z najvšestrannejších zariadení na komunikáciu vôbec.

## Softvér

**Hardvér** (hardware alebo HW) je súhrnný názov pre technické vybavenie počítača a počítačových komponentov. Medzi hardvér patria všetky počítače a ich súčasti, periférie



(zobrazovacie jednotky, zariadenia na vstup a výstup údajov) a tiež aj zariadenia, ktoré je možné pripojiť k počítaču. V úvodzovkách „všetko, čo sa dá chytiť“, čiže všetko hmotné čo patrí ku počítaču.

**Softvér** (angl. software alebo SW) je označenie pre programové vybavenie počítača alebo súhrn všetkých programov, ktoré sa dajú použiť na výpočtovom zariadení. Má nemateriálnu povahu, hovoríme o vývoji softvéru. Rozlišujeme systémový softvér a aplikačný softvér. Medzi softvér zaradíme tak operačné systémy a ovládače zariadení, ako aj všetky druhy aplikačných programov, napríklad textové editory a iné kancelárske aplikácie, grafické aplikácie, aplikácie na prehrávanie multimédií, hry a všetky ostatné programy. Počítačový softvér vyrábajú programátori, grafici atď v rôznych spoločnostiach, alebo tiež v komunite používateľov. Tu pramení aj hlavné rozdelenie softvéru na softvér bezplatný a platený. Bezplatný softvér je voľne šírený a môže byť jednoducho upraviteľný. Platený softvér môžete používať iba vtedy, ak zaplatíte za licenciu softvéru, a nebudete túto licenciu porušovať. Platený softvér teda nemožno vlastniť, je len možné používať ho po zaplatení a splnení podmienok. Softvér sa dá zohnať na Internete, lokálnej sieti, na pamäťových médiách (CD, DVD...), alebo už predinštalovaný na novom počítači.

## Softvér

Fungovanie počítača nie je možné bez programového vybavenia. Samotný počítač je kompilácia programovateľných obvodov, pamäti a úložisk, ktorá však nie je schopná samostatnej práce. Pre svoju činnosť potrebuje základný tzv. operačný systém, na ktorom bežia aplikácie – aplikačný softvér.

## Operačný systém

Operačný systém (skratka OS) je základné programové vybavenie počítača. Je to „rozhranie“, ktorým používateľ komunikuje s hardvérom. Operačný systém zabezpečuje čiastočne komunikáciu s užívateľom prostredníctvom periférnych zariadení, pridelovanie prostriedkov systému užívateľom alebo aplikáciám, pridelovanie času užívateľom (tzv. timesharing) alebo aplikáciám (multitasking), organizácia programov a súborov údajov na vonkajších pamäťových médiách, vytváranie a spúšťanie užívateľských programov, diagnostické funkcie systému, ochraňuje systém proti strate údajov pri výpadku napätia, proti neoprávnenej manipulácii, komunikáciu s inými systémami (počítačmi) v sieti - pri použití sieťových operačných systémov. Operačný systém sa skladá z jadra, používateľského rozhrania napr. GUI a ovládačov (softvér prepájajúci špecifický hardvér s jadrom operačného systému).

V súčasnosti je najpoužívanejší operačný systém Microsoft Windows. Je to viacgeneračný systém. Jeho hlavnou výhodou je rozšírenie, takže pre neho existuje obrovské množstvo komerčného i voľne šíriteľného softvéru a aplikácií.

Najrozšírenejšou alternatívou MS Windows je operačný systém OS X firmy Apple a operačný systém Linux. Hlavnou výhodou je, že je zadarmo, ale i to, že preň už existuje veľké množstvo aplikácií z oblasti voľne šíriteľného softvéru.

## Aplikačný softvér

V súčasnosti sú možnosti osobných počítačov také, že zahŕňajú takmer všetko, čo sme schopní zadefinovať. Preto existuje i obrovské množstvo [softvéru](#), ktoré tieto predstavy realizuje.

- **Textový editor** (MS Word, Writer a pod.) Softvér pre písanie textov a vytváranie stránok kombinujúcich text a grafiku.
- **Tabuľkový kalkulátor** (MS Excel, Quattro, Calc atď.) Je program na vytváranie tabuliek. Je to matica čísel (textu, vzorcov) medzi ktorými je možné vykonávať matematické a textové operácie.
- **Databázové systémy** (Microsoft Access, [MySQL](#)) Programy na vytváranie a správu databáz.
- **Prezentačný softvér** (Microsoft PowerPoint, Impress) Programy na vytváranie prezentácií, multimediálnych súborov určených pre prezentovanie činnosti, alebo vecí. Je možné vkladať text, obraz, video, zvuk ...
- **Antivírusové programy a antispýwarové** (NOD, AVG, Avast, Norton Antivirus, [ClamAV](#)) slúžia na ochranu počítača pred tzv. škodlivými programami - [vírusmi](#), [trójskymi koňmi](#), [spyware](#) a pod.
- **Správa súborov** (Windows Commander, Servant Salamander, Turbo Navigator, [Midnight Commander](#), [Free Commander](#)) Programy na správu súborov. Umožňujú kopírovanie, prenášanie, náhľady, sledovanie zmien, vytváranie a správu adresárovej štruktúry.
- **Zálohovacie, archivačné a kompresné nástroje** (Nero Burning ROM, WinZip, WinRAR, Power Archiver, UltimateZip, 7-Zip ...) Programy pre vytváranie záloh, kompresiu dát a pod.
- **Multimediálne utility** (CD a DVD prehrávač, MP3player, [VLC](#), prezerač grafických súborov) Programy pre prehrávanie a vytváranie multimediálneho obsahu.
- **Grafický editor** ([Adobe Photoshop](#), [Paint Shop Pro](#), [Corel Draw!](#), Corel Photo-Paint, [GIMP](#), [Inkscape](#))
- **Vývojové prostredia programovacích jazykov** (Turbo Pascal, Borland Delphi, MS Visual Basic, [FreePascal](#)...) Sú programy na vytváranie softvéru. Vyššie programovacie jazyky umožňujú naprogramovať aplikačný softvér. Súčasťou balíka sú ladiace programy a simulátory.
- **Internetové nástroje** ([Opera](#), [Firefox](#), [Thunderbird](#), [Internet Explorer](#), [Netscape](#), off-line sťahovače www stránok, [FTP](#), downloadery, HTML editory, ... ) Nástroje na prezeranie obsahu internetu, sťahovanie, vyhľadávanie stránok, sťahovanie a odosielanie pošty.
- **Systémové nástroje** (ovládače, defragmentácia) Systémové nástroje sú programy na správu operačného systému, správu a údržbu softvéru.
- **Vzdelávacie programy** (encyklopédie, slovníky, prekladače a iné)
- **Geografické informačné systémy a GPS** Programy pre prácu s GPS, mapové softvéry, navigačné softvéry.
- **Ekonomické aplikácie** (účtovnícke, skladové, výpožičné) Ekonomické balíky, účtovníctvo, mzdy, fakturácia, správa skladu, spolupráca s registračnými pokladňami.
- **DTP** (DeskTop publishing) Programy pre prípravu tlače - kníh a časopisov.
- **CAD a 3D modelovanie** Programy pre kreslenie stavebným a strojárskych výkresov, modelovanie, 3D výstupy, renderovanie a pod.
- **Programy na oddych a relax (hry)**

### c. Uved'te príklady nevyhnutnej potreby informačných technológií v rozvoji vedy, ekonomiky, vzdelávania.

informatizácia spoločnosti je postupný prechod k maximálnemu využívaniu informačných a komunikačných technológií vo všetkých oblastiach spoločenského, politického a hospodárskeho života. V oblasti školstva zavádzanie najmodernejších informačno-komunikačných technológií do vyučovacieho procesu, tvorba e-learningového obsahu vyučovania a školenia pedagógov pre aktívne používanie informačno-komunikačných technológií vo vyučovacom procese je kľúčovým faktorom ďalšieho rozvoja vzdelanosti v Slovenskej republike. Úlohy ministerstva v oblasti informatizácie plní Sekcia informatiky. Medzi jej úlohy patrí vypracovávanie koncepcií a stratégií rozvoja informatiky a informatizácie rezortu školstva. V spolupráci s Ústavom informácií a prognóz školstva a v súčinnosti s inými orgánmi štátnej správy a samosprávy plánuje a riadi informačné systémy v školstve.

Informačná spoločnosť: Zmeny vyvolané narastajúcim významom informácií a IKT (informačné a komunikačné technológie) nemajú len technologický charakter a neovplyvňujú iba ekonomiku. Nemenej významným faktorom je aj ich difúzia do života spoločnosti a občanov. Dôsledkom akcelerujúceho vplyvu IKT je kvalitatívna zmena charakteru spoločnosti z industriálnej na informačnú (hovorí sa aj o informačnej revolúcii).

Možnosti využitia IKT v živote spoločnosti sú obrovské a ďalekosiahle. Popri uplatnení v ekonomických odvetviach, ktoré majú priamy vplyv na život občanov (napr. doprava, energetika), sa uplatňujú vo vzdelávaní, zdravotníctve, ochrane životného prostredia atď. Nezanedbateľný je aj význam IKT pri sprístupňovaní informácií občanom, pri zjednodušení ich komunikácie s verejnou správou a pri zvyšovaní podielu občana na riadení štátu. Rozvoj informačnej spoločnosti sa stal politickým programom všetkých ekonomicky vyspelých krajín a zoskupení (USA, EÚ, OECD). Politické dokumenty sa stali východiskom na stanovenie dlhodobých strategických programov stimulujúcich proces rozvoja informačnej spoločnosti (ďalej informatizácia spoločnosti). Spoločným cieľom strategických programov je podporiť ekonomický rozvoj, uľahčiť prácu a spríjemniť život, sprístupniť a skvalitniť vzdelávanie a efektívne využitie voľného času každému človeku.

Programy informatizácie spoločnosti vyspelých krajín, okrem vytýčenia strategických zámerov a cieľov štátu, stanovujú konkrétne kroky pre rozvoj informačnej spoločnosti. Tieto programy sa orientujú najmä na podporu budovania informačnokomunikačnej infraštruktúry, na vytvorenie potrebnej legislatívy, na pilotné projekty informatizácie verejnej správy, na rozvoj digitálnej gramotnosti a na podporu informatického výskumu. Popritom vytvárajú priestor pre rozvoj elektronického podnikania, informatizáciu zdravotníctva, dopravy, kultúry atď. V tomto kontexte je nevyhnutné riešiť celý rad sociálnych otázok, ochranu ľudskej dôstojnosti, detí, mládeže, spotrebiteľov, ochranu informácií, právo na informačné sebaurčenie atď. Význam štátnej strategickej vízie spočíva aj v tom, že vytvára priestor pre investície podnikateľského sektoru.

Informatizácia spoločnosti nie je len technologickým problémom - svojím záberom a dôsledkami sa stáva najmä ekonomickým, legislatívnym, regulačným, sociálnym, politickým a etickým problémom. Ak sa proces informatizácie spoločnosti realizuje nedôsledne, povrchno alebo oneskorene, môže mať aj politické, ekonomické, sociálne a etické dôsledky v polarizácii

spoločnosti (t. j. vznik dvoch kategórií občanov, digitálne gramotných a digitálne negramotných

## 24. Reťazce – práca s reťazcami po znakoch

Vysvetlite pojmy: znak, znaková premenná, reťazec

*Príklad*

Napište program, ktorý načíta text telegramu zadaný veľkými písmenami a cenu za jedno slovo. Nahraďte všetky medzery v telegrame hviezdíčkou a vypíšte zmenený text na obrazovku. Nakoniec vypíšte sumu, ktorú by platil odosielateľ.

*Príklad výpisu:*

```
výpis:  Zadať text:
vstup:  zajtra bude neskoro
výpis:  Zadať cenu za slovo
vstup:  0.50
výstup:  zajtra*bude*neskoro
          Zaplatíte: 1.50 eur
```

### Úloha č.2:

a) Vypočítajte čas potrebný pre stiahnutie 2 MiB – tovej fotografie zo servera pripojeného k počítačovej sieti do počítača pripojeného k počítačovej sieti, ak prenosová rýchlosť je 64 kibps.

b) Vysvetlite pojmy: súbor, priečinok, hierarchická štruktúra, cesta k súboru.

*Úloha:*

Pomocou ľubovoľnej aplikácie alebo príkazov operačného systému zobrazte hierarchickú štruktúru koreňového priečinka (bez súborov) systémového disku.

c) Uveďte dôvody elektronizácie štátnej a miestnej správy, vysvetlite jej výhody a nevýhody.

Pojem elektronická verejná správa

Ide o komplexnú premenu existujúceho systému fungovania verejnej správy na virtuálnu inštitúciu, ktorá bude schopná poskytovať služby, riešiť požiadavky a naopak možnosť riešenia povinností občanov a firiem za pomoci internetu a ďalších moderných komunikačných prostriedkov.

Podstatou je on-line prístupný obsah a rozsah aktuálnych a správnych informácií v elektronickej podobe použiteľných na právne úkony v procesoch administratívnych, správnych, rozhodovacích, riadiacich a normotvorby v rámci celej riadiacej a výkonnej štruktúre verejnej správy.

Verejná správa prostredníctvom využívania Internetu približuje administratívu bližšie k občanom a podnikateľským subjektom. Stimulovaním prístupu k základným on-line službám

verejnej správy a k ich využívaniu môže elektronická verejná správa do značnej miery prispieť k urýchleniu prechodu na znalostnú ekonomiku. Okrem toho, proces informatizácie verejnej správy je podnetom pre transformáciu verejnej správy, ktorá môže zlepšiť verejné služby najmä ich sprístupnením a zrýchlením.

Podnikateľské subjekty môžu profitovať aj z menšieho administratívneho zaťaženia, čo môže prispieť k efektívnosti a ekonomickému rastu. V neposlednom rade transparentnejšia a interaktívnejšia verejná správa môže stimulovať aj účasť občanov na demokratickom procese.

Prechod na elektronickú komunikáciu spôsobuje veľké zmeny interných administratívnych procesov, čo môže byť zložité riadiť. Výzvou pre administratívu je teda prispôbiť sa a inovovať prístupy k práci, vrátane vhodných a stabilných partnerských vzťahov so súkromným sektorom.

Za hlavné hybné sily rozvoja informačnej spoločnosti sa považujú tri skutočnosti:

1. Objektívne narastá tlak na verejný sektor, aby poskytoval vyššiu kvalitu než doteraz. Tento tlak je spôsobený aj tým, že ľudia si zvykli na vysokú kvalitu služieb ponúkanú komerčným sektorom a analogickú kvalitu začínajú očakávať aj od verejného sektora. Stručne zhrnuté, ide o vnútorný tlak zo strany zákazníkov doma.
2. Prejavuje sa vplyv globalizácie v súvislosti s podporou hospodárskej súťaže, existencie nadnárodných trhov a potrebe podporovať prílev zahraničných investícií. V podstate je to reakcia na tlak zvonku.
3. Demografické trendy. Populácia vo všeobecnosti starne, znižuje sa percento aktívne pracujúcich v neprospech dôchodcov. Nároky na poskytované služby zo strany dôchodcov však narastajú.

Elektronizácia verejnej správy z dlhodobého hľadiska je prostriedkom na zvýšenie efektivity činnosti verejnej správy. Súčasne má obrovský potenciál na poskytovanie kvalitnejších služieb, ktoré môže poskytovať 24 hodín denne. Uvedené skutočnosti sa v plnej miere vzťahujú aj na rozvoj elektronických služieb verejnej správy. Je všeobecne známe, že dôvera zákazníkov v bezpečnosť on-line služieb sa buduje veľmi ťažko.

2. Zmeny, ktoré nastanú po zavedení e-governmentu.

1. zmenia sa základné princípy fungovania verejnej správy
2. zabezpečenie výkonu verejnej správy bez papierových dokumentov
3. výkon verejnej správy bez osobného kontaktu
4. nahradenie vlastnoručného podpisu elektronickým
5. zrušenie územného princípu a bydliska
6. zmení sa forma komunikácie

7. odstránia sa prebytoční infosprostredníci

E-government sa delí do troch kategórií.

Prvou je elektronická verejná správa, ktorá zaisťuje informácie a služby pre styk s verejnosťou.

Druhou kategóriou je elektronické verejné obstarávanie. To znamená, nákup, predaj, fakturácie alebo výberové riadenie verejnej zákazky.

Poslednou kategóriou je interná komunikácia vo verejnej správe.

3. Oblasti využitia elektronickej verejnej správy

informačné služby- poskytujú sa utriedené a klasifikované informácie na základe požiadaviek verejnosti. Skvalitní sa vzťah verejná správa - verejnosť.

komunikačné služby- ide o mnohonásobné zvyšovanie informovanosti. Komunikácia a výmena informácií medzi verejnou správou a verejnosťou.

transakčné služby- slúžia pre priame poskytovanie sľužieb alebo zasielania dát, zrýchľuje sa nimi správne konanie

integračné služby- ide tu o horizontálnu a vertikálnu integráciu sľužieb, ktorá by zabezpečila prístup k sľužbám cez jeden kontaktný bod.

Informačné služby  
Komunikačné služby  
Transakčné služby  
Bežný život  
Inform. o práci  
Vzdelávanie  
Zdravotníctvo  
Kultúra

## **25. č.1: Podmienený príkaz – viacnásobné vetvenie**

vysvetlite pojem vetvenie, viacnásobné vetvenie, prepínač

*Príklad*

Vytvorte program pre jednoduchú kalkulačku s operáciami: sčítanie, odčítanie, násobenie a delenie. V programe najprv načítajte z klávesnice prvé celé číslo, potom zadajte aritmetickú operáciu do znakovkej premennej (použite znaky +, -, \*, / ), nakoniec zadajte druhé celé číslo.

Po zadaní vstupov nech program vypočíta pomocou príkazu viacnásobného vetvenia výsledok a vypíše ho na obrazovku.

### **Úloha č.2:**

a) V inzeráte ste si prečítali, že na predaj je počítač s týmito parametrami:

Intel® Core™ i3 3.3GHz, Gigabyte® H77DS3H, ATI Radeon HD6xxx, 4 GB DDR3 1333, 500GB 7200rpm SATA 6Gb/s, DVD+/-RW DL SATA II

Vysvetlite jednotlivé skratky.

- Vysvetlite možnosti vyhľadávania informácií na Internete. Vyhľadajte na Internete koľko eur stojí pamäť RAM z predchádzajúcej úlohy.
- Ktoré informatické zamestnania poznáte a ktoré vďaka počítačom vymizli?

## 26 č.1: Aritmetika – operácie s reálnou premennou

Vysvetlite pojem reálna premenná. Vymenujte základné matematické operácie pre tento typ premennej a ako sa používa.

*Príklad*

Napíšte program, v ktorom si používateľ zvolí, či chce počítať objem alebo povrch kvádra. Podľa zvolenej možnosti užívateľ zadá požadované údaje v metroch a počítač hľadanú hodnotu vyráta. Výsledky najprv vypíšte skráteným výpisom na 2 desatinné miesta. Potom ich zaokrúhlite a vypíšte po zaokrúhlení. ( $V=a.b.c$ ,  $S=2ab+2ac+2bc$ )

*Príklad výpisu:*

```
výpis:  Zadať čo chceš počítať:
        (a) Objem kvadra (b) Povrch kvadra
vstup:  b
výpis:  Zadať rozmery kvadra a, b, c:
vstup:  5.1 2.4 3.5
výstup: S = 42.84 po zaokruhlení S = 43
```

Úloha č.2:

- Vysvetlite pojem netiketa a uveďte jej základné pravidlá.

V nasledujúcom príbehu nájdite, čo najviac porušení netikety a rozhodnite, ako by ste správne mali túto situáciu riešiť:

*„Vo svojej emailovej schránke s adresou ema@posta.sk ste objavili poštu od neznámeho odosielateľa s predmetom správy „Nase spoločne fotky“. Keďže ste ohromne zvedaví poštu otvoríte a prečítate. Z textu sa dozviete, že nejaký Ivan posielal fotky zo spoločnej dovolenky svojej priateľke Eve. Usúdite, že správne mal email putovať na adresu eva@posta.sk. S napätím si prezriete priložené fotky a objavíte pre vás veľmi vtipnú fotku. Keďže sa vám zdá zábavná, dáte si ju do pozadia a pošlete všetkým svojim kamarátom. Potom email vymažete. Keďže sa však v počítačoch vyznáte, pošlete odosielateľovi anonymný podrobný niekoľkostranový email s opisom jeho neschopnosti odoslať email.“*

## Netiketa

Termín **netiketa** vznikol spojením a skomolením dvoch slov **net** - sieť a **etiketa** - súhrn zásad spoločenského správania. **Netiketa je teda súhrn zásad spoločenského správania, ktorý by ste mali dodržiavať pri komunikácii v sieti.** Vaše správy sa môžu dostať k desiatim tisícom ľudí na celom svete. Samozrejme záleží na tóne a obsahu správy, aký dojem z vás títo ľudia budú mať. Vždy majte na pamäti, že nekomunikujete s počítačmi, ale s ľuďmi, ktorí majú city podobne ako vy.

Mnoho ľudí si myslí, že keď sa ponoria do internetového sveta je ich anonymita nekonečná. To je veľký omyl, pretože takmer vždy sa dá každý surfer vystopovať pomocou rôznych praktík a metód. Väčšina užívateľov o tom nemá ani potuchy, a preto sa správajú, resp. vyjadrujú vulgárne, robia útoky na druhé osoby, osočujú ich alebo sa z nich vysmieávajú. Toto sa často stáva na rôznych chatoch alebo internetových diskusiách. Treba si uvedomiť jednu vec, a to, že v internetovom svete by sme sa mali správať ako v reálnom svete a vyjadrovať sa ako civilizovaní ľudia.

## Pravidlá

1. Dodržiavajte všetky pravidlá slušnosti z normálneho života. Čo je zlé v bežnom živote, bude určite nevhodné aj na internete.
2. Zistite si, kde sa nachádzate. Cez internet totiž komunikujete s ľuďmi z celého sveta. A čo je v jednej skupine na internete dovolené, iná to môže považovať za neprípustné. Politika, náboženstvo a iné rozporuplné témy by mali byť diskutované s maximálnou ohľaduplnosťou a taktom.
3. Nebuďte grobianom! Aj keď píšeme bez diakritiky (bez dĺžňov a mäkčeňov) snažme sa o správny pravopis. Publikovať nepravdivé informácie, alebo niekoho ohovárať tiež nie je vhodné. Nevydávajte za svoje prácu niekoho iného. Obrázky, texty a rôzne iné súbory sa z internetu dajú ľahko stiahnuť. Akoby sa vám páčilo, keby niekto iný vydával vaše dielo za svoje? Ak využijeme prácu iných, mali by sme spomenúť ich autorstvo.
4. Rešpektujme súkromie iných. Omylom vám prišla správa, ktorá vám nepatrí? Správajte sa tak, ako by sme chceli, keby niekto iný našiel našu poštu...
5. Nezneužívajte svoju moc a vedomosti. Užívatelia so špeciálnymi privilégiami, napr. správcovia serverov ktorí majú prístup k pošte ostatných musia mať dôveru bežných užívateľov.
6. Odpúšťajte druhým chyby. Aj vy ste niekedy začínali. Nemusíme hneď reagovať výsmešne alebo so zlosťou.
7. Neširte reťazové listy a poplašné správy, **HOAX**, typu pošli túto správu x ľuďom, pretože takýmto správaním spomaľujete internet.
8. Nerozosielajte **spam**.
9. Rešpektujte autorské práva iných. Nepublikujte cudzí text pod svojim menom, vždy uvádzajte meno pravého autora a zdroj odkiaľ je text prevzatý.

## Niektoré skratky zaužívané na Internete

- **:P** - vyplazenie jazyku
- **:-D** - smiech, rehot,úsmev...
- **:-)** - úsmev, nie je myslené až tak vážne
- **:~)** - veselý úsmev
- **;-)** - potmehúdsky úsmev
- **:-I** - indiferentný úsmev
- **:-(-** - kyslý úsmev
- **:-c** - som skutočne nešťastný
- **:-\*** - posielam pusu
- **:-x** - mlčanie



b. Rozdeľte počítačové siete podľa viacerých hľadísk. Počítačové siete si môžeme rozdeliť podľa viacerých kritérií:

#### *Podľa rozľahlosti*

- **LAN** (ang. **Local Area Network**) - lokálna počítačová sieť. Označenie prepojenia počítačov v malej oblasti, napr. v jednej budove, príp. v blízkych budovách.
- **MAN** (ang. **Metropolitan Area Network**) - mestská počítačová sieť. Označenie prepojenia počítačov v meste alebo obci.
- **WAN** (ang. **Wide Area Network**) - rozľahlá počítačová sieť. Prepojenie počítačov vo veľkom meste, prípadne v okrese, kraji, štáte a kontinente.
- **PAN** (ang. **Personal Area Network**) - veľmi malá osobná sieť. Spolupracujúce zariadenia obvykle slúžia len jednej osobe (typicky prepojenie mobilu a počítača, PDA, notebooku ...) spája zariadenia rádovo v dosahu metrov. Na prepojenie sa obvykle používajú bezdrôtové technológie (WiFi, IrDA, Bluetooth).

#### *Podľa typov pripojených počítačov*

Podľa druhov počítačov použitých v sieti delíme siete na:

1. **homogénne**
2. **heterogénne**

**Homogénne siete** sú také, že všetky pripojené počítače sú rovnakého druhu. Druhom siete sa rozumie napr. sálové počítače IBM, minipočítače VAX, osobné počítače IBM PC, atď. V súčasnosti sú najviac rozšírené siete osobných počítačov IBM PC a kompatibilných.

**Heterogénne siete** môžu obsahovať viacej druhov počítačov. Existujú siete, kde sú prepojené veľké sálové počítače, minipočítače aj osobné počítače. Typickým predstaviteľom týchto sietí sú verejné dátové siete.

#### *Podľa funkčného vzťahu*

##### **Architektúra Klient-server**

Jedná sa o efektívne prepojenie viacerých počítačov, ktoré si vymieňajú veľké objemy dát.

**Server** je centrálny počítač, kde na jeho pevné disky ukladajú užívatelia svoje dáta.

**Klient** je každý počítač, ktorý je pripojený k serveru a využíva dáta uložené na ňom.

Všetky súbory v architektúre klient-server sú uložené na centrálnom počítači (server), čím sa uľahčuje správa súborov, ich zálohovanie a ochrana. Avšak ak sa vyskytne chyba servera, celá sieť prestávať fungovať.

Serverov môže byť viacero typov - podľa typu poskytovaných služieb - súborový server, tlačový server, poštový server, www server, ftp server ... nemusí platiť vzťah, že jeden server je jeden počítač. Na jednom fyzickom počítači môže existovať viacero serverov.

### Architektúra peer-to-peer

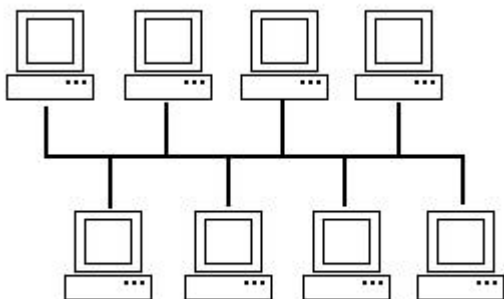
Označenie pre počítače, ktoré sú rovnocenné = **rovný s rovným**. Ozn. *p2p sieť*. Jedná sa o rýchle a lacné prepojenie počítačov navzájom. Súbory a dáta sú uložené na pevných diskoch jednotlivých počítačov. Prostredníctvom siete je možné prísť k dátam konkrétneho počítača. Avšak správa súborov a ich zálohovanie je obťažnejšie. Ak zlyhá jeden počítač, neohrozí to zvyšok architektúry.

### Podľa topológie

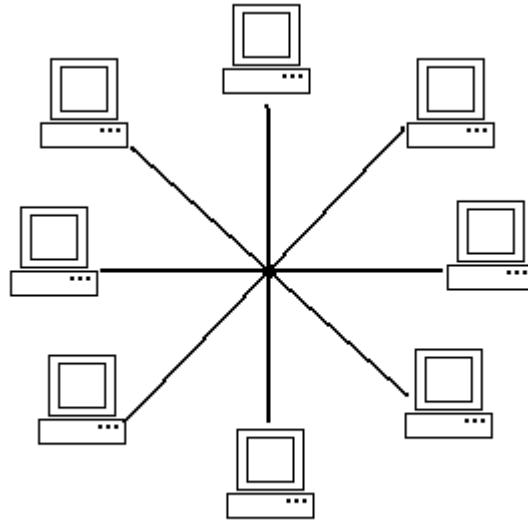
Kľúčovú úlohu v sieťach akéhokoľvek typu zohrávajú **aktívne prvky**. Ich úlohou je prepájať jednotlivé časti sietí, menia typy rozhraní, spôsoby komunikácie, zaisťujú bezpečnosť a riadia sieť. Sieť môže byť prepojená priamo z počítača do počítača, alebo s využitím týchto aktívnych prvkov.

Všetky návrhy siete vychádzajú z troch základných topológií:

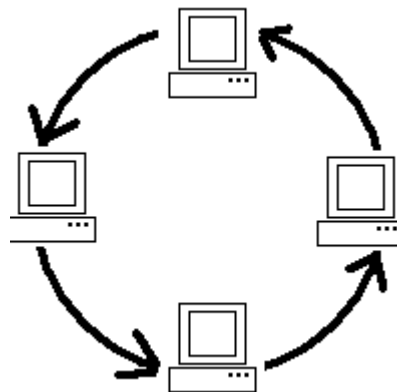
**Zbernicová topológia siete** - ak sú zapojené za sebou pozdĺž jediného kábla (segment)



**Hviezdicová topológia siete** - ak sú počítače zapojené k segmentom, ktoré vychádzajú z jediného bodu (rozbočovača)



**Kruhová topológia siete** - ak sú počítače zapojené ku káblu, ktorý tvorí prstenec



### Výhody

- Hviezda - ľahká modifikácia a pridávanie nových staníc siete
- Strom - jednoduché rozširovanie siete
- Zbernica - jednoduchá, spoľahlivá. Ľahko sa rozširuje. Jednoduché pripájanie ďalších staníc do siete.
- Kruh - rovnocenný prístup pre všetky stanice siete
- Úplná - veľmi spoľahlivá.

### Nevýhody

- Hviezda - ak zlyhá centrálny uzol - rozbočovač, zlyhá celá sieť.
- Strom - pri výpadku centrálného uzla je nefunkčný celý podstrom siete.
- Zbernica - pri porušení prenosového média je sieť nefunkčná.
- Kruh - sťažená inštalácia siete a obmedzený počet staníc v kruhu. Zlyhanie jednej stanice siete spôsobí nefunkčnosť celej siete.
- Úplná - zle rozširovateľná.

c. Pomocou ľubovoľného nástroja alebo príkazov operačného systému zistíte názov počítača, názov a verziu operačného systému vašej pracovnej stanice.

## 27. Cyklus – porovnanie cyklov

Vysvetlite pojem cyklus, akými spôsobmi sa deklaruje v jazyku C, vysvetlite podmienky behu cyklu pre jednotlivé prípady.

*Príklad*

Napište program pre konverziu čísla z desiatkovej číselnej sústavy (base-10) do dvojkovej číselnej sústavy (základ-2), veľkosť čísla má 32 bitov. Na vstupe zadáte celé číslo v desiatkovej sústave a výstup bude v binárnom kóde.

### Úloha č.2:

- a) Obhájte potrebu šifrovania informácií a demonštrujte použitie symetrického a asymetrického šifrovania na jednoduchom príklade.

Úloha:

Dešifrujte text pomocou Cézarovej šifry pre  $n=3$  (n-posunutie):

**OBÚWŠKÚB**

Pomôžte si abecedou z obrazovej prílohy.

## Šifrovanie

Kódy veľmi často používajú vojaci, námorníci alebo policajti (kód 75 – naháňam zlodeja).

**Pokiaľ je účelom kódovania aj určité utajenie informácií, hovoríme o šifrovaní.**

**Kryptológia** – veda zaoberajúca sa šiframi. Kryptológia pozostáva z **kryptografie**, ktorej cieľom je vytvoriť nerozlúštiteľnú šifru a **kryptoanalýzy**, ktorá ju má rozlúštiť. Pokiaľ chceme údaje zašifrovať, potrebujeme na to minimálne šifrovací/dešifrovací **algoritmus** – postup, na základe ktorého sa pôvodná správa zmení na zašifrovanú. Väčšina algoritmov vyžaduje na šifrovanie a dešifrovanie **kľúč**. Cieľom šifrovania vždy bolo nájdenie takej **šifry**, ktorú by nezasvätený nemohol rozlúštiť.

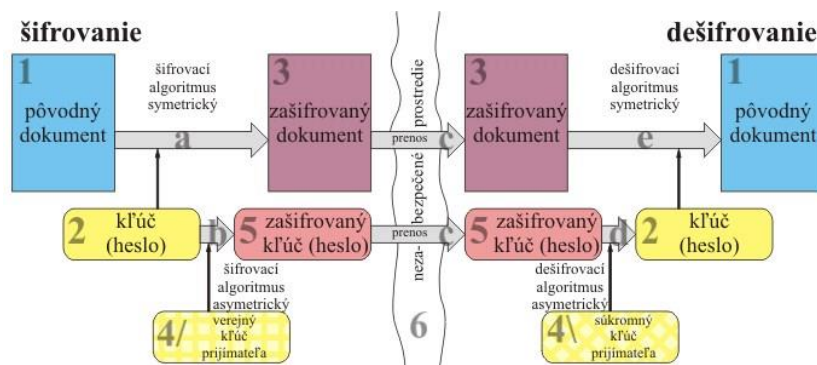
Šifry:

- Scytale – šifrovací valec,
- Cézarová šifra – spočíva v posúvaní znakov o zadanú hodnotu,
- Vigenerová šifra – posúva znaky, ale posun je daný zložitejším algoritmom,
- transpozičná šifra – napísanie textu do tabuľky tak, aby text tiekol odhora nadol.

Súčasné šifry možno rozdeliť do dvoch základných kategórií:

- **Symetrické šifry** predstavujú kategóriu šifier, v ktorých sú šifrovacie kľúče pre šifrovanie a dešifrovanie rovnaké.
- **Asymetrické šifry** sú založené na myšlienke používania dvojice kľúčov – verejného a súkromného. Prostredníctvom jedného z nich sa správa zašifruje, prostredníctvom druhého dešifruje.

V praxi sa najčastejšie používa kombinácia symetrickej a asymetrickej kryptografie.



## Význam šifrovania:

Významy šifrovania sú:

- utajenie obsahu správy
- zabezpečenie nezameniteľnosti - integrity správy
- zabezpečenie jednoznačnosti určenia autora správy
- zabezpečenie jednoznačnosti určenia času vytvorenia správy, poprípadne i zabezpečenie jednoznačnosti určenia času doručenia a času prečítania správy oprávnenou osobou

## Druhy šifrovania:

Z tých základných informácií čo viem, je dôležité to, že spôsob šifrovania sa delí na dve hlavné skupiny.

Symetrické – to sú tie štandardné šifry, kde sa používa rovnaké heslo na zašifrovanie aj dešifrovanie.

Asymetrické – na takéto šifrovanie sú potrebné dva kľúče (heslá). Toto je trochu zložitejšie, tak si to vysvetlíme na príklade:

Chcete aby Vám ľudia posielali e-maily zašifrované, tak vygeneruje tieto dva kľúče. Verejný sa zavesí niekde na web, kde si ho ľudia môžu stiahnuť a použiť na zašifrovanie správy pre Vás. Ak by aj správu niekto po ceste odchytil, nehrozí, že by ju rýchlo dešifroval (aj keď, dá sa všetko) pomocou verejného kľúča, lebo dešifrovací kľúč – privátny, vlastníte iba Vy.

## Princíp symetrického šifrovania, jeho výhody a nevýhody

Symetrické šifrovanie je postup, ktorým jednoznačne zašifrujeme pomocou kľúča čistý text na zašifrovaný text, pričom z tohto zašifrovaného textu dostaneme pôvodný text len v prípade, že poznáme pri šifrovaní použitý kľúč.

Princíp symetrického šifrovania teda spočíva v tom, že odosielateľ aj príjemca správy zdieľajú tajný kľúč, ktorým odosielateľ správu zašifruje a ktorým príjemca túto správu aj dešifruje.

Jedným z prvých šifrovacích algoritmov bol DES (v súčasnosti sa nepovažuje za bezpečný, používa sa jeho modifikovaná verzia 3-DES). V posledných rokoch vzniklo niekoľko riešení na náhradu šifry DES, napr. IDEA, CAST, Blowfish alebo RC4.

### **Výhody a nevýhody symetrickej šifry:**

Podstatnou výhodou symetrických šifier je ich nízka výpočtová náročnosť. Algoritmy pre šifrovanie s verejným kľúčom môžu byť i stotisíckrát pomalšie. Na druhú stranu veľkou nevýhodou je nutnosť zdieľania tajného kľúča, takže sa odosielateľ a príjemca tajnej správy musia dopredu dohodnúť na tajnom kľúči.

### **Princíp nesymetrického šifrovania, jeho výhody a nevýhody**

Problém so symetrickým šifrovaním je v zabezpečení prenosu kľúča, ktorý sa musí preniesť cez nejaké médium. Elektronický kanál je ľahko odpočúvateľný, fyzický prenos je na druhej strane veľmi pomalý. Nesymetrické šifrovanie tento problém rieši veľmi efektívne. Je založené na jednoduchej myšlienke: správa je dešifrovaná iným kľúčom než bola šifrovaná. Každý z komunikujúcich partnerov vlastní dvojicu kľúčov, jeden tajný tzv. privátny, a jeden verejný. Správa je zakódovaná verejným kľúčom, ktorý je distribuovaný všetkým partnerom dotyčnej osoby, ale táto osoba môže správu dekodovať len svojim privátnym kľúčom.

### **Výhody a nevýhody nesymetrickej šifry:**

Nevýhodou asymetrického šifrovania je pomalosť, ktorá je zapríčinená zložitou výpočtu. Na druhej strane je to bezpečnejší spôsob ako ponúka symetrické šifrovanie. **Princíp a možnosti využitia „hash“ funkcií**

Symetrické a asymetrické šifrovanie sa zaoberá hlavne problematikou utajovania dát.

Ďalším problémom je však integrita (neporušenosť) dát. Na jeho vyriešenie sú používané jednosmerné (hash) funkcie. Tieto transformujú ľubovoľne dlhý reťazec znakov na reťazec pevnej dĺžky (fingerprint). Porovnaním fingerprintu pôvodnej a doručenej správy je možné zistiť integritu prenášaných dát. Na hashovacie funkcie sú kladené nasledujúce požiadavky:

K danému výstupu (fingerprintu) prakticky nie je možné zostrojiť pôvodný dokument, (matematicky vyjadrené: neexistuje inverzná funkcia k hash funkcii), prakticky neexistujú dva dokumenty, ktoré majú rovnakú hashovaciu hodnotu (fingerprint), ak bol zmenený jeden bit v dokumente, fingerprint sa oproti pôvodnému musí zmeniť viac ako v jednom bite.

### **Využitie:**

Hash funkcie sú dôležité aj pri ukladaní prístupových hesiel do informačných systémov, pri generovaní tzv. One Time Passwords (hesiel na jedno použitie).

Všetko z tohto má veľmi dôležité miesto v informačných technológiách, hlavne v elektronickom bankovníctve. Sú aj súčasťou tvorby digitálneho podpisu.

- b) **Naplánujte pomocou možností OS Windows na Vašom počítači spustenie webového prehliadača hneď po prihlásení na užívateľské konto.**
- c) **Definujte pojem počítačový vírus. Vymenujte typy počítačových vírusov.**

Počítačový vírus je jednou z mnohých hrozieb bezpečnosti a integrity počítačových systémov. Z programátorského hľadiska je počítačový vírus počítačový program, ktorý môže infikovať iný počítačový program takým spôsobom, že do neho skopíruje svoje telo, čím sa infikovaný program stáva prostriedkom pre ďalšiu aktiváciu vírusu.

Pojem vírus definoval Fred B. Cohen na Pennsylvánskej univerzite, kde experimentoval so samoreplikujúcim sa kódom. Prvé publikácie v oblasti vírovej problematiky sa datujú do rokov 1984-85. V súčasnosti poznáme viac ako 30 000 rozličných, viac či menej nebezpečných, vírusov.

Počítačový vírus je špeciálna programová aplikácia, ktorá je schopná vytvárať kópie. Tato aplikácia má vždy svojho autora. Aby zabezpečil svoje spúšťanie, pripája sa k súborom na disku alebo sa ukladá do systémových oblastí počítača. Vírusy potom môžu neočakávane meniť činnosť programu a môžu mať až charakter naprogramovaného sebazničenia súborov programových aplikácií s časovým oneskorením – tzv. časovaná bomba. Vírusy, ktoré prenikli do počítača za pomoci nevinných súborov – tzv. trójske kone, môžu napríklad zničiť ochranu heslom. Ak je počítač pripojený k sieti Internet môže byť ľahko napadnutý hackerom – človek, ktorý má veľké poznatky o sieti Internet a jeho snahou je deštrukcia súborov alebo vykonanie čo najväčších škôd v databázach počítačov.

### **Vlastnosti počítačového vírusu**

- Nutnosť hostiteľa:
  - a) systémová oblasť disku (zväčša boot sector)
  - b) spustiteľné programy (väčšina vírusov)
- Malá veľkosť víroveho programu
- Naprogramované priamo v assembleri (hlavne v minulosti)
- Symbióza s hostiteľom (vírus najmä vo svojom vlastnom záujme zabezpečuje po vykonaní vírových inštrukcií aj vykonanie tela pôvodného infikovaného programu)

### **Delenia počítačových vírusov:**

podľa umiestnenia v pamäti:

- rezidentné (ostávajú v operačnej pamäti počítača aj po ukončení vykonávania infikovaného programu)
- nerezidentné

podľa cieľa infekcie:

- bootovacie

- súborové
- clustrové

podľa chovania:

- stealth (schovávajú sa a kódujú)
- polymorfné (schopné meniť svoj kód, ktorým sa ďalej rozširujú)

Vírusy nie sú jediné hrozby, ktoré ohrozujú bezpečnosť počítačov. Im podobné sú:

Trójsky kôň: Tento program najčastejšie okamžite po svojom spustení prevádza deštrukčnú rutinu. Častokrát trójske kone slúžia na vypustenie nového vírusu.

Makro-víry: Ide o programy naprogramované v jazyku na tvorbu makier v textovom procesore, alebo tabuľkovom kalkulátore a vložené do takého dokumentu. Kvalitný programovací jazyk im umožňuje replikáciu, ale je diskutabilné, či ich radiť medzi vírusy, pretože neobsahujú inštrukcie procesora.

Červy-worms: Takýto program neinfikuje spustiteľné súbory, ale rozširuje sa počítačovou sieťou. Červ nepotrebuje hostiteľa.

Bomby: Programy, ktoré po spustení čakajú na aktivačný podnet (tzv. rozbuška), zväčša kľúč z klávesnice, zmena nejakého súboru, aktuálny dátum, alebo čas, a prevedú deštrukčnú rutinu.

## Prejavy vírusov

Základným prejavom je schopnosť šíriť sa. Deštruktívne vírusy najčastejšie formátujú pevný disk, prepisujú náhodne vybrané sektory náhodnými dátami, menia obsah súborov, mažu súbory, zašifrujú dáta, atď. U nedeštruktívnych vírusov majú aktivačné rutiny najčastejšie charakter vizuálnych prejavov (zobrazovanie rôznych textových správ) a akustických prejavov (drvivou väčšinou je nástrojom takéhoto prejavu zabudovaný reproduktor - PC-speaker).

## Antivírové prostriedky a mechanizmy prevencie a liečby

Reakciou na vznik počítačových vírusov boli rôzne antivírové programy. Na detekovanie vírusov používajú rôzne mechanizmy:

Scanner: scanner zisťuje prítomnosť víru v pamäti, alebo na disku pomocou vírových identifikačných reťazcov. Vírový identifikačný reťazec je jednoznačne definovaná postupnosť bytov reprezentujúcich daný vírus. Na zvýšenie účinnosti sa používa viac reťazcov na jeden vírus naraz.



Rezidentný štít: rezidentný štít je program, ktorý beží v reálnom čase. Najčastejšie je to program typu scanner, ktorý prevádza antivírusovú kontrolu práve spracovávaných dát. Môže tak zakázať skopírovanie zavírených súborov z diskety na pevný disk, či zakázať spustenie infikovaného programu a pod.

Monitor diskových zmien: monitor sa zameriava na sledovanie zmien stavu v počítači, najmä spustiteľných súborov. Uchováva databázu popisov základných vlastností súborov, najmä dĺžku, dátum a čas poslednej aktualizácie a hlavne kontrolný súčet. Slabinou je prípad vírusov zameraných na zmazanie kontrolnej databázy.

Základnou a spoľahlivou metódou prevencie je zálohovanie. Najvhodnejšie je zálohovanie viacerých verzií na viac médií naraz. Keďže sú málo rozšírené vírusy napádajúce komprimované archívy, je vhodné do takýchto archívov svoje zálohované údaje ukladať. Pri liečbe je vhodné používať viac druhov softwaru naraz. Vždy treba mať k dispozícii nezavírenú systémovú disketu obsahujúcu základné vybavenie pre čistenie master boot record (FDISK s prepínačom /MBR), boot sektora (SYS C:) spolu so scannerom, heuristickým analyzátorom a cleanerom. V každom prípade netreba pri nákaže prepadnúť panike, ale zistiť čo najviac informácií o víruse pred podniknutím ďalších krokov a v prípade neistoty prenechať odvírenie skúsenejšiemu antivírusovému odborníkovi.

Vírusy sa šíria aj Internetom a elektronickou poštou, najčastejšie však otvorením nejakej neznámej pošty alebo prílohy v emaili. Je preto potrebné neotvárať neznámu poštu alebo aspoň prílohy v nej. Najnebezpečnejšie vírusy ako bol napríklad I love you sa šírili práve týmto spôsobom. Veľmi dôležité sú preto aj firewally.

## 28. Cyklus – riadiaca premenná cyklu

Vysvetlite pojmy: cyklus, riadiaca premenná cyklu. Akým spôsobom dokážeme v príkaze cyklu s podmienkou doplniť neexistujúcu riadiacu premennú cyklu?

*Príklad*

Vytvorte program, ktorý na začiatku načíta od užívateľa kladné celé číslo väčšie ako 1 a na konci vypíše či je toto číslo prvočíslo (deliteľné len samo sebou a číslom 1) alebo zložené číslo (deliteľné aspoň jedným ďalším číslom).

Doplňte program tak, že v prípade zadania nevyhovujúceho čísla (menšieho ako 2), musí užívateľ zadať znova. Po treťom chybnom zadaní nech program skončí.

### Úloha č.2:

- a) Vysvetlite pojmy: súbor, priečinok, hierarchická štruktúra, cesta k súboru.

*Úloha:*

Na ploche vytvorte priečinok s názvom **Maturita**. Skopírujte doň všetky súbory s príponou **.doc** z priečinku Všetci študenti/Maturita, jeden ľubovoľný súbor premenujte na **ahoj.doc**. Vymažte všetky súbory okrem súboru ahoj.doc.

Nakoniec zmažte priečinok Maturita z plochy a znova ho obnovte z Koša.

Súbor je v PC (najmenšie možné) logické zoskupenie údajov (dát). Každý súbor má svoj názov, ikonu, presnú adresu na disku alebo diskete, CD-čku.

Súbory zvyčajne rozdeľujeme na dve skupiny:

- a) Programy (aplikácie) – *súbory* vytvorené programátormi, po ich nainštalovaní na disk a spustení môžeme vykonávať činnosti, ktoré umožňujú.
- b) Dokumenty – *súbory*, ktoré sami vytvoríme, upravíme, prečítame a pod. Ide o údaje, ktoré spracovávame, preto takéto súbory označujeme aj ako *dátové*.

Priečink (adresár - directory, zložka – folder) Priečink slúži na prehľadné ukladanie súborov na disku. Každý priečink má svoje meno a je k nemu presne definovaná adresárová cesta. Priečinky majú stromovú štruktúru. Ako prvý je hlavný adresár a v ňom sa môžu nachádzať ďalšie priečinky, a v nich ďalšie...

Ku každému súboru existuje **jednoznačná adresárová cesta** – vytvoríme ju zapísaním všetkých priečinkov od vrcholu hierarchického stromu až k priečinku v ktorom je súbor uložený napr: D:\Dokumenty\Praca\TVVP

Priečinky umožňujú prehľadné uloženie (usporiadanie) súborov. Zabezpečujú lepšiu orientáciu užívateľa v pamäti počítača, pretože jednotlivé aplikácie (programy) a dáta (tvoriace dokumenty) sú umiestnené v jednotlivých priečinkoch v ktorých ich je možné ľahko nájsť.



Obr. 16 Adresárová štruktúra

Ikona je malý farebný obrázok, ktorý vidíme na pracovnej ploche a obrazovke počítača.

**b) Uveďte a stručne charakterizujte najpoužívanejšie služby Internetu.**

Vymenujte najznámejšie internetové služby a stručne ich popíšte

HTTP (HyperText Transfer Protocol)

Ide o protokol, ktorý sa využíva na prenos dát na internete v architektúre klient – server. Prenášajú sa ním najmä stránky o ktoré cez prehliadač požiada klient.

## FTP (File Transfer Protocol)

Ide o internetový protokol využívaný na prenos súborov medzi počítačmi. Môže sa použiť napríklad na sťahovanie voľne šíriteľných programov z nejakého servera, alebo o diaľkové nahrávanie stránky na server. Využíva telnet.

## Email

Elektronická pošta. Je druhou najpoužívanejšou službou. Email nám slúži na komunikáciu ako klasická pošta, s tým rozdielom, že nemusíme platiť poštovné, musíme mať nejakú emailovú schránku na nejakom servere a je to aj ekologickejšie, pretože nemíňame papier a teda šetríme lesy.

Je to jeden z najrýchlejších spôsobov posielania pošty, hoci niekedy môže byť aj nebezpečný.

## TCP/IP

Siete LAN sú homogénne, čiže väčšinou ide o sieť počítačov v jednej budove, či na jednom poschodí, čo spôsobuje, že metóda store-and-forward (ulož a odovzdaj) protokolu UUCP nie je dostatočne pružná.

V týchto sieťach sa používa protokol TCP/IP, ktorý používa metódu packet-switched (prenos paketov), čo znamená, že informácie, ktoré sa posielajú po sieti, aj po internete sa najprv rozkúsujú, potom sa posielajú priamo cieľovému hostiteľovi a ten ich opäť poskladá. Protokol TCP/IP sa používa na komunikáciu medzi počítačmi v sieťach LAN, ale aj na internete.

## Videokonferencia

Je vzdialená interaktívna komunikácia medzi dvomi a viacerými účastníkmi, pričom dochádza medzi nimi k prenosu zvukovej a obrazovej informácie (účastníci sa vidia aj počujú).- multimediálny počítač – počítač, ktorý vie pracovať s viacerými typmi dát (nielen textovými a grafickými, ale i zvukovými)- videokonenčná aplikácia (NetMeeting) – okrem hardwaru a rýchleho sieťového pripojenia je potrebné mať na počítači nainštalovaný software, ktorý umožní realizovať videokonferenciu.

## Služby fungujúce v reálnom čase

chat – rozhovor

IRC – internet relay chat – zasa klient-server; posiela celé vety; je nepríjemná, že musí mať aj program

WWW (World Wide Web - WWW alebo W3) je sieťová informačná služba vychádzajúca z Internetu a ponúkajúca systém zbierania informácií vo forme hypertextových multimédií. Web ponúka metódu zobrazenia informácií rôznych formátov spôsobom rýchlym, výkonným, konzistentným a ľahko pochopiteľným.

## 3. Objasnite princíp služby WWW

Ako prostriedok prístupu na Internet je WWW jedným z najjednoduchších dostupných

systemov. Web je sieťový systém klient/server. Počítač, ktorý informácie dodáva sa nazýva server. Počítač, ktorý informácie zobrazuje sa nazýva klient. Pre webovský klient aj server potrebujeme zvláštny softvér.

Ako príklady prehliadačov možno uviesť Netscape Navigator, Mosaic, Internet Explorer.

#### 4. Vymenujte a popíšte činitele, ktoré musí zohľadniť tvorca web stránok pri ich zostavovaní vo vzťahu na prenosové trasy a technické vybavenie klienta

Tvorca by mal predovšetkým každú web-stránku otestovať približne takýmto testom: Týmto testom budú testované stránky z pohľadu prístupnosti, použiteľnosti, optimalizácie pre vyhľadávače a technických parametrov.

##### Test prístupnosti

Týmto testom zistíme, či sa stránka správne zobrazuje v rôznych prehliadačoch, ovládanie klávesnicou, či je optimalizovaná pre tlač, či každý netextový prvok má textovú alternatívu.

##### Test použiteľnosti

Týmto testom bude zistené, ako ľahko sa dá na stránkach orientovať, či má jasnú navigáciu, či je obsah zrozumiteľne delený do nadpisov a zoznamov.

##### Test optimalizácie pre vyhľadávače

Tento test slúži na zistenie, ako stránka spolupracuje s vyhľadávачmi. Či sú správne napísané meta tagy, či sú správne vybrané kľúčové slová, či na stránku vedie dostatočný počet spätných odkazov.

##### Test technických parametrov

Týmto testom bude zistená validita kódu a kaskádového štýlu, funkčnosť všetkých odkazov, dĺžka načítavania stránky a pomer obsahu ku celkovej veľkosti stránky.

Dodržiavaním štandardov pre tvorbu web stránok, by tvorca webu nemal prísť do sporu s klientom, čo sa týka hlavne prenosových tras a technického vybavenia.

Stránky by preto nemali obsahovať priveľké množstvo nadbytočných vecí, ktoré potom zbytočne zaťažujú a zväčšujú veľkosť web-stránky. Čím väčšia je stránka, tým je aj čas na jej zobrazenie dlhší a to môže byť práve jeden z dôvodov na jej odradenie, pretože loading - načítavanie stránky trvá hodnú chvíľu. Takéto stránky su zaťažované predovšetkým na grafiku a dizajn. Tak isto by sa mala zohľadňovať aj výkonnosť, ktorá určuje množstvo času a námahy, ktorú musí užívateľ vynaložiť pri prezeraní stránky.

#### 5. Popíšte rôzne vyhľadávacie nástroje na vyhľadávanie informácií na Internete

- katalógy: [www.yahoo.com](http://www.yahoo.com), [www.zoznam.sk](http://www.zoznam.sk), [www.atlas.sk](http://www.atlas.sk), [www.superzoznam.sk](http://www.superzoznam.sk), [www.best.sk](http://www.best.sk), [www.surf.sk](http://www.surf.sk)

Vyvinuli sa zo špecializovaných zoznamov adres, kt. doplnili autory odkazmi na kategórie. Výhoda jednoduchej orientácie. Málokedy sa len náhodne aktualizujú, obsahujú staré infos (nevýhoda). S pribúdaním nových WEB stránok sa katalógy rozšírili, stali sa neprehľadnými a vznikli hľadania podľa kľúčových slov.

- hybridné vyhľadávacie stroje

[www.excite.com](http://www.excite.com), [www.lycos.com](http://www.lycos.com), [www.webcrawler.com](http://www.webcrawler.com), [www.kompac.seznam.cz](http://www.kompac.seznam.cz)

- metavyhľadavacie nástroje  
Príjmu žiadosť a pošlú ju ďalším vyhľadávateľom

#### WWW.YAHOO.COM

Jeden z prvých metainformačných systémov ITE. Používa ho 400.000 ľudí denne. Je to predmetový adresár. Vyhľadáva sa cez systém:

- a) browse (podľa hierarchickej štruktúry)
- b) search (podľa kľúčových slov. Umožňuje kombináciu hľadaných termínov pomocou OR, AND \* - len na pravo

#### WWW.WEBCRAWLER.COM

- 1. Browse
- 2. Search

Je to prvý nástroj, kt. umožňuje vyhľadávať v úplných textoch dokumentov. Denne je schopný odpovedať na 3 mil dotazov. Stačí napísať len slová, kt. vystihujú predmet rešeršnej požiadavky.

#### WWW.ALTAVISTA.COM

Je to stroj od 1995. Denne vybaví 12. mil dotazov. Umožňuje vyhľadávať vo www stránkach a el. konferenciách. Vyhľadávanie typu BROWSE a SEARCH. Umožňuje jednoduché vyhľadávanie +, -, \*, zdokonalené vyhľadávanie v BOOLOVEJ algebre AND, OR, NOT, NEAR

#### WWW.METACRAWLER.COM

Umožňuje paralelne vyhľadávanie pomocou strojov. Nemá vlastnú internú bazu dát, ale využíva BD iných systémov. Je hradený sponzormi a využíva vyhľadávanie v 9 systémoch. yahoo, opentext, lycos, webcrawler, altavista, galaxy, infoseek, inktomi.

6. Uvedte príklad na zadanie požiadavky na vyhľadanie informácie s použitím kľúčových slov link, domain

Pri vyhľadávaní je možné využiť: Spojenie kľúčových slov pomocou booleovských operátorov (AND, OR, NOT) s využitím vyhľadávacích polí.

Hviezdičku \* - pravostrané rozšírenie slova – súbežné vyhľadanie rôznych tvarov slov (napr. manage\* – systém vyhľadá manager, management, managers apod.)

Hviezdička \* - umiestnená uprostred slova, namiesto akéhokoľvek počtu písmen (napr. h\*emoglobin – systém vyhľadá hemoglobin, haemoglobin apod.)

Otáznik ? – umiestňujeme na ktoromkoľvek mieste v slove, alebo na konci slova namiesto jedného písmena (napr. analy?e – systém vyhľadá analyse alebo analyze).

Úvodzovky “” – môžeme použiť pre vyhľadanie presného slovného spojenia (napr. "specific heat").

Úvodzovky “” - môžeme použiť i pre vyhľadanie špecifických cudzích písmen v slove a

interpunkčných znamienok .

Proximitné operátory – sú tvorené slovami PRE/n a W/n. (Za n dosadzujeme ľubovoľné číslo od 1 do 255).

Proximitný operátor pre /n – obmedzuje vzdialenosť hľadaných termínov v poradí v akom boli zapísané (napr. pain pre/6 morphine - systém vyhledá slová v rozpätí 6 slov).

Proximitný operátor w/n – umožní vyhledat' dve slová, nachádzajúce sa v jednej vete, v jednej fráze alebo odseku (napr. pain w/5 morphine – systém vyhledá obidve slová vzdialené od seba v rozmedzí 5 slov).

Pre vyhledanie termínov v jednej fráze treba použiť operátor w/3, w/4, w/5., pre vyhledanie termínov v jednej vete treba použiť operátor w/15 a pre vyhledanie termínov v jednom odseku treba použiť operátor w/50).

a) Basic Search (Základné vyhledávanie)

b) Author Search (Vyhledávanie podľa autora)

c) Advanced Search (Pokročilé vyhledávanie)

Dobré umiestnenie vo výsledkoch vyhledávania je základom, aby ľudia našli vašu lokalitu. Mnoho ľudí nepoužije z výsledkov vyhledávania viac než prvú stránku. Byť na začiatku zoznamu je navyše lepšie, ako byť na jeho konci.

## 7. Popíšte využitie e-bankingu v praxi

Elektronické bankovníctvo je na slovensku pomerne nový, ale bezpochyby efektívny spôsob komunikácie klienta so svojou bankou. Pod pojmom elektronické bankovníctvo sa skrýva široká paleta služieb, ktorá pomocou moderných technológií umožňuje uskutočniť najbežnejšie operácie ako sú napríklad: zistenie stavu účtu, jednotlivých pohybov na účte, zadanie platobných príkazov a taktiež objednávanie si niektorých služieb u Vašej banky. Všetky tieto služby môžete vykonávať bez nutnosti návštevy pobočky Vašej banky, stačí ak o ich poskytovanie požiadate u Vašej banky. Bankami v SR sú v súčasnosti poskytované nasledujúce produkty elektronického bankovníctva: Internet banking, Mobil banking, ePay, Mailbanking

### **c. Čo je to *hoax*? Pri akej komunikácii sa s ním môžete stretnúť**

Hoax (číta sa houks) znamená predovšetkým elektronicky šírenú správu s nezmyselným obsahom. Často nabáda na preposielanie a čo je horšie, často aj preposielaná je.

Najčastejšími témami hoaxu sú:

1. reťazové maily, ktoré sľubujú šťastie, prípadne aj peniaze, väčšinou za preposielanie;
2. varovanie pred nebezpečenstvom poškodenia počítača najmä v dôsledku nebezpečného vírusu šíriaceho sa mailom, obsahujúci predmet s istým názvom;
3. prosba o pomoc v núdzi – najčastejšie darovanie krvi;
4. varovanie pred ovládnutím alebo zničením mobilného telefónu;
5. žiadosť o podporu nejakej petície ap.

Prvý druh hoaxu je našťastie na ústupe. Avšak varovný a prosebný typ hoaxu stále dokáže obalamutiť mailových užívateľov. Lebo zatiaľčo rozosielenie mailov so sľubovaným šťastím je už často chápané ako obťažovanie, druhé dva vychádzajú z dobrých ľudských vlastností – túžbe pomôcť alebo varovať pred nebezpečenstvom.

Klasická schéma varovného mailu informuje o šírení nejakého vírusu, ktorý môže poškodiť dáta alebo priamo časť hardvéru, napr. zničenie pevného disku, občas aj explóziu počítača. Niekedy pre dôveryhodnosť sa obsah tvári, akoby išlo o vyhlásenie nejakej veľkej IT alebo bezpečnostnej inštitúcie (napr. Microsoft, IBM, Polícia SR ap.), neraz býva podpísaná aj konkrétnym menom.

Stačí vedieť, že vírusové aplikácie len veľmi zriedka (najmä pri starých typoch počítačov) môžu poškodiť hardvér. Okrem toho, ako prvé reagujú na vznik vírusov antivírusové programy, ktoré sú už dnes bežne integrované v samotných mailových kontách. Takže, čokoľvek upozorňuje na vírus, ide spravidla o hoax. Preposielaním takejto správy nikomu nepomôžeme, iba ho môžeme obťažovať.

Typickou schémou prosby o pomoc býva veľmi stručný srdcervúci príbeh o umierajúcom dieťati, ktoré potrebuje napr. urgentne zásoby krvi. Občas je takýto mail doplnený aj autenticky pôsobiacou fotografiou.

Tu je dobré si uvedomiť, že ak by aj išlo o skutočnú prosbu, tak na jej reakciu treba spravidla veľmi krátky čas (niekoľko hodín, prípadne zopár dní). Takéto maily dokážu kolovať po internete aj niekoľko rokov. Stačí sa zamyslieť, aby sme dokázali odlíšiť skutočnú prosbu od fiktívnej. Fiktívna prosba sa snaží zachytiť čo najviac ľudí a preto reálna akcia musí byť čo najviac vymedzená svojím miestom (napr. okres Ilava), podmienkami (iba krvná skupina typu 0), ale najmä časom – odkedy to predmetná osoba potrebuje. Takže najneskôr do zopár týždňov po danom dátume by sa mala reťaz zastaviť. Väčšinou takáto správa obsahuje aj mobilné telefónne číslo, ktoré je spravidla hluché. Málokto si však dá tú námahu, aby si to číslo preveril a správu hneď preposiela.

## **Ako sa k hoaxu správať?**

V prvom rade si treba uvedomiť, že pri danej správe máme do činenia s hoaxom. Existuje internetový portál – [hoax.cz](http://hoax.cz) – ktorý robí nielen v tejto veci výbornú osvetu, ale aj obsahuje databázu hoaxových správ. Tu je možné svoju dilemu či ide o hoax často priamo konfrontovať. Po prečítaní znení zopár hoaxových správ človek rýchlo pochopí klasické znaky hoaxu a naučí sa ich ľahko sám identifikovať.

Treba byť zvlášť opatrný a obozretný, pokiaľ hoaxová správa navádza nejakým spôsobom k poskytnutiu súkromných informácií. Môže ísť o vyplnenie petície na nedôveryhodnom serveri alebo len výzvu, že stačí poslať prázdnu správu na nejaký mail.

Netreba snád' zdôrazňovať, že daný hoax určite nie je vhodné preposielať. Navyše, na prosbu o fiktívnu pomoc možno reagovať skutočnou a jednoduchou pomocou. Napr. informovať odosielateľa hoaxu o tom, že posiela hoax a odkázať ho na spomínaný portál ([hoax.cz](http://hoax.cz)), zároveň ho aj vyzvať k preštudovaniu webu, aby bol v budúcnosti za počítačom prezieravejší.

## 29. Cykly – porovnanie cyklov

Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?

*Príklad*

Naprogramujte hru „Hádaj číslo“. Počítač zvolí náhodné celé číslo z intervalu 1..100 a priradí ho do premennej *los*. Hráč vždy dookola v cykle zadáva číslo do premennej *tip*, počítač mu vždy na obrazovke vypíše či uhádol a potom hráč zadáva znova. Hra končí uhádnutím čísla alebo zadaním čísla mimo intervalu 1..100. Doplňte program tak, že počítač vždy hráčovi poradí, či je hľadané číslo *los* menšie alebo väčšie ako zadané *tip*.

### Úloha č.2:

- Vysvetliť význam a spôsob používania doménových mien. Popíšte jednotlivé časti tejto adresy : *www.fei.tuke.sk* . Preverte funkčnosť spojenia s uvedeným servrom pomocou príkazu *ping* a zistite jeho *ip* adresu.

System DNS vznikol v roku 1984. DNS je skratka, ktorá vznikla zo slov Domain Name System. Je to služba Internetu, ktorá slúži na preklad doménových mien na IP adresy. Každý IP datagram je v Internete smerovaný na základe cieľovej IP adresy. Keďže IP adresy sa pre človeka ťažko pamätajú, používajú sa tzv. doménové mená. Nie každé sieťové rozhranie však musí mať pridelené svoje doménové meno.

Pokiaľ sieťové rozhranie nemá pridelené doménové meno, môžeme napísať

*http://194.160.210.18/*

alebo môžeme poslať e-mail na adresu *user@[194.160.210.18]*

Pre pochopenie systému DNS sa musíme oboznámiť s pojmami:

- doména
- doménové meno
- reverzná doména
- zóna
- name server

### Doména

Je skupina mien, ktoré k sebe logicky patria. V rámci domény je možné vytvárať podskupiny, tzv. subdomény. Z jednotlivých mien podskupín je potom vytvorené doménové meno uzla (napr.: *prometheus.student.sk*). Pri adresovaní počítača v Internete uvádzame jeho doménové meno.

### Doménové meno

Skladá sa z reťazcov vzájomne oddelených bodkou. Meno sa preskúmava smerom sprava doľava. Najvyššou inštanciou v mene je root doména a je označená samostatnou bodkou. V root doméne sú definované domény najvyššej úrovne (TLD, Top Level Domain). Medzi TLD patria domény *aero*, *arpa*, *biz*, *com*, *coop*, *edu*, *gov*, *info*, *mil*, *museum*, *name*, *net*, *org*, *pro* a dvojznakové kódy štátov definovaných podľa normy ISO-3166 (*sk*, *cz*, *at*, atď). Pod TLD doménami sú definované ďalšie





## Name server

Name server prekladá doménové mená na IP adresy a opačne. Name servery rozdeľujeme do štyroch skupín:

### 1. primárny name server

- Na primárnom name servery sú uložené všetky údaje o danej zóne.
- Primárny name server môže byť len jeden.
- Akékoľvek zmeny zóny sa vykonávajú na primárnom name servery.
- Primárny name server poskytuje autoritatívne odpovede k údajom o zóne, ktorú spravuje.

### 2. sekundárny name server

- Sekundárny name server preberá v určitých časových intervaloch údaje o zóne od primárneho name servera.
- Počas výpadku alebo nedostupnosti primárneho name servera poskytuje autoritatívne odpovede o zóne, ktorú spravuje.
- Počet sekundárnych name serverov nie je obmedzený.
- Sekundárny name server poskytuje autoritatívne odpovede k údajom o zóne, ktorú spravuje.
- 3. caching only name server
- Nie je pre žiadnu doménu primárnym, ani sekundárnym name serverom, avšak využíva všeobecné vlastnosti name servera. To znamená, že do svojej pamäte ukladá údaje, ktoré ním prechádzajú. Poskytuje vždy neautoritatívne odpovede.

### 4. root name server

- Root name server je špeciálny typ name servera.
- Spravuje údaje o TLD.
- Každý z root serverov je primárnym name serverom, aj keď každý obsahuje rovnakú databázu údajov o zóne.
- Vo svete existuje 13 root name serverov. Ich zoznam nájdete na adrese <http://www.root-servers.org/>
- Výpadok týchto name serverov by znamenal výpadok siete Internet na úrovni DNS. Bez znalosti IP adresy cieľa by sme nevedeli adresovať počítač v sieti.
- V roku 2002 sa podarilo 1 hodinovým DDoS (distributed denial-of-service) útokom odstaviť 9 name serverov, avšak vďaka tomu, že root servery sú pripojené na tie najrýchlejšie linky v sieti Internet, systém DNS bol ohrozený minimálne..

Podľa spôsobu, akým name server pracuje, rozdeľujeme name servery na:

1. forwarding - tento server nespracováva dotaz sám, ale prepošle ho inému name serveru (forwarderu), ktorý ho spracuje namiesto neho. Keď je dotaz spracovaný, forwarding server získa od forwardera odpoveď. Pokiaľ sa odpovede od forwardera nedočká spracuje dotaz sám. Forwarding server je pripojený na Internet pomalou linkou a preto posielá všetky dotazy forwarderu, ktorý je pripojený na Internet rýchlou linkou.

2. slave - pracuje rovnako ako forwarding server, ale v nikdy nerieši dotaz sám, čaká na odpoveď od forwardera.

## DNS dotaz

Aplikácia odosiela požiadavku o preklad doménového mena (DNS dotaz) name serveru prostredníctvom resolvera. Resolver je to časť operačného systému alebo aplikácie, ktorá zabezpečuje spojenie a komunikáciu s name serverom. Odpoveď, ktorú resolver získa od name servera môže byť:

### 1. autoritatívna

Túto odpoveď dostaneme, ak sa pýtame priamo primárneho alebo sekundárneho servera pre danú zónu. Tento dotaz prebieha nasledovne:

- aplikácia odošle svoju požiadavku resolveru
- resolver sa pozrie do svojej cache pamäte a keď nájde zhodu, preloží doménové meno na IP adresu
- resolver sa pozrie do súboru hosts a v prípade keď nájde zhodu, preloží doménové meno na IP adresu
- resolver kontaktuje primárny alebo sekundárny name server. Ten zistí, že má záznam pre doménu vo svojej databáze a odošle svoju odpoveď späť. Takáto odpoveď sa nazýva autoritatívna.

### 2. neautoritatívna

Neautoritatívnu dostaneme, ak dotazovaný name server musel získať informáciu o doméne od iného name servera. Samotný name server pritom obdržal autoritatívnu odpoveď. Tento dotaz prebieha rovnako ako predchádzajúci, až do chvíle, keď primárny name server zistí, že požadovaný záznam nemá vo svojej databáze. Potom dotaz prebieha nasledovne:

1. name server kontaktuje jeden z root name serverov, aby zistil, ktorý name server spravuje údaje o TLD

*searching for prometheus.student.sk A record at a.root-servers.net [198.41.0.4]: Got referral to NS1.SK-NI*

2. name server kontaktuje TLD name server, aby zistil, kto spravuje požadovanú doménu  
*Searching for prometheus.student.sk A record at NS1.SK-NIC.sk. [195.12.159.3]: Got referral to mercury.student*

3. name server kontaktuje name server pre danú doménu, získava buď preloženú adresu alebo adresu ďalšieho name servera  
*Searching for prometheus.student.sk A record at mercury.student.sk. [193.87.12.1]: Reports prometheus.student.sk.*

4. name server odosiela resolveru odpoveď (neautoritatívnu)  
*prometheus.student.sk. A IN 3600 194.160.210.18*

- b) Vysvetlite dôvody na aktualizáciu operačného systému, výhody a nevýhody automatickej aktualizácie.

*Úloha:*

Nastavte automatickú aktualizáciu operačného systému na každý pondelok o 3:00 hod.

c) *Definujte pojem počítačová kriminalita a popíšte jej formy.*

Počítačová kriminalita je relatívne novým druhom závažnej trestnej činnosti. Od klasickej kriminality sa odlišuje celým radom osobitných charakteristík a zvláštností. Trestný čin môže byť spáchaný v anonymite na diaľku, sprostredkovane a to všetko v priebehu niekoľkých sekúnd bez toho, aby poškodený zaregistroval spáchanie takéhoto trestného činu a niekedy sa o tom vôbec dozvedel. Internet, anonymita a nedostatočná legislatíva, robia z počítačovej kriminality mocný nástroj na páchanie domácich a medzinárodných trestných činov veľakrát závažného charakteru s priamym dopadom na ekonomiku krajiny a jej bezpečnosť.

Od nástupu informačného veku je stále veľké množstvo právnych noriem zastaraných alebo v praxi nevykonalných. Vznikli nové oblasti, s ktorými žiadny právny systém pred pár rokmi nepočítal. Jedná sa najmä o zákony, súvisiace s uplatňovaním autorských práv a zákony súvisiace s pohybom informácií.

V oblasti autorského práva to bola digitalizácia, ktorá spôsobila revolúciu a umožnila lacné vytváranie kópií digitálnych dokumentov. Veľa spoločnosti, kopíruje a vymieňa digitálne dokumenty, chránené autorským právom (film, hudba, program) v dokonalej kvalite za účelom finančného profitu.

V dôsledku vplyvu najznámejších internetových stránok sa problémy objavujú aj v oblasti médií. Tieto stránky majú rovnaký mediálny vplyv ako tlačené noviny, časopisy, alebo televízia, ale štát nie je vlastníkom internetu a tým na internete nie sú obmedzené zdroje, ktoré by bolo možné pridelovať. Je to nová situácia, keď štát bude musieť nanovo definovať problém regulácie médií.

Úprave sa nevyhne ani pracovné právo v súvislosti s teleworkingom. Je to progresívna forma práce a je potrebné ju zrovnoprávniť s ostatnými formami práce. So zvyšujúcim sa objemom obchodovania cez internet, súvisí aj problematika daní.

Najzávažnejším problémom stále ostáva problematika riešenia internetovej kriminality všeobecne vo všetkých jej známych podobách na všetkých stupňoch závažnosti. Aby sme mohli hovoriť o počítačovej kriminalite, musí páchatel' k svojmu jednaniu použiť výpočtovú techniku, prípadne iný prostriedok na spracovanie informácií. Jeho konanie musí napĺňať skutkovú podstatu trestného činu. Objasňovanie a dokazovanie takejto trestnej činnosti je vo všetkých prípadoch tak zložitá, že bez včasného zaistenia stôp a dôkazov, je len malá úspešnosť zistenia, zadržania a odsúdenia páchatel'ov. Dôvodom je najmä používanie metód a prostriedkov na vysokej technickej a intelektuálnej úrovni.

Celý tento problém má zložitú legislatívne pozadie. Internet nemá žiadneho vlastníka, ktorého by bolo možné postihovať. Okrem toho je internet nadštátny a nadnárodný teda nespadá do jurisdikcie žiadneho štátu. Možnosťou by bola koordinovaná globálna regulácia. Ale aj tu sa môžeme stretnúť s potenciálom zneužitia regulačných nástrojov v rukách diktátorov a mocenských režimov.

Termínom **počítačová kriminalita** sa označujú trestné činy zamerané proti počítačom ako aj trestné činy páchané pomocou počítača. Ide o nelegálne, nemorálne a neoprávnené konanie, ktoré zahŕňa zneužitie údajov získaných prostredníctvom výpočtovej techniky alebo ich zmenu. Počítače v podstate neumožňujú páchať nový typ trestnej činnosti, iba poskytujú novú technológiu a nové spôsoby na páchanie už známych trestných činov ako je sabotáž, krádež, zneužitie, neoprávnené užívanie cudzej veci, vydieranie alebo špionáž.

### **Motivácie počítačovej kriminality**

Najčastejšími motívmi páchania počítačovej kriminality sú hnev voči nadriadeným, nedostatočný kariérny postup, experimenty a žarty, príležitostné činy, ideologické dôvody a finančný profit. Medzi ďalšie motivácie patria napríklad aj zvedavosť, výzva, recesia, provokácia, politické motivácie, pomsta, deštrukcia, vydieranie, umelý trh v oblasti IT bezpečnosti, ochrana záujmov, konkurenčný boj, strata kontroly ako aj získavanie údajov a veľa ďalších.

### **Druhy kriminality**

Vzhľadom na široké pole nasadenia a používania výpočtovej techniky v rôznych oblastiach je ťažké reálne zmapovať a popísať všetky prejavy počítačovej kriminality.

Všeobecne sa jedná o tieto kategórie:

1. **útok na počítač, program, údaje, komunikačné zariadenie:** fyzické útoky na zariadenie výpočtovej techniky, magnetické médiá, vedenie počítačovej siete alebo elektrického rozvodu a pod., vymazanie alebo pozmenenie dát, formátovanie pamäťových médií nesúcich dáta, pôsobenie počítačových infiltrácií, nelegálna tvorba a rozširovanie kópií programov, získanie kópie hospodárskych dát, databáz zákazníkov, v štátnych orgánoch únik informácií o občanoch a pod. Z hľadiska rozsahu najväčších škôd pravdepodobne najväčší podiel patrí nelegálnej tvorbe a predaju autorsky chráneného programového vybavenia v počítačovom slangu označovaná ako [Warez](#).
2. **neoprávnené užívanie počítača alebo komunikačného zariadenia:** využívanie počítačovej techniky, faxov, prostriedkov počítačových sietí, databáz a programov zamestnancami firiem a organizácií na vlastnú zárobkovú činnosť.
3. **neoprávnený prístup k údajom, získanie utajovaných informácií (počítačová špionáž) alebo iných informácií o osobách, činnosti a pod.:** prenikanie do bankových systémov, systémov národnej obrany, do počítačových sietí dôležitých inštitúcií a pod. Niekedy táto činnosť spôsobuje priame škody veľkého rozsahu, napr. nelegálne bankové operácie, ako aj nepriame škody spôsobené únikom informácií. V súvislosti s týmto trestným činom môže byť aj súbežný trestný čin ako napr. vydieranie, nekalá súťaž, ohrozenie hospodárskeho tajomstva, vyzvedačstvo, ohrozenie štátneho tajomstva.
4. **krádež počítača, programu, údajov, komunikačného zariadenia**
5. **zmena v programoch a údajoch (okrajovo i v technickom zapojení počítača resp. komunikačného zariadenia):** zmena programov a údajov inými programami alebo priamymi zásahmi programátora, úprava v zapojení alebo inom atribúte technického vybavenia počítača.

6. ***zneužívanie počítačových prostriedkov k páchaniu inej trestnej činnosti:*** manipulácia s údajmi ako napr. zostavy v skladoch, tržby, nemocenské poistenie, stavy pracovníkov, stav účtov a pod., patria sem aj krádeže motorových vozidiel, falšovanie technickej dokumentácie, priekupníctvo, daňové podvody, falšovanie a pozmeňovanie cenín, úradných listín a dokladov, dokonca aj peňazí.
7. ***podvody páchané v súvislosti s výpočtovou technikou:*** využitie niečieho omylu vo svoj prospech (hry s vkladom finančnej čiastky a rozosielaním listov “následníkom” so sľubom zaručeného zisku). Tento druh trestnej činnosti možno vykonávať aj bez použitia výpočtovej techniky, ale s jej použitím je táto činnosť efektívnejšia.
8. ***šírenie poplašných správ:*** vytvorenie poplašnej správy upozorňujúcej na fiktívne nebezpečenstvo. Najčastejším motívom páchatel'ov tejto trestnej činnosti je pobaviť sa na nevedomosti ostatných, no môže ísť i o správy spojené s páchaním inej trestnej činnosti. Tieto správy sú v počítačovom slangu označovaná ako [Hoax](#).

## Prejavy internetovej kriminality

**Priemyselná špionáž.** Existencia tohto zločinu siaha do minulosti pred vynálezom výpočtovej techniky. Dnes však prienik do systému konkurencie a zneužitie dát je vykonávaný hlavne prostredníctvom počítača. Prevedenie býva v reálnom čase, infikovanie vírusom alebo prienik do cudzieho počítača, ktorého účelom je získať informácie z hostiteľského systému.

**Šírenie pornografie.** Okrem porušovania autorských práv a hackingu patrí táto oblasť medzi najčastejšie nelegálne aktivity páchané prostredníctvom počítača a internetu. Pred internetom sa táto aktivita rozvíjala a šírila formou časopisov a video nahrávok. Vďaka nadnárodnému pôsobeniu internetu sú stránky s nelegálnym obsahom viditeľné po celom svete. V SR je trestná výroba a distribúcia pornografických materiálov, v ktorých sú znázornené násilné a ľudskú dôstojnosť ponižujúce činnosti, styky s deťmi a zvieratami, prípadne iné patologické sexuálne praktiky. Rovnako je trestné sprístupňovanie pornografických materiálov maloletým.

**Hacking** je útok vo forme prieniku so zámerom skompromitovania počítača. Snaha vykonať veci ku ktorým používateľ nemá oprávnenie, ale hackeri väčšinou získané dáta nezneužívajú. Hacking so svojou bohatou históriou je najvýraznejšou oblasťou počítačovej kriminality. Nebezpečie, pokusu o prienik hackera do domáceho počítača je veľmi malé. Väčšinou sa pokúšajú preniknúť do firemných počítačov a serverov. Známe sú aj prípady tzv. násilného hackingu. Jedná sa o priamy fyzický útok na osobu s administrátorským právom v systéme, ktorú násilím alebo vyhrážkami donútia vyčleniť prístupové heslo, prípadne vykonať požadovanú operáciu pre útočníka. Podľa historických štatistík hackingu, organizáciám hrozí väčšie nebezpečenstvo od ľudí oprávnených pohybovať sa v systéme ako od anonymných hackerov.

**Carding** alebo zneužívanie platobných kariet. Vznik súvisí s rozvojom internetu. Platobná karta sa stala bežným platobným nástrojom. Jej zabezpečenie nie je vždy dostatočné. Krádežou karty, alebo jej čísla sa vykonáva veľa druhov trestnej činnosti. Jeden z prvých bol kreditný nákup (nákup tovaru bez platenia v hotovosti). Páchatelia na zistenie čísla kreditnej karty používajú generátory čísiel. Osobné údaje získavajú od majiteľov účtu rôznymi spôsobmi. Medzi najčastejšie parí sociálne inžinierstvo kde sa páchatel' vydáva napr. za zamestnanca banky, v ktorej bola karta vydaná a predstiera problémy v bankovom systéme. Ďalšími spôsobmi sú, hľadanie vyhodnených výpisov z bankomatov, inštalovanie kamery nad

bankomat, zistenie PIN čísla karty pomocou tenkej klávesnice a následné okradnutie obete, falošné bankomaty, pri platbe cez internet z účtu zákazníka strhnúť viac, ako bola zverejnená cena za tovar.

**Sociálne inžinierstvo** (Social Engineering). Je to praktika na získavanie dôvernej informácie manipuláciou a zavádzaním legitímnych používateľov. Najjednoduchšou, ale stále veľmi efektívnou metódou je presvedčiť používateľa, že potrebuje pomoc administrátora systému na záchranu, alebo obnovenie jeho údajov, a tak od neho dostať heslo k používateľskému kontu či ho presvedčiť, aby odhalil nejakú inú citlivú informáciu.

**Extrémizmus.** Internet je pre aktivity extrémistických skupín ideálnym prostriedkom. Je masovo využívaný ku komunikácii medzi jednotlivými členmi skupín ktorými sú napríklad anarchistí, neonacisti, radikálne ekologické iniciatívy, náboženské sekty či komunisti.

**Informačná vojna.** Patrí medzi extrémne prejavy moderných technológií a ich hrozieb. V súčasnosti sa zvyšujú rôzne aktivity v boji proti hackerom a počítačovej kriminalite. Toto platí pre hlavne pre internet a k nemu pripojené systémy. Táto téma sa stáva aj politickým záujmom. Vo vojnových konfliktoch sa väčšinou počítače využívali len ako podpora vedenia vojny. Informačná vojna ale znamená nový aspekt pretože počítače sa stali zbraňou a zároveň aj terčom útokov.

Je tu však ešte širší pohľad na vzťah medzi bezpečnosťou a informačnými technológiami. Nejde len o ohrozenie napríklad akcieschopnosti armády, ale aj o možné ohrozenie celého demokratického systému Internetom v jeho dnešnej, celkom neregulovanej podobe. Veľkú časť užívateľov Internetu predstavujú mladí niekedy aj neplnoletí ľudia, ktorí sú neskúsení a nevyzretí a tým aj ľahko ovplyviteľní a rôznymi spôsobmi manipulovateľní. Vojna na internete môže byť v podobe propagandistického pôsobenia. V modernej dobe sa konflikt vyhráva nielen zbraňami a dobývaním územia, ale aj pozornosťou verejnosti v elektronických médiách.

## **Prienik do počítačového systému**

Najobávanejším druhom počítačovej kriminality je útok na počítačový systém. Človek zaoberajúci sa touto činnosťou sa v počítačovom slangu nazýva Hacker.

V minulosti sa termín hacker spájal s vyjadrením vysokého stupňa odbornosti a šikovnosti na úrovni počítačového experta. Dnes verejnosť označuje za hackerov všetkých, ktorí neoprávnene prenikajú do cudzích počítačov a sietí.

Typológia hackerov:

- **Hacker** - počítačový expert, dobrý programátor, hľadajúci bezpečnostné diery v systémoch, za účelom zlepšenia ich bezpečnosti. O nájdených chybách a nedostatkoch informuje autorov programov, správcov systému aj verejnosť.
- **Cracker** - má technické schopnosti ako hacker, ktoré ale používa vo svoj prospech, väčšinou ilegálne. Patria sem aj takzvaní softvéroví, filmoví a hudobní piráti, lovci čísiel kreditných kariet a iní. Najčastejším motívom je pre nich uznanie v komunite a peniaze. Jedná sa o plánovanú a premyslenú činnosť.

- **Script-Kiddie** - útočí na bežných užívateľov. Jeho technické znalosti nezvyknú byť veľké. Väčšinou sa jedná o mládež, ktorá skúša použiť dostupné nástroje na tvorbu vírusov, alebo preniknutie do systémov za účelom páchania škôd. Hovoríme im aj internetoví vandali. Motívom je zábava, zaháňanie frustrácie, chválenie sa. Konajú náhodne, pod vplyvom času a nálady.
- **Spamer** - je človek, ktorý prevádzkuje a organizuje rozposielanie nevyžiadaných správ. Činnosť vykonáva pre svojich klientov, ktorí s jej pomocou chcú zbohatnúť.

Najčastejšie metódy na prienik do systému sú tieto:

### Útok hrubou silou

Útok hrubou silou je metóda, ktorá spočíva vo vyskúšaní všetkých možných kombinácií znakov. Útočník zostrojí program, ktorý sa pokúša postupným vyskúšaním všetkých možností uhádnuť Vaše heslo. Rozlúšteniu takéhoto hesla zabránite použitím dostatočne dlhého hesla (pri súčasnom výkone počítačov sa odporúča minimálne 8 znakov). Dôležité je použiť čo najširší možný okruh znakov – malé i veľké písmená, čísla a ďalšie symboly. Toto heslo je potrebné tiež často meniť. Samozrejme to záleží na dôležitosti príslušného hesla. Je nanajvýš nevhodné ukladať hesla na verejne dostupných počítačoch.

### Slovníkový útok

Tento útok spočíva v skúšaní všetkých slov daného jazyka. Takémuto útoku sa dá predísť tak, že použijete heslo, ktoré nie je slovom žiadneho jazyka. Bezpečné heslo si môžete odvodiť napríklad takto: Vezmeme si prvé písmená vety, ktorú si ľahko zapamätáme: A predsa sa točí. Galileo Galilei. Dostaneme Aprsto-GaGa. Toto heslo môžeme ešte vylepšiť napríklad takto Apr100-2\*Ga.

### Odpočúvanie sieťovej komunikácie (Sniffing)

Používa sa pri ladení sieťových aplikácií, hľadani chýb v aplikácii respektíve v sieti. Zneužíva sa na odpočúvanie a odchyťovanie nešifrovanej komunikácie. Vaše heslo sa dá veľmi jednoducho získať odpočúvaním nezabezpečených komunikačných liniek ako sú http:// a ftp://. Preto nikdy nezadávať svoje údaje do stránky, ktorá nie je zabezpečená šifrovanou komunikáciou https:// alebo ftps:// (poprípade inou).

### Využitie neukončeného spojenia

Útočník môže využiť, že sa zabudnete odhlásiť zo systému. Využije otvorené spojenie, ktoré zneužije vo svoj prospech. Niektoré stránky sa proti takýmto útokom chránia automatickým ukončením spojenia pri nečinnosti (preto sa nedá odoslať mail, ktorý píšete dlhšie ako 15 minút).

### Zadné vrátka

Útočník zostrojí program nazývaný Backdoor (zadné vrátka), ktorý mu umožní pripojiť sa do systému bez nutnosti poznať správne používateľské meno a heslo. Tento program rozšíri pomocou počítačového červa alebo trójskeho koňa.



## **Odchytenie hesla**

Útočník zostrojí program nazývaný Keylogger, ktorý zaznamenáva stlačené klávesy a takto získané údaje mu odosiela prostredníctvom Internetu. Tento program rozšíri pomocou počítačového červa alebo trójskeho koňa.

## **DoS útok**

Útok typu DoS (denial of service) spočíva v zahltení servera čím vznikne nedostupnosť serverových a aplikačných služieb. Útok sa prevádza napríklad posielaním veľkých e-mailov, neustálím pingom a inými technikami zahlcovania. Tento útok sa v praxi využíva na likvidáciu konkurencie, alebo ako vydieranie a elektronické výpalné. Proti novej on-line mafii sú úrady aj firmy takmer bezradné. Vydierači, prostredníctvom DoS útoku vydierajú majiteľov firiem a hrozia im úplným ochromením prevádzky ich webovej stránky. Pre niektoré firmy to môže znamenať stratu ziskov, alebo úplne zruinovanie. Dôvodom prečo sa nový druh organizovaného zločinu ťažko objasňuje, je precízna a dôsledná stratégia páchatel'ov.

## **XSS útok**

*Cross Site Scripting* (skriptovanie naprieč stránkou) je špeciálny druh hackingu internetových stránok, ktoré používajú zle ošetrené vstupy. Tieto vstupy sa dajú zneužiť na vloženie akéhokoľvek kódu útočníka.

## **CSRF/XSRF útok**

*Cross-site request forgery* je veľmi podobný predošlému XSS útoku. Pre predanie argumentov stránke obeti využíva metódu GET, čo znamená že parametre sú posielané priamo v URL adrese. Tento typ útoku v podstate nie je chybou ani prehliadača ani užívateľa a je podstatou samotného protokolu.

## **Počítačové bankové krádeže (phishing, pharming, spoofing),**

Medzi špeciálnu kategóriu útokov patria Phishing a Pharming, kde sa niekto iný vydáva za seriózne web stránky a touto cestou sa pokúša získať od používateľa dôležité informácie ako prístup k bankovým kontám a podobne. Odkazy na tieto web stránky sú rozposielané formou spamu.

Bankové krádeže uskutočnené pomocou počítača sú zatiaľ u nás zriedkavé no vo svete sa začínajú čoraz viac vyskytovať. Známe sú nasledujúce tri typy krádeží:

### **Phishing**

Phishing (rybárčenie) využíva metódy sociálneho inžinierstva a rozširuje sa v prvom rade e-mailom. Falšovaná správa, ktorá vás pod určitou zámenkou nabáda ku zmene osobných údajov. Správa vyzýva užívateľa, aby zmenil resp. upravil svoje osobné údaje v bankových inštitúciách, alebo na web stránkach, kde sa držia finančné prostriedky, resp. záväzné objednávky. V takomto emaile je umiestnený odkaz, na ktorom si heslo máte zmeniť. Odkaz však nesmeruje na inú web stránku, ako je originálna web stránka spomenutej inštitúcie. Táto

web stránka vyzerá a správa sa takmer úplne rovnako, užívateľ si to nevšimne. Prihlásením sa na tejto web stránke získa nepovolaná osoba prístupové údaje a možnosť okradnúť užívateľa o finančné prostriedky. Takéto správy sú väčšinou veľmi formálne napísané. Niektoré dokonca vyzerajú tak, že ich odosielateľom je samotná banka. Preto si vždy overte pravosť takejto správy a neotvárajte stránku cez odkaz v pošte.

## **Pharming**

Pharming (farmárčenie) má rovnaký efekt ako phishing, ale pracuje na inej báze. V tomto prípade je napadnutý buď internetový poskytovateľ, alebo užívateľský počítač, kde sú zmenené tzv. DNS záznamy. Touto cestou adresy web stránok ukazujú na iné miesta, ako by mali, takže nastáva rovnaký efekt ako pri phishingu - užívateľ sa dostane na falošnú web stránku, kde prezradí dôležité tajné informácie ako napríklad meno a heslo k bankovým operáciám. Každý mennej adrese napríklad `ib.vub.sk` prislúcha číselná adresa napríklad `215.5.214.144`. Pomerne jednoduchým spôsobom sa dá toto nastavenie zmeniť. Ak zadáte mennú adresu do vášho prehliadača, miesto stránky banky sa zobrazí jej dokonalá napodobenina. Potom miesto mennej adresy do prehliadača zadáte číselnú adresu (napríklad `https:// 215.5.214.144`). Ďalšou možnosťou je overovanie platnosti certifikátu a upozornenie pri prechode zo zabezpečenej stránky na nezabezpečenú. Tieto funkcie sa dajú nastaviť v internetovom prehliadači. Niektoré banky sa proti takémuto spôsobu elektronického podvodu bránia tak, že Vám ihneď po prihlásení do systému pošlú SMS s kódom, ktorý musíte zadať alebo Vás aspoň upozornia, že sa niekto prihlásil k Vášmu účtu.

## **Spoofing**

Do kategórie Spoofing patria všetky metódy, ktoré používajú hackeri na zmenu totožnosti odosielaných správ. DNS spoofing - snaha predstierať, že útočník vlastní doménové meno, podhodenie mena. IP spoofing - snaha predstierať, že útočník posiela údaje z tejto adresy. Odosielateľov špiónážnych programov, vírusov, spamu je veľmi ťažké identifikovať a lokalizovať miesto odkiaľ vykonávajú útoky. Väčšina z nich je inteligentných, preto rozosielanie, vykonávajú prevažne vo verejných internetových kaviarňach, školách, knižniciach. Každý počítač pripojený na internet, má svoju identickú jedinečnú IP adresu. Na základe toho, v spolupráci s providerom (poskytovateľom internetového pripojenia) je možné páchatel'a identifikovať a lokalizovať. Existujú špeciálne programy, ktoré vedú zmeniť nielen IP, ale aj MAC adresu, čo je jedinečná adresa technického zariadenia Najviac nebezpečnou formou spoofingu je metóda nazývaná MITM (man-in-the-middle v preklade „muž v strede“). Táto metóda spočíva v narušení komunikácie medzi klientom a bankou, pri ktorej útočník naruší šifrovací systém verejného a súkromného kľúča, ktorý sa používa pri komunikácii. Použiť metódu MITM však nie je jednoduché, pretože na narušenie komunikácie je potrebné získanie kľúča (niekedy tiež označovaný ako certifikát) banky, ktorý sa často mení. Je preto dôležité nastaviť Váš internetový prehliadač tak, aby overoval, či je certifikát ešte platný.

## **Dialery**

Ďalším zákerným spôsobom ako od používateľa získať finančné prostriedky sú programy, ktoré menia telefonické pripojenie počítača tzv Dialery. Tieto programy používajú za účelom prístupu na platené služby v Internete bez použitia kreditnej karty. Je preto možné, že niektorí používatelia môžu kdekolvek na svete nainštalovať tento typ programu. Dialery však dokážu

zmeniť konfiguráciu systému tak, aby boli používatelia, ktorí sa pripájajú do Internetu cez modem, presmerovaní na čísla s vysokou tarifikáciou. Tento spôsob zarábania je síce neetický, ale legálny, pretože používateľ sám vykoná akciu, ktorá číslo zmení.

### 30. Úloha č.1: Textový súbor – zápis údajov do nového súboru

Vysvetlite postupnosť činností počítača a im odpovedajúce príkazy v jazyku C pri zápise údajov do textového súboru.

#### *Príklad*

Napíšte program, ktorý načíta text po znakoch z textového súboru *vstup.txt* a všetky malé písmená prepíše na veľké. Vytvorte nový súbor *vystup.txt* a zapíšte do neho zmenený text s veľkými písmenami. Načítavanie údajov z textového súboru a zápis do nového súboru môže prebiehať naraz v tom istom cykle.

#### Úloha č.2:

- a) Nájdite na Internete 5 obrázkov o prírode a spracujte z nich krátke video s titulkami. Použite program Movie Maker
  
- b) Vysvetlite význam vybraných parametrov procesora – taktovacia frekvencia, veľkosť registrov, šírka zberníc, počet jadier, veľkosť vyrovnávacej pamäte.

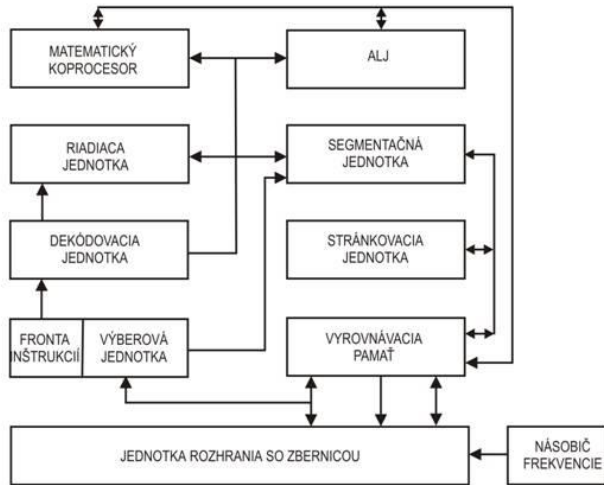
## Mikroprocesor

**Mikroprocesor** je srdcom každého počítača. Hlavne od neho závisí aký bude systém výkonný a rýchly, teda ako rýchlo bude spracovávať zadané údaje.

Mikroprocesor spracováva **inštrukcie** programu, ktorý ho riadi. Niektoré inštrukcie je schopný spracovávať sám, na spracovanie iných zase využíva iné komponenty počítača. Základnou vlastnosťou mikroprocesora je jeho **programovateľnosť** a integrácia všetkých obvodov do jedného puzdra.

Jarom každého mikroprocesoru je logický obvod, ktorý dokáže spracovať sadu jednoduchých

inštrukcií, čo sú vlastne jednoduché príkazy  
Logická štruktúra mikroprocesora



Keď je program skopírovaný do **operačnej pamäte**, mikroprocesor dostane informáciu, na ktorej adrese v pamäti spustená aplikácia začína. Z tejto adresy si procesor načíta **inštrukciu**, ktorá sa zapíše do **vyrovnávacej pamäte** pre ďalšie použitie a **odovzdá sa ďalej procesoru**.

Inštrukčný **dekóder** sa stará o to, aby procesor vedel čo má robiť, akú inštrukciu má vykonávať a aké sú jej parametre – **dekóduje inštrukciu**. Po dekódovaní inštrukcie procesor spozná, aká je dlhá a kde začína nasledujúca inštrukcia.

Podľa toho, o akú inštrukciu sa jedná, zaistí CPU (procesor) jej vykonanie. **Riadiaca jednotka** určí, kam rozkódovaná inštrukcia poputuje. Či sa o ňu bude starať **ALJ** (spracováva aritmetické a logické inštrukcie), **matematický koprocesor** (spracováva inštrukcie pracujúce s desatinnými číslami), alebo iný blok procesora.

Zmeny sa môžu premietnuť do pamäte, alebo len do **vnútorných registrov**, prípadne program ovplyvňuje chovanie **periférií** pomocou **portov**.

**Registre** - každý procesor má niekoľko pamäťových miest - registrov, do ktorých sa ukladajú práve spracovávané údaje. Počet a použitie týchto registrov závisí od typu procesora.

**Periférie** – zariadenia pripojené k počítaču a riadené počítačom (tlačiareň, monitor, scanner ...)

Samotná inštrukcia môže žiadať ďalšie údaje z pamäte, na ktoré sa potom čaká. Aj tieto sa zapisujú do **vyrovnávacej pamäte**, pre prípad, že by boli neskôr potrebné. Aby celý tento systém fungoval synchronizovane, je súčasťou procesoru **násobič frekvencie** ktorý násobí základnú frekvenciu hodinového impulzu (**FSB**) čím vytvára **pracovnú frekvenciu procesora**. Táto určuje momenty, kedy sa inštrukcia odovzdáva medzi jednotlivými časťami procesoru. Počíta tiež tiky pre jednotlivé bloky procesoru v prípade, že im spracovanie inštrukcie trvá dlhšie, než hodinový takt.

**Cache pamäť**

Na kľúčových miestach procesora sú umiestnené vyrovnávacie pamäte. Cache pamäť je vlastne veľmi rýchly pamäťový zásobník určený na dočasné ukladanie dát, ktoré procesor potrebuje.

**Cache sa snaží načítať a uchovávať tie dáta, ktoré procesor bude v najbližšej dobe s veľkou pravdepodobnosťou potrebovať.** Vďaka tomu v momente potreby procesor načíta tieto dáta z pamäte rovnakou rýchlosťou ako pracujú jeho ostatné časti - **jadro procesora**. Údaje z operačnej pamäte a iných častí počítača do procesora sa prenášajú cez vodiče, ktoré tvoria **systémovú zbernicu**.

## Charakteristiky mikroprocesora

Každý mikroprocesor má svoje špecifické vlastnosti, ktoré samozrejme ovplyvňujú aj jeho cenu:

- **Šírka slova (vnútorná šírka dát)** je určená počtom bitov, ktoré je procesor schopný spracovať v rámci jednej inštrukcie. Môže nadobúdať hodnotu 8, 16, 32 a 64 bitov. Táto hodnota vyjadruje šírku vnútornej zbernice procesora.
- Nezávisle od šírky slova je nutné údaje do procesora dodávať a odoberať. **Šírka dátovej zbernice** je parametrom, ktorý vyjadruje rozmer vonkajšej zbernice určenej na komunikáciu procesora s okolím. Najdôležitejším dôsledkom vyplývajúcim zo šírky dátovej zbernice je to, že táto šírka definuje aj veľkosť jednej bunky pamäte.
- **Šírka adresnej zbernice** vlastne definuje veľkosť operačnej pamäte, ktorú je procesor schopný využiť.

486/Pentium	-32 bitov	- 4 GB
PII, PIII, P4	- 36 bitov	- 64 GB
ITANIUM	- 44 bitov	- 16 TB
Intel Core 2 , Athlon X2	- 64 bitov	- 16 EB

- **Vnútorné registre** určujú koľko informácií je procesor schopný spracovať počas jedného cyklu a akým spôsobom sú dáta v procesore presúvané. Všeobecne sa dá povedať, že väčšina súčasných procesorov obsahuje 64 bitové vnútorné registre.
- **Taktovacia frekvencia** reprezentuje výkon a „rýchlosť“ procesora. Udáva sa v hertzoch, v súčasnosti v **GHz**. Tento parameter udáva koľkokrát za sekundu je procesor schopný zmeniť svoj stav. Najmenšou jednotkou času z hľadiska procesora je **1 cyklus**. Každá akcia vyžaduje minimálne jeden cyklus, platí však, že väčšina bežných operácií vyžaduje viac cyklov. Napríklad PIII potrebuje pre prvý prenos dát do pamäte aspoň 3 cykly, pre následné 3 až 6 prenosov potrebuje už len 1 cyklus na prenos.  
Pentium a ostatné procesory 5. Generácie počas jedného cyklu vykonali 1 až 2 operácie.  
PII /PIII/P4/Celeron a ATHLON/DURON vykonajú 3 a viac inštrukcií v jednom cykle. Potom napr. PIII pracujúci na 1000 MHz zodpovedá svojím výkonom teoretickému procesoru Pentium bežiacemu na 1500 MHz.

**Rýchlosť procesora je násobkom rýchlosti základnej dosky danej frekvenciou FSB.** Platí, že rýchlosť procesorovej zbernice sa nastavuje tak, aby zodpovedala rýchlosti pamäťových modulov.

Ďalším problémom pri vzájomnom porovnávaní procesorov je ich architektúra. Dá sa povedať, že procesor s určitou vnútornou architektúrou je schopný vykonávať určité úkony rýchlejšie ako obdobný procesor s inou vnútornou architektúrou.

Preto je okrem frekvencie základnou jednotkou merania výkonu procesora **FLOPS (Floating Point Operations Per Second)** udávajúci počet operácií v pohyblivej rádovej čiarky, ktoré je procesor schopný vykonať za jednu sekundu.

Ďalšie vlastnosti ovplyvňujúce výkonnosť mikroprocesora (jeho architektúra a podporované technológie):

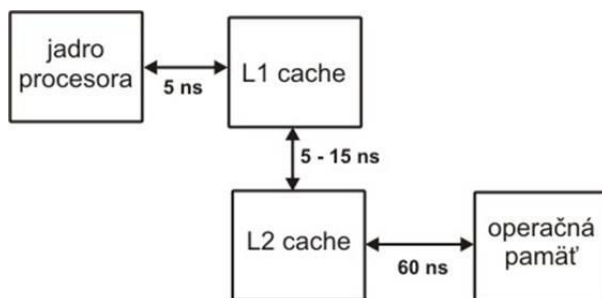
### Vyrovňavacia pamäť

V súčasnosti sa v počítačoch používajú tri úrovne vyrovnávacej pamäti: L1 cache, L2 cache a L3 cache.

#### L1 cache:

Sú ňou vybavené procesory od 486DX (8 kB). V súčasnosti je rozdelená na cache pre dáta a cache pre inštrukcie – procesory Intel štandardne 2 x 16 kB.

Je integrovaná do puzdra procesora a pracuje na plnej rýchlosti (frekvencii) jadra. Preto je procesor schopný k nej pristupovať bez akýchkoľvek čakacích stavov.

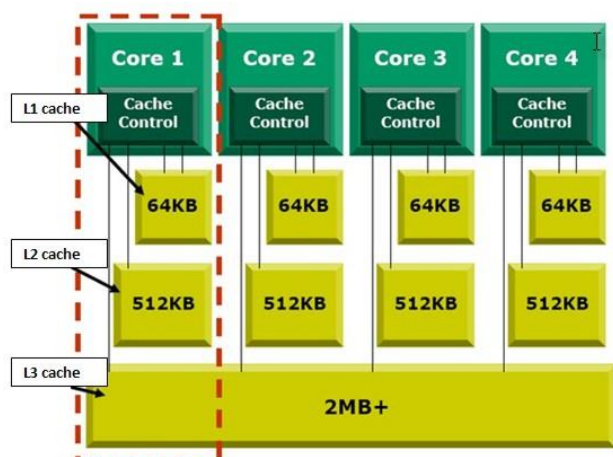


Procesor hľadá údaje v L1 cache. Ak tam nie sú načíta ich z L2 cache. Ak nie sú údaje ani v L2 cache čaká procesor na naplnenie cache z operačnej pamäte.

#### L2 cache:

Pridáva sa, aby nedochádzalo k výraznému spomaleniu systému pri každom chybnom cachovaní. U prvých PC bola L2 cache umiestnená na základnej doske. Pracovala na frekvencii systémovej zbernice a bola rýchlejšia ako operačná pamäť. V súčasnosti je L2 cache integrovaná na čípe a pracuje na plnej frekvencii jadra.

Súčasný mikroprocesory majú v sebe integrovanú aj L3 cache pamäť.



### Viacjadrová architektúra

Súčasný procesory pozostávajú z viacerých spúšťacích jadier, ktoré sú začlenené do jedného procesora. Najlepšiu predstavu asi získate, ak si predstavíte viac procesorov v jednom. Dvojjadrové spracovávanie inštrukcií znamená, že zatiaľ čo jedno jadro je zamestnané bežiacou aplikáciou alebo vláknom, napríklad zisťovaním vírusov, druhé spúšťacie jadro je stále dostupné, aby mohlo spracovávať ďalšie úlohy vykonávané koncovým používateľom, ako sú prezeranie obsahu na Internete alebo práca s tabuľkovým editorom.

V súčasnosti sa vyrábajú už aj trojjadrové, štvorjadrové, šesťjadrové a osemjadrové mikroprocesory.

- c) Čo je ergonómia? Popíšte orgány, ktoré pri práci s počítačom trpia najviac a navrhните vhodnú starostlivosť o ne.

## Ergonómia

- vedecká disciplína, ktorá sa zaoberá výkonnosťou pracujúceho človeka a prispôbovaním pracovných prostriedkov a pracovného prostredia vlastnostiam a potrebám človeka.

## Počítač a naše zdravie

### Vplyv pracovného prostredia na človeka pri práci s počítačom

Či si to už človek uvedomuje viac, alebo menej, na jeho psychiku, fyzický a zdravotný stav pôsobí v pracovnom procese celý komplex vonkajších ale aj vnútorných vplyvov:

- **Svetlo.** Jeho kvalita, umiestnenie zdrojov, odrazy a odlesky majú priamy vplyv na záťaž nášho zraku. Svetlo ovplyvňuje kvalitu a množstvo získavaných zrkavých informácií, vplyva na efektivitu práce a pôsobí aj na náš psychický stav.
- **Farby.** Patria k zrkavým vnemom a ovplyvňujú psychiku pri práci, niektoré priaznivo a ukludňujúco, iné záporne. Týka sa to aj nielen farebného riešenia priestorov kancelárie, ale aj počítača a jeho komponentov a používaného software.
- **Hluk.** Neprijemné dôsledky hluku pri práci s PC zapríčiňujú stratu koncentrácie a zvýšenú chybovosť, hoci samotné počítače nie sú zdrojom škodlivých prejavov hluku. Pri práci s počítačom sa zdrojom rušivého hluku stávajú najčastejšie zvuky

doliehajúce z ulice do miestnosti. Zdroj hluku však môže byť aj priamo vo vašej kancelárii. Mikroklima. Teplo, chlad, vlhko, prúdenie vzduchu.

- **Nízke alebo príliš vysoké teploty pracovného prostredia** citelne ovplyvňujú chybovosť a výkonnosť obsluhy zariadení vplyvajú na chorobnosť ľudí.
- **Žiarenie.** Najčastejším negatívnym sprievodným javom katódových obrazoviek počítačových monitorov je elektromagnetické žiarenie. Jeho následky začínajú únavou v niektorých prípadoch podráždením pokožky tváre a očí. Stres. Pôsobí na psychiku a ovplyvňuje výkonnosť a chybovosť. Pri dlhodobom pôsobení môže zapríčiniť psychické ale aj telesné ochorenia.
- **Tvar a umiestnenie.** Tvarové riešenie ovládačov a priestorové uporiadanie jednotlivých častí pracovného miesta má zásadný vplyv na fyzickú a psychickú pohody pri práci s počítačom, vplyva na výkonnosť a kvalitu práce.

## Ergonomické pomôcky

Pri častejšej práci s počítačom je potrebné myslieť na naše zdravie a štandardnú počítačovú zostavu rozšíriť o pomôcky, ktoré pomáhajú predchádzať bolestiam chrbtice, krku, predlaktia, ramien, a ďalším dôsledkom dlhodobého sedenia pri počítači.

### Myš



### Opierky zápästia

Požiadavky EÚ: Priestor pred klávesnicou má byť dostatočne veľký na pohodlné podoprenie zápästí a rúk používateľa. Dlhodobá práca na klávesnici môže spôsobiť chorobu nazvanú syndróm karpálneho kanála. Šľachy na zápästí, ktoré opuchnú, stláčajú cievy a nerv, následkom čoho nastáva chvenie prstov, precitlivosť až bolestivosť zápästia. Táto choroba, keďže sa vyskytuje vo viac ako 60 profesiách, ktoré vyžadujú opakovaný monotónny pohyb, je uznaná Svetovou zdravotníckou organizáciou ako choroba z povolania. Experti odporúčajú pri práci neohýbať ruky nahor, nadol a do strán, v ideálnom prípade má byť ruka, zápästie a predlaktie v jednej línii.

Standard ISO: Dizajn opierky zápästia má byť taký, aby minimalizoval statický postoj a nebránil v pohybe. Povrch podložky by mal byť v rovnakej výške ako klávesnica, šírka by mala byť aspoň 50 -120 mm, okraje by mali byť upravené, aby nespôsobili zranenie rúk alebo zápästí. Dĺžka by mala korešpondovať s dĺžkou klávesnice, respektíve pracovnej plochy a počas práce má byť stabilná.



## **Držiak dokumentov**

Požiadavky EÚ: má byť stabilný a nastaviteľný, aby minimalizoval nepohodlný pohyb hlavy a očí

Štandard ISO: pracoviská s počítačom, kde sa často prepisujú údaje z predlohy, by mali byť vybavené držiakom dokumentov. Na prispôsobenie sa požiadavkám na čitateľnosť dokumentov a individuálnym požiadavkám používateľov by mal byť držiak dokumentov nastaviteľný výškovo aj sklonom a vo výške obrazovky. Plocha držiaka by mala byť prispôbena veľkosti predlohy, najlepšie ak je o 10 mm menšia z každej strany pre ľahší prístup k dokumentom. Povrch držiaka by mal byť matný, aby nešlo k zníženiu čitateľnosti dokumentov. Mal by byť stabilný a dostatočne robustný, pre prípad potreby upevnenia viac dokumentov.

Význam držiaka je v tom, že pri prepisovaní z predlohy, nemusí hlava a oči používateľa stále vykonávať pohyb dokument - monitor, zaoštrovať oči na rôzne vzdialené predmety. Predloha vo výške a vzdialenosti monitora rieši tento problém.

## **Podložka pod nohy**

Požiadavky EÚ: Pracovná stolička by mala byť výškovo nastaviteľná s možnosťou nastaviť sklon a výšku operadla podľa požiadavok jednotlivca. Každý kto o to požiada by mal byť vybavený podložkou pod nohy

Štandard ISO: Povrch podložky by nemal byť klzký a mal by byť dostatočný na pohyb nôh. Sklon by mal byť nastaviteľný.

## **Ergonomické klávesnice**

Klávesnice sa často podceňujú - celkom neprávom. Bud' totiž prácu na počítači významne uľahčujú, alebo ju neúmerne sťažujú. Veľkosť dlane a prstov, stupeň zbehlosti v písaní na stroji alebo zmyslové vnímanie a záľuba rozhodujú u kupujúceho, či mu je klávesnica sympatická alebo nie je.

Klávesnica je pracovný nástroj a musí spĺňať určité požiadavky, ktoré uvádza a upravuje norma ISO 9241-4.

Už niekoľko rokov sa veľa hovorí o ergonomických klávesniciach. Sú to klávesnice s rozdelenými a často i tvarovanými poľami kláves, ktoré vychádzajú v ústrety prirodzenému držaniu ruky viac než štandardné ploché klávesnice. Tie nútia písuceho, aby držal obidve ruky rovnobežne, čo vyžaduje u obrovskej väčšiny ľudí neprirodzené držanie paží, pretože rozmery klávesnic sú omnoho menšie než je šírka ramien. Neprestajné zaťažovanie šliach v zápästí môže tiež u profesionálnych pisárov vyvolať tzv. syndróm **RSI** (Repetitive Strain Injury), nemoc, ktorá môže mať za následok nielen opuchy a necitlivosť, ale tiež zápal púzdiar šliach a mazových vačkov. Okrem prestávok a častých zmien polohy pomáha pri namáhaní šliach a svalov tiež udržiavať predlaktia a dlaňové časti v priamke a čo najuvoľnenejšie držanie. Práve toto sľubujú klávesnice. Ergonomické klávesnice sú v priemere dvakrát drahšie než štandardné klávesnice. Je to dôsledok náročnej konštrukcie, ale tiež skutočnosti, že ide spravidla o špičkové modely výrobcov. Základnou veličinou pri posudzovaní klávesnice je priebeh sily v závislosti na zdvihu. Podľa cieľovej skupiny sa konštruujú raz krátke, ľahko idúce tlmené zdvihy kláves a inokedy zas dlhé, výrazné a netlmené. Prvá charakteristika vychádza v ústrety pisárom desiatich prsto, pretože podporujú vysoké rýchlosti písania, zreteľné údery vyhovujú skôr neistým pisárom, pretože znižujú pravdepodobnosť chýb. Väčšina výrobcov sa snaží výjsť v ústrety obom cieľovým skupinám.

## **Monitor**

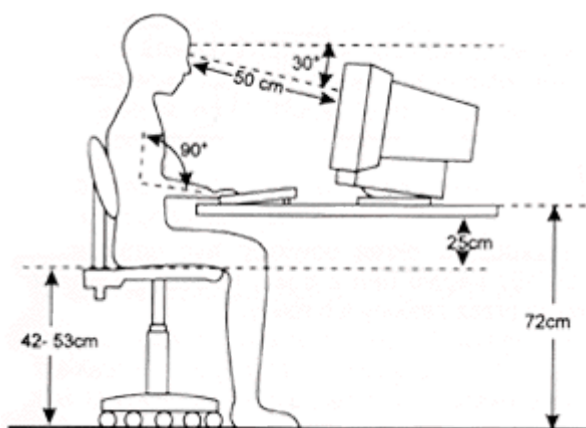
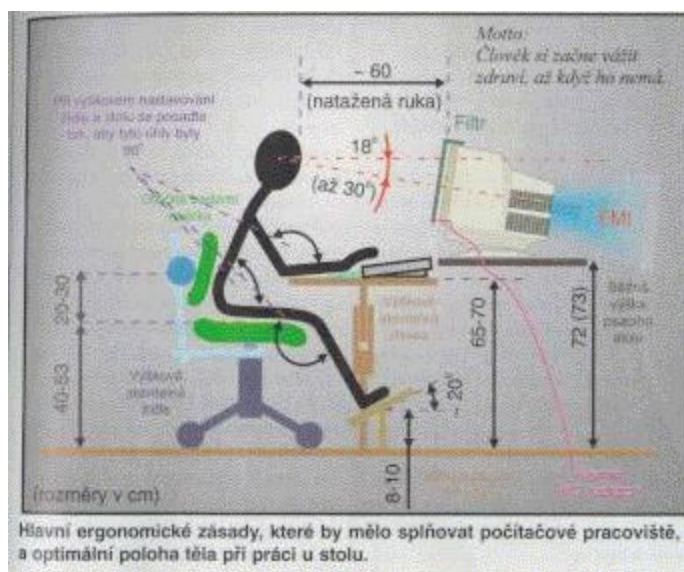
TCO Na monitoroch je často nalepená nálepka s názvom TCO95 alebo TCO99. Čo toto logo vlastne znamená? Je to medzinárodný certifikát organizácie TCO (Švédske odborové hnutie, <http://www.tco-info.com>) v spolupráci so Švédskou organizáciou pre ochranu prírody a NUTEK (Národná švédska komisia pre priemyslový technický rozvoj), ktorý posudzuje vzťah daného zariadenia (monitora) k životnému prostrediu a zdraviu používateľa. Postupne vznikli normy MPR I (1990), MPR II (1991), TCO 92 (1992), TCO 95, TCO 99 (1998). Platí, že čím lepšiu normu monitor spĺňa tým lepšie. V súčasnosti je minimálna požiadavka, aby monitor spĺňal normu TCO 95.

Práca pri počítači najviac zaťažuje oči, z čoho potom prichádzajú bolesti hlavy a celková únava. Okolité svetlo, ktoré dopadá na monitor spôsobuje odrazy, ktoré veľmi znižujú kontrast obrazu na monitore (rozmazávajú ho). Používateľ je nútený prižmurovať oči, zaostrávať ich. Po pár hodinách sa dostavia bolesti hlavy a únava.

- Monitor musí byť umiestnený vo výške očí pracovníka opretého o stoličku.
- Monitor musí mať ľahko otočný podstavec s možnosťou horizontálneho a vertikálneho natáčania.
- Na monitore treba zvoliť čo najvyššiu obnovovaciu frekvenciu (min. 85 Hz) aj keď na úkor rozlíšenia.

## **Pravidlá správneho usporiadania počítačového pracoviska**

- vzdialenosť očí do monitora by mala byť cca 60 cm, tj. na vzdialenosť natiahnutej ruky,
- stred monitora má byť asi 18° pod rovinou očí; ak je vyššie, môže nás záhadne bolieť krk,
- klávesnica by mala byť v takej výške, aby paže v predlaktí zvierali pravý uhol, pod zápästie patrí podložka, tak, aby ruka nesmerovala príliš nahor,
- pred klávesnicou a po jej stranách je potrebné mať voľný priestor; po jednej strane klávesnice musí byť miesto pre myš (len tak na okraj: ak ste praváci, považujte o umiestnení myši na ľavej strane; naučíte ovládať ju ľavou rukou, pravá bude voľná pre inú činnosť, napr. pre písanie),
- monitor nepostavte pred okno, prudký prechod svetla dráždi oči; najlepšie je umiestniť monitor pred stenu, na ktorej je obraz s príjemným motívom,
- monitor umiestnite tak, aby na obrazovke nevznikali odrazy - tie zvyšujú únavu očí,
- vedľa monitora nepostavte kovovú lampu - dochádza tak k sekundárnemu vyžarovaniu elektromagnetického poľa; stojan je vhodné umiestniť minimálne vo vzdialenosti 30 od boku monitora,
- monitory nemajú byť umiestnené v rade za sebou, rozhodne nedopustíte, aby v kuželi EMI (elektromagnetického poľa), ktorý monitor zo zadnej časti vyžaruje, sedel ďalší spolupracovník; z rovnakého dôvodu nie je dobré opierať sa o zadnú časť monitora,
- monitory, prinajmenšom tie, ktoré nespĺňajú hygienické normy (MPR II, či ešte prísnejšie TCO92, TCO95 resp. TCO99), by mali byť doplnené sklenenými filtermi od renomovaného výrobcu; ale aj u monitorov spĺňajúcich normu oceníme prinajmenšom odstránenie nepríjemných odrazov, ktoré dobrý polarizačný filter zabezpečí.



## Čo škodí počítaču alebo načo dať pozor

- prach - pre diskety je to brúsny materiál, na doskách vo vnútri počítača zamedzuje odvádzaniu tepla a na monitore predstavuje veľmi nevhodný filter,
- dym - ide o obdobu prachu v jemnejšom prevedení,
- teplotné šoky - striedanie tepla a chladu (najmä v zime),
- vlhko - spôsobuje koróziu;
- otrasy - škodia predovšetkým rotujúcim častiam, ktorých je v počítači niekoľko: pevné disky, diskety v disketovej mechanike, kompaktné disky v mechanike CD-ROM a vetrák; otrasy môžu navyše spôsobiť aj uvoľnenie prepájacích konektorov vo vnútri počítača,
- elektrické šoky - správne poradie zapínania je: tlačiareň, monitor a nakoniec centrálna jednotka (tá je najcitlivejšia a mala by byť najviac chránená); vypíname v opačnom poradí; nikdy nevypíname akékoľvek elektronické zariadenie obyčajným vytiahnutím zo zásuvky a ak chceme počítač ochrániť aj pred šokmi prichádzajúcimi z elektrickej siete (aj tie môžu spôsobiť vážne poruchy), zaobstaráme si aspoň tzv. napájací filter (vyzerá ako predlžovací kábel alebo ako priradená zásuvka, ale zároveň chráni pred príliš vysokým napätím, ktoré sa občas, hoci len na veľmi krátku dobu, v bežnej zásuvke môže vyskytnúť),
- teplo - počítač pracuje spoľahlivo pri izbovej teplote; chladnejšie prostredie mu prospieva, vysoká teplota (nad 50 ° C vo vnútri skrine) spôsobuje chybovosť alebo úplné zastavenie činnosti počítača, ak nie dokonca jeho poškodenie,
- statická elektina - tá môže zničiť v počítači takmer všetko a jej zdrojom býva práve užívateľ; zbavíme sa jej napr. tak, že sa dotkneme radiátora alebo si umyjeme ruky - prinajmenšom vždy pred otvorením počítača; nesadáme si na umelú hmotu a nenosíme oblečenie z umelých materiálov (radšej ani topánky na gumovej podrážke), aby sme sa trením látky o podložku nenabíli.

### Niektoré situácie, ktorých by sme sa mali vyvarovať

- odporúča sa jedenkrát do roka počítač otvoriť a odstrániť usadený prach (ak však chceme túto operáciu uskutočniť sami, obmedzíme sa iba na opatrné vysatie prachu; lepšie je objednať si firmu špecializovanú na čistenie počítačov); pred

otvorením počítača samozrejme odpojíme počítač od siete a zbavíme sa statického náboja, prach vysávame a jemne prisúvame k hadici vysávača štetcom; na vnútrajšok počítača vlhké prostriedky nepoužívame vôbec,

- otvory na monitore nesmú byť zakryté - nikdy na ne neukladáme papiere ani čokoľvek iné,
- skrinku počítača neprirážame k stene, za počítačom aj za monitorom necháme väčší priestor pre ventiláciu; prirazenie skrinky ku stene tiež nesvedčia konektorom (ako na monitore, tak na skrinke počítača), u ktorých môže dôjsť k poškodeniu pripojených drôtikov,
- voľné otvory po doskách a konektoroch na zadnej stene počítača by mali byť zaslepené, aby nimi nemohol vniknúť prach,
- medzi dvoma zapnutiami počítača počkáme 15 až 20 sekúnd, aby mal počítač čas na ustálenie všetkých prechodových stavov, stabilizáciu napätových pomerov elektronických súčastí a skludnenie zotrvačných hmôt pohyblivých súčastí; inak sa môžeme dočkať aj toho, že náš počítač dopočíta,
- počítač, monitor aj tlačiareň by mali byť napájané z jedného napájacieho okruhu; je preto istejšie použiť predlžovací kábel, tzv. PES (rozdvojky radšej nie - niektoré sú nevhodne prepojené!),
- pri búrke počítača radšej vypneme a šnúru vytiahneme zo zásuvky (nie všade je vedenie chránené pred bleskom),
- po príchode zvonku do vyhriatej miestnosti sa nám okrem okuliarí orosí aj notebook, musíme počkať, až sa vytemperuje približne na teplotu okolia; po bezprostrednom zapnutí by sme tiež mohli ísť hneď do servisu! (to sa týka všetkých elektrických zariadení, netrpezlivosť sa nevypláca),
- tam, kde hrozí riziko výpadku prúdu, je vhodné zaobstarať si záložný zdroj (UPS), to je obzvlášť dôležité pri databázových systémoch, kde výpadok v nevhodnom okamihu môže dáta zničiť; inak väčšinou postačí prácu priebežne ukladať na disk, prideme tak najviac o dáta od poslednej úpravy,
- keď klávesnicu nepoužívame, zakryjeme ju dodaným plastickým krytom alebo poťahom (dá sa dokúpiť),
- občas klávesnicu "vyklepeme", nakloníme ju klávesami dole a mierne do nej niekoľkokrát ťukneme - ak nemáme na pracovisku príliš čisto, budeme sa možno čudovať, koľko smetia vypadne,

---

---

## **TOTO SA NAUČIŤ ! PLATÍ VŠEOBECNE PRE PROGRAMOVANIE A VYSVETLENIE ČO PROGRAMUJEM**

### 1. Algoritmy a programovacie jazyky

Vlastnosti, zápis a štruktúra algoritmov.

Program a postup tvorby programu.

Programovacie jazyky a ich rozdelenie z rôznych hľadísk.

---

**Algoritmus** – postup alebo návod ako riešiť zadanú úlohu. Ide o presne stanovený postup ktorý niekomu adresujeme. V informatickej terminológii je adresátom procesor (vykonávateľ). Algoritmus je postup, ktorého realizáciou získame zo zadaných vstupných údajov po konečnom počte činností v konečnom čase správne výsledky.

Algoritmus musí spĺňať **základné vlastnosti**:

- je konečný (rezultatívny) = vždy vedie k určitým výsledkom a prebieha v konečnom počte krokov
- hromadnosť (všeobecnosť) = rieši úlohu nezávisle na použitých (ale iba prípustných) vstupných údajoch. Algoritmus nepočíta  $1 + 2 = 3$ , ale  $a + b = c$ .
- jednoznačnosť (determinovanosť) = všetky operácie a ich návaznosti sú jednoznačne určené (definované), nič nesmie byť náhodné, neurčené..
- opakovateľnosť = ak sú zadané rovnaké vstupné údaje, dostaneme vždy rovnaký výsledok, tak pri prvom behu algoritmu ako aj pri stopäťdesiatom

P praxi sa veľmi sa doporučuje dodržiavať ešte dve ďalšie vlastnosti:

- zrozumiteľnosť a
- prehľadnosť algoritmu
- efektívnosť.

**Zápis algoritmov:**

- ♦ slovné – v prirodzenom jazyku (algoritmickom jazyku)
- ♦ matematicky (vzťahom medzi veličinami, sústavou rovníc, maticami), rozhodovacie tabuľky
- ♦ graficky (vývojový diagram, štruktúrogram)

**Štruktúra algoritmov:**

1. sekvencia – postupnosť príkazov
2. vetvenie – členenie algoritmu na základe podmienky
3. cyklus – opakovanie časti algoritmu na základe podmienky

**Programovanie** môžeme teda definovať ako zápis algoritmov v jazyku, ktorému počítač rozumie – tzv. programovacím jazyku.

**Programovací jazyk** je sústava slov a pravidiel na zápis algoritmu tak, aby mohol byť spracovaný počítačom.

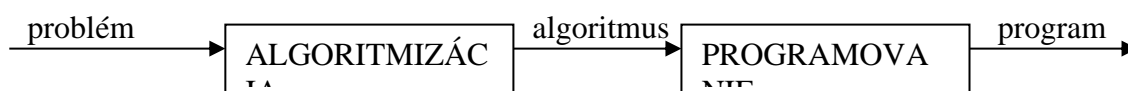
**Program** je postupnosť príkazov (inštrukcií) programovacieho jazyka, ktorou popisujeme, ako má počítač riešiť určitú úlohu.) Program je algoritmus vyjadrený programovacím jazykom.

**Etapy tvorby programov:**

1. **Zadanie úlohy a presné vymedzenie problému.** Rozbor problému ( **čo** ideme riešiť, presne sformulujeme zadanie problému a samozrejme aj požiadavky kladené na program). Výsledkom rozboru je popis vstupných a výstupných informácií a vzťahov medzi nimi.
2. **Navrhujeme postup riešenia** – vytvoríme algoritmus – všeobecný zápis návodu riešenia ( **ako** riešiť, metódy a spôsoby riešenia). Výsledkom návrhu riešenia je zápis jednotlivých

krokov (príkazov) algoritmu a to buď v algoritmickej jazyku alebo graficky pomocou vývojového diagramu alebo štruktúrogramu. Ak sa vyskytnú viaceré možnosti riešenia, snažíme sa zistiť, ktorý je najefektívnejší. Pokúšame sa odhadnúť prácnosť naprogramovania každého z nich a po zohľadnení oboch týchto kritérií si jeden spôsob vyberieme.

3. **Realizácia** (prepis algoritmu do programovacieho jazyka – tie majú rôznu syntax – súbor príkazov a pravidiel, kýmami sa v ňom zapisujú programy)  
**Ladenie a testovanie** – ladenie programu (oprava chýb syntaktických a logických, novšie verzie, prispôsobenie softvéru požiadavkám používateľa)
4. **Dokumentácia a údržba** Ak je program dobrý a chceme ho ponúknuť aj iným ľuďom, je dobré, ak sa k nemu vytvorí minimálne pomocník. Vhodná je aj dokumentácia, ktorú môžu využiť iní programátori, kde opíšete, ako ste pristupovali k riešeniu jednotlivých problémov atď. Ak sa program uchyťí, postaráme sa o jeho údržbu (modifikácie, vylepšenia) a podporu (napríklad možnosť napísať vám mail a opýtať sa na niečo).



## Programovacie jazyky

Programovací jazyk slúži k zápisu programu. V skutočnosti musí byť program preložený do jazyka, ktorému rozumie počítač, konkrétne procesor, t.j. do **strojového kódu**.

(strojový kód	assembler	
3E 02	LD A, 02	vlož
C6 04	ADD A, 04	pripočítaj
Kód inštrukcie, udaj	skratka inštrukcie)	

Prevod z programovacieho jazyka do strojového kódu nazývame **preklad (kompilácia)**.

**Prekladač** (napr. DEV C++) je program, ktorý preloží nami vytvorený program do strojového kódu.

Skladba programovacieho jazyka sa nazýva **syntax** a určuje pravidlá, ako sa majú jednotlivé príkazy a vôbec časti programu zapisovať.

### Rozdelenie programovacích jazykov

Jazyky sa delia na dve základné skupiny:

1. strojovo orientované jazyky
2. vyššie programovacie jazyky

**Strojovo orientované jazyky** sa zvyknú nazývať aj assemblermi. Tento pojem však nie je správny, lebo označuje prekladač z jazyka symbolických adries do strojového kódu. Dnes sa takmer nepoužívajú, aspoň čo sa týka napísania celého programu v takomto jazyku (výnimkou by mohli byť niektoré práčky, mikrovlnky...). Strojovo orientované jazyky sú závislé na použítom type procesoru a ich portabilita (prenositelnosť) na inú platformu je tak veľmi slabá. **Vyššie programovacie jazyky** sa dajú rozdeliť na viacero menších skupín. Ja ich popíšem iba z hľadiska dvoch kritérií:

1. podľa programátorovho prístupu k riešeniu problému a
2. podľa spúšťania programu.

Prvé kritérium nám dovolí rozdeliť jazyky na **imperatívne** a **deklaratívne**. V imperatívnom

programovacom jazyku programátor priamo popisuje celý postup riešenia problému. Teda on sám vytvára algoritmus. Príklady: Algol, Pascal, C, Basic... Pri použití deklaratívneho jazyka programátor iba popíše problém a nestará sa o presný postup, ako sa dopracujeme k výsledkom. Takéto sú jazyky používané v oblasti umelej inteligencie - LISP, Prolog, a ak sa nemýlim, tak aj databázové jazyky - SQL.

Podľa druhého kritéria zase poznáme jazyky **prekladané** a **interpretované**. Prekladané sú také, ktoré po napísaní treba preložiť do strojového kódu, aby im rozumel aj počítač (C, Pascal, C++). Naopak, interpretované sa prekladať nemusia. Tie sa spúšťajú pomocou špeciálnych programov – interpretov – špeciálne programy, ktoré analyzujú jednotlivé príkazy a bezprostredne vykonávajú príslušnú akciu. Akcia sa realizuje využitím dopredu zostaveného postupu v strojovom kóde. Z hľadiska užívateľa sa chová tak, ako by pracoval priamo. Je ale pomalší. Takými jazykmi sú napríklad Java a v podstate všetky "sieťové" jazyky: HTML, PHP, JavaScript, SQL, Logo.

### **Opakovanie:**

#### **Slovník dôležitých pojmov:**

- premenná
- identifikátor
- dátový typ
- deklarácia
- operátory a štandardné funkcie
- syntax

**Premenná** je miesto v pamäti počítača, v ktorom sa nachádza určitá hodnota. Premenná je označená identifikátorom; (pomenované pamäťové miesto). Priradíme jej hodnotu priradovacím príkazom, alebo ju napr. načítame – zadáme z klávesnice. Napr. urob "a 10, urob "a čítajslovo, a := 10, read (a)

**Identifikátor** je názov objektu programu (premennej, konštanty, podprogramu...) napr. X, Y, N alebo objemgule, pocetOpakovani. Je dobré voliť tzv. mnemotechnické identifikátory, tj. také, ktoré približujú premennú, ktorá je v nich uložená. Takže premennú, do ktorej uložíme výsledok výpočtu plochy obdĺžnika, môžeme nazvať napr. P1, ale lepšie PlochaObdlnika. (Vo väčšine programovacích jazykov nesmie identifikátor obsahovať medzery ani písmena, ktoré nie sú v anglickej abecede.)

**Dátový typ.** Premenné sú vždy určitého typu a podľa ich typu s nimi môžeme pracovať. Takže premenná môže byť napr. typu:

- ZNAK, (potom ju ale nemôžeme použiť vo výpočte, dostaneme chybové hlásenie)
- CELÉ ČÍSLO ( potom ale do nej nemôžeme vložiť výsledok delenia, ten často nie je celé číslo)
- REÁLNE ČÍSLO
- LOGICKÁ PREMENNÁ (platí alebo neplatí, áno-nie)

Typov je oveľa viacej, konkrétna sa ich naučíme z učebnice programovacieho jazyka.

**Deklarácia.** Väčšina programovacích jazykov vyžaduje na začiatku programu stanoviť pre všetky premenné ich typ. Tomu sa hovorí deklarácia premenných.

Napr.: POPIS je typu ZNAK, N je typu CELÉ ČÍSLO, PlochaObdlnika je typu REÁLNE ČÍSLO atď.  
Konkrétny zápis deklarácie opäť závisí na programovacom jazyku.

**Operátory a štandardné funkcie.** Programovací jazyk ponúka základné matematické (+,-,/,\*) a logické (=,<,>,<=,>=, <> [nerovná sa] ) operátory a množstvo štandardných funkcií: ln x, sin(x), sqrt(x) atď.

**Syntax** (vetná skladba) je presné určenie, ako sa majú jednotlivé príkazy a vôbec časti programu zapisovať. Ide o súbor príkazov a pravidiel akými sa v jazyku zapisujú programy. Napr. ak príkaz výpisu na obrazovku má syntax: pís :a, pis "AHOJ", write(a), writeln ('AHOJ')

### Zápis algoritmov:

- ♦ slovné – v prirodzenom jazyku (kuchársky recept) alebo v algoritmickej jazyku (zmes prirodzeného a nejakého konkrétneho programovacieho jazyka)
- ♦ matematicky (vzťahom medzi veličinami, sústavou rovníc, maticami)
- ♦ graficky (vývojový diagram, štruktúrogram)

Algotmickej jazyk – zmes prirodzeného jazyka a príkazov nejakého konkrétneho programovacieho jazyka.

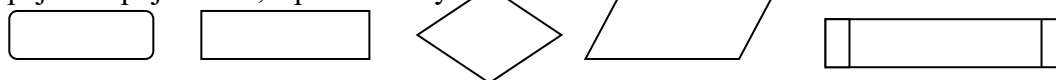
Programovacie jazyk - je sústava slov a pravidiel na zápis algoritmu tak, aby mohol byť spracovaný počítačom.

Vývojový diagram je grafické znázornenie jednotlivých príkazov, z ktorých sa algoritmus skladá, a ich návaznosti pomocou normalizovaných značiek.

#### Základné značky používané vo vývojových diagramoch:

- počiatková a koncová značka (medzná značka)
- priradenia; vykonanie [spracovanie (príkaz, operácia činnosť)]
- rozhodovanie a vetvenie
- vstup a výstup hodnôt
- poznámka
- predom definovaná činnosť

spájanie spojnicami, šípka musí byť ak



### Štruktúra algoritmov

Jednotlivým krokom algoritmu hovoríme tiež príkazy. Množinu príkazov, ktorej procesor rozumie, nazývame slovník jazyka. Pri zápise algoritmu používame nielen jednoduché príkazy (príkaz priradenia) ale aj zložitejšie riadiace príkazy.

#### Základné príkazy (Štruktúra algoritmov)

tie obsahuje snáď každý programovací jazyk

1. sekvencia – postupnosť príkazov
2. vetvenie – členenie algoritmu na základe podmienky
3. cyklus – opakovanie časti algoritmu na základe podmienky

**Postupnosť** (sekvencia) príkazov obsahuje napr. priradovací príkaz ( $X=5$ ), alebo príkaz pre výpočet hodnoty ( $X=Z+5$ )

Príkaz **if** slúži k rozvetveniu programu na 2 vetvy (tzv. binárne vetvenie), pričom program využije len jednu vetvu a druhú preskočí. Ktorú vetvu využije, to závisí od splnenia podmienky (true, false).

**Podmienený príkaz, vetvenie programu podľa podmienok** – veľmi často potrebujeme, aby sa pri splnení nejakej podmienky vykonal príkaz A a pri jej nesplnení príkaz B. To nám umožní podmienený príkaz, často realizovaný príkazom IF (ak platí), THEN (potom urob) a ELSE (ak neplatí, urob).

Príkazy cyklu umožňujú nariadiť viacnásobné opakovanie príkazu alebo postupnosti príkazov. Časť, ktorá má byť opakovaná sa nazýva **telo cyklu**. Ďalej musí každý príkaz cyklu obsahovať určenie počtu opakovaní a to buď:

- explicitne – uvedením konkrétnych čísel (for)



- implicitne - podmienkou (while, repeat)

**Cykly, tj. opakovanie príkazov do splnenia nejakých podmienok alebo po stanovená počet opakovaní** – bežne sa používajú tri druhy cyklov: dva druhy cyklov s podmienkou (cyklus REPEAT...UNTIL a cyklus WHILE...DO) a jeden druh cyklov s pevne stanoveným počtom opakovaní (cyklus FOR) Cyklus WHILE je obecnější (cyklus s podmienkou na začiatku), príkaz v ňom uvedený sa nemusí previesť ani raz, na rozdiel od neho príkaz v cykle REPEAT sa uskutoční ešte pred testovaním podmienky, vždy sa teda aspoň raz vykoná. Cyklus FOR nepotrebuje žiadnu podmienku, musíme však zadať, koľkokrát sa má určitý príkaz vykonať. *Vzhľadom na to, že niekedy aj na vyriešenie jednoduchých úloh je potrebný pomerne zložitý algoritmus, rozdelí sa úloha na určitý počet menších jednoduchších podúloh - dekompozícia. Jednotlivé podúlohy sa spájajú pomocou riadiacich štruktúr. Medzi základné štruktúry patrí sekvencia, vetvenie alebo selekcia, opakovanie (cyklus). Sekvencia je určité množstvo za sebou nasledujúcich príkazov. Pri vetvení dochádza k výberu príkazov (ktoré sa budú vykonávať) v závislosti od splnenia určitej podmienky, alebo selektívny výber v závislosti od hodnoty, ktorú nadobudne tzv. selektor (premenná).*

*Pri použití cyklu sa vykonáva určité množstvo príkazov určitý početkrát. Tento počet je buď presne daný (cyklus s pevným počtom opakovaní), alebo závisí od platnosti podmienky na začiatku alebo na konci cyklu.*

#### Realizácia

Realizácia (prepís algoritmu do programovacieho jazyka – tie majú rôznu syntax – súbor príkazov a pravidiel, akými sa v ňom zapisujú programy)

---

#### **Úlohy:**

1. Napíšte algoritmus na výpočet obsahu a objemu hranola pomocou vývojového diagramu.
2. Napíšte algoritmus na zistenie maximálnej hodnoty z dvoch zadaných čísel.
3. Zapište algoritmus na výpočet mocniny  $a^n$ .
4. Zostavte algoritmus pre výpočet hodnoty súčtu prvých N prirodzených čísel.

#### **Riešenie:**

- 1. rozbor a analýza problému (vstupy, výstupy, požiadavky)**
- 2. vytvorenie algoritmu (vývojový diagram, štruktúrogram)**
- 3. realizácia, testovanie, ladenie**

#### **Otázky:**

1. Čo je to algoritmus?
2. Aké vlastnosti musí mať riešenie úlohy, aby to bol algoritmus (vlastnosti algoritmu)?
3. K čomu slúži vývojový diagram?
4. Popíšte spôsoby zápisu algoritmov a uveďte pomocou akých riadiacich štruktúr algoritmov môžeme zapísať ľubovoľný výpočtový proces.
5. Aký je rozdiel medzi imperatívnym a deklaratívnym programovacím jazykom?
6. Ako delíme programovacie jazyky podľa kritéria – spúšťania programu?
7. Čo je to spustiteľný súbor (program)?
8. Načo slúžia programovacie jazyky?
9. Čo je to prekladač (kompilátor)?
10. Aké konkrétne programovacie jazyky poznáte? (vymenujte aspoň päť)

## Vysvetlenie pojmov jazyka C

### 2. Podmienený príkaz – vnorené podmienky

kedy je potrebné použiť vnorené podmienky v algoritme. Prečo nie je možné vždy použiť príkaz viacnásobného vetvenia ?

### 2. logická premenná

Vysvetlite pojmy: logická premenná, hodnoty logickej premennej

### 3. príkazy vstupu a priradenia

Vysvetlite pojmy: príkaz vstupu, príkaz priradenia. Existuje nejaká ďalšia možnosť načítania údajov do premenných?

*Výrazový príkaz* vzniká pridaním znaku ";" za výraz. Pretože výrazom je aj volanie funkcie (napr. štandardnej).

```
a=1;
b+=1+3*a;
printf("a=%d b=%d\n",a,b);
```

### 4. príkazy vstupu a výstupu

Vysvetlite pojmy: príkazy vstupu a výstupu, textová obrazovka,

### 5. číselné premenné

Vysvetlite pojmy: premenná, konštanta, číselná premenná

**Premenná** je časť operačnej pamäte. Každá premenná má svoje meno – **identifikátor**. Aby sme mohli v programe používať premenné, musíme požiadať systém o pridelenie operačnej pamäte pre ne a pomenovať ich. To všetko sa stane deklaráciou premennej. V programe potom už len používame mená premenných a systém vie, s ktorou časťou operačnej pamäte má pracovať. Takže, **deklarácia premennej je fyzické pridelenie operačnej pamäte pre premennú a dočasné pomenovanie pridelenej pamäte identifikátorom**. Aby systém vedel, koľko operačnej pamäte má pre premennú zadovážiť, musíme v deklarácii uviesť jej typ. **Typ premennej zároveň určuje, čo do nej smieme uložiť**. Jazyk C používa jednoduché a zložené typy premenných.

**Jednoduché typy** sú celé čísla, racionálne čísla, znak a logické hodnoty.

**Zložené typy** sú pole, záznam, súbor.

Skúste si premennú predstaviť ako škatuľku na topánky. Škatuľky môžu byť rôzneho druhu. Druh

škatuľky určuje, čo do nej môžeme vložiť. Ak chceme používať v programe premenné – škatuľky, tak si ich musíme zadovážiť. Zadováži nám ich systém, len mu musíme povedať, koľko škatuliek akého druhu potrebujeme a každej škatuľke musíme dať meno – identifikátor, to aby systém vedel, s ktorou má pracovať. Príkazu na zadováženie si škatuliek a pridelenie mien škatuľkám hovoríme deklarácia.

**Konštanta** je hodnota, ktorá sa nemení, napríklad Ludolfovo číslo  $n$ , tiaž Zeme či DPH. Niekedy sa však zide aj konštantu zmeniť, preto naša vláda občas DPH zmení. Aby sme nemuseli hodnotu konštanty meniť v celom programe, stačí použiť konštanty a zmeniť ju iba na jednom mieste. Ako na to sa naučíme práve dnes.

Troška teórie:

**Konštanta je hodnota, ktorá má svoje meno a hodnotu. Hodnota konštanty sa počas behu programu nemení!**

Podobne ako je to u premenných, tak aj konštanty slúžia ako miesta, kde sa uchovávajú dáta. Ako už samotný názov napovedá, konštanty, na rozdiel od premenných, nemôžu zmeniť svoju hodnotu.

Keď definujeme konštantu, musíme ju inicializovať a neskôr jej už nemôžeme priradiť inú hodnotu.

## 6, Cykly – vnorené cykly, viacnásobné vetvenie

Vysvetlite pojmy: cyklus, vnorený cyklus, viacnásobné vetvenie

Veľakrát sa v živote dostávame do situácie, kedy sa musíme rozhodnúť, ktorou cestou pôjdeme. Rozhodujeme sa na základe nejakých podmienok. Keďže program je návod na riešenie problému zapísaný v programovacom jazyku do počítača aj on musí obsahovať príkaz, ktorý nám umožní na základe podmienok ísť tou správnou cestou. Tým príkazom je príkaz vetvenia.

Troška teórie:

Sekvencia príkazov je zložený riadiaci príkaz. Využíva sa vždy vtedy, **keď potrebujeme z viacerých príkazov vytvoriť jeden príkaz v iných riadiacich príkazoch.**

{ príkaz

príkaz; }

Fungovanie: V sekvencii príkazov sa vykonáva postupne jeden príkaz za druhým tak, ako nasledujú príkazy za sebou.

Všetky programy, ktoré sme doteraz vytvárali tvorili jednu sekvenciu príkazov.

**Vetvenie** používame vtedy, keď nemôžeme postupovať pri riešení problému sekvenčne, ale na základe vyhodnotenia podmienky musíme riešenie dočasne rozdeliť aspoň na dve vetvy. V programe sa vykoná

iba raz!

Príkazy vetvenia sú

- a) **preskok - neúplné vetvenie** - neúplný tvar príkazu vetvenia, testuje logickú podmienku - **if bez else**
- b) **alternujúce vetvenie - úplné vetvenie** - úplný tvar príkazu vetvenia, testuje logickú podmienku - **if s else**
- c) **prepínač** - testuje hodnotu premennej alebo funkcie, má **aspoň** tri vetvy - **switch**

**if (výraz) príkaz1;  
else príkaz2;**

Fungovanie: Najprv sa vyhodnotí logický výraz. Ak je pravdivý, vykoná sa príkaz1, príkaz2 sa preskočí a pokračuje sa za príkazom vetvenia. Ak je nepravdivý, príkaz1 sa preskočí, vykoná sa príkaz2 a pokračuje sa za príkazom vetvenia. **Zložený logický výraz musí tvoriť jeden celok uzavretý v okrúhlych zátvorkách.**

**7.vnorené cykly, zacyklenie** Vysvetlite pojmy: cyklus, vnorený cyklus, zacyklenie programu. Kedy môže dôjsť k zacykleniu ?

Opakuje príkaz donekonečna (kým program nezastavíme).

Napríklad môžeme donekonečna vykresľovať náhodné kruhy, ale pozor, každý kruh berie trochu miesta v pamäti, po čase zaplníte pamäť a počítač môže dramaticky spomaliť, alebo program skončí s chybovou hláškou.

### **8.Cyklus s podmienkou – podmienky a ich použitie**

Vysvetlite: Načo sú podmienky v príkaze cyklu, kedy ich vytvárame a ako ich programovací jazyk vyhodnocuje?

### **9.Cyklus s podmienkou – počet opakovaní**

Vysvetlite dokedy platí podmienka v cykle a ako podmienka vplýva na počet opakovaní cyklu.

### **10. Podmienený príkaz – neúplný**

- syntax neúplného podmieneného príkazu. Uvedením vhodných viet demonštrujte použitie tohto cyklu v bežnej hovorovej reči.

*Podmienený príkaz* umožňuje vykonávať vetvenie priebehu programu. Má dva tvary - skrátenejší a úplný :

```
if (výraz) príkaz; /* skrátenejší */
```

```
if (výraz) príkaz1;  
else príkaz2; /* úplný */
```

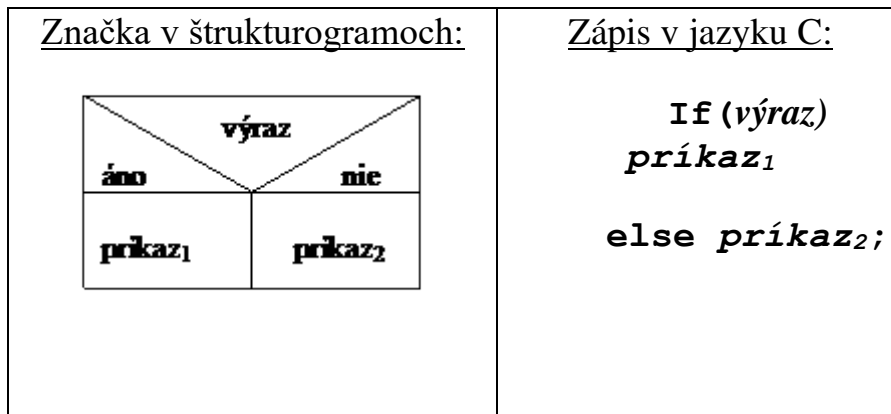
Vetvenie sa vykoná na základe vyhodnotenia podmienky, ktorou je výraz v zátvorkách. Ak je hodnota výrazu nenulová (odpovedá true), vykoná sa príkaz, resp. v prípade úplného tvaru príkaz1. Ak je hodnota výrazu nulová (odpovedá false), vykoná sa iba v druhom prípade príkaz2.

Pri vnáraní podmienených príkazov sa vzťahuje else vždy k najbližšiemu predchádzajúcemu if, preto musíme správne použiť vnorený blok.

### **11. Podmienený príkaz – úplný**

Vysvetlite syntax úplného podmieneného príkazu. Uvedením vhodných viet demonštrujte použitie tohto príkazu v bežnej hovorovej reči.

```
if (výraz) príkaz1;  
else príkaz2; /* úplný */
```



Fungovanie: Najprv sa vyhodnotí logický výraz. Ak je pravdivý, vykoná sa príkaz<sub>1</sub>, príkaz<sub>2</sub> sa preskočí a pokračuje sa za príkazom vetvenia. Ak je nepravdivý, príkaz<sub>1</sub> sa preskočí, vykoná sa príkaz<sub>2</sub> a pokračuje sa za príkazom vetvenia.

**Príklad 2:** Zostavte program, ktorý načíta celé číslo a vypíše či je párne alebo nepárne.

**Riešenie:**

```
#include <stdio.h>  
#include <conio.h>  
main()  
{int x;  
  clrscr();  
  printf("Zadaj cele cislo: ");  
  scanf("%d", &x);  
  if(x%2==0) printf("Zadane cislo je parne.");  
  else printf("Zadane cislo je neparne.");  
  if(getch()==0) getch();  
}
```

Vysvetlite pojmy: logická premenná, logické operácie, zložená podmienka

Logické hodnoty a operácie s nimi      Logické hodnoty sú len dve: pravda (true) a nepravda (false). Na uloženie logických hodnôt používame premenné typu **boolean**. Veľkosť premennej logického typu je 1B.

### Operácie nad logickým údajovým typom

- logická negácia: **not** – počet operandov: 1
- logický súčin: **and** – počet operandov: 2
- logický súčet: **or** – počet operandov: 2
- logický výhradný súčet (nonekvivalencia): **xor** – počet operandov: 2

<i>Hodnoty operandov</i>		<i>Výsledky operácií</i>			
<i>p</i>	<i>q</i>	<i>p and q</i>	<i>p or q</i>	<i>p xor q</i>	<i>not p</i>
false	false	false	false	false	true
false	true	false	true	true	true
true	false	false	true	true	false
true	true	true	true	false	false

Vyššiu prioritu ako ostatné logické operácie má operácia **not**, preto sa vždy vykonáva ako prvá. Operácie **and**, **or** a **xor** sú čo do priority rovnocenné. Ak potrebujete zmeniť prioritu vykonávania operácií, použite zátvorky!

Príklad 1: Akú pravdivostnú hodnotu budú mať logické výrazy pre a=true, b=false a c=true?

- a) **a and b**
- b) **a or b or c**
- c) **not b xor c**
- d) **a xor not c**

### Riešenie:

- a) **a and b = true and false = false**
- b) **a or b or c = true or false or true = true or true = true**
- c) **not b xor c = not false xor true = true xor true = false**
- d) **a xor not c = true xor not true = true xor false = true**

Každý programovací jazyk poskytuje prostriedky, ako počítaču povedať, že nejaké príkazy má vykonať len za istých okolností, že niečo má opakovať dovtedy, kým nie je splnená nejaká podmienka alebo že si má vybrať z viacerých možností, ako pokračovať v tom, čo robí. Všetky tieto veci vyžadujú, aby sa dalo zistiť, či sú splnené nejaké podmienky.

Na zápis podmienok sa v jazyku C používajú **logické výrazy**. (Niekedy sa nazývajú aj **booleovské** na pamiatku pána Boolea, ktorý ich študoval z

matematického hľadiska.) Logické výrazy môžu nadobúdať dve hodnoty -- hodnotu **pravda** (anglicky **true**) a hodnotu **nepravda** (anglicky **false**). Niektoré jazyky (napr. Pascal) majú pre logické výrazy samostatný typ premenných. V C-čku sa však pre logické hodnoty používa typ **int**. Pričom hodnota 0 znamená "nepravda" a akákoľvek iná hodnota znamená "pravda".

Logické výrazy môžu testovať, či má alebo nemá premenná nejakú hodnotu, môžu sa v nich porovnávať hodnoty a jednoduché výrazy sa môžu spájať do zložitejších. Na porovnávanie dvoch čísel slúži operátor `==`. (Tie rovnítko tam musia byť dve, pretože jedno rovnítko slúži ako príkaz priradenia.) Takže výraz `cis == 7` nadobúda hodnotu "pravda", ak je v premennej `cis` hodnota 7 a inak nadobúda hodnotu "nepravda". Podobne sa dajú použiť znamienka `>`, `>=`, `<`, `<=`, ktoré znamenajú "väčší", "väčší alebo rovný", "menší", "menší alebo rovný". Na test, či sú dve hodnoty rôzne, sa používa operátor `!=`. Takže výraz `7 > a` nadobúda hodnotu "pravda" len vtedy, ak je v premennej `a` uložená hodnota menšia ako 7 a výraz `i != j` nadobúda hodnotu "pravda" iba vtedy, keď sú hodnoty uložené v premenných `i` a `j` rôzne. Ako sú logické výrazy vyhodnocované uvidíte v úlohe:

**Úloha č.1:** Pozrite si nasledovný program a na papier si napíšte váš tip, čo bude vypisovať. Potom program skompilujte a spustite. Porovnajete si vaše tipy s realitou. Vysvetlite, prečo to píše to, čo to píše.

```
#include <stdio.h>

main()
{
    int i = 3, j = 4;

    printf("%d\n", i == 3);
    printf("%d\n", j >= 6);
    printf("%d\n", j = 5);
    printf("%d\n", i < 6);
    printf("%d\n", j == 4);
}
```

Niekedy je nutné v jednej podmienke otestovať viacero vecí naraz. Napríklad potrebujete zistiť, či je aj premenná `i` aj premenná `j` kladná. Každú vec zvlášť viete zistiť jednoducho (`i > 0`, `j > 0`). Oba testy je však treba nejakým spôsobom spojiť. Ak chcete zistiť či sú splnené obidva naraz, použije sa operátor `&&`. Takže podmienka, ktorá zistí, či sú hodnoty v oboch premenných kladné sa napíše `i > 0 && j > 0`.<sup>0.6</sup>

Ak potrebujeme zistiť, či je splnená aspoň jedna z týchto podmienok, použijeme operáciu `||`.<sup>0.7</sup> Takže výraz `i > 0 || j > 0` nadobúda hodnotu "nepravda" jedine vtedy, ak sú hodnoty oboch premenných `i` a `j` menšie alebo rovné ako nula. Stačí, keď je hodnota v jednej z premenných kladná (kľudne môžu byť aj obidve), podmienka nadobudne hodnotu "pravda".

Ak chceme hodnotu logického výrazu zmeniť na opačnú, používa sa na to operátor `!`.<sup>0.8</sup> Ak napríklad chceme napísať podmienku "v premennej `znak` nie je veľké písmeno", (teda podmienku, ktorá nadobudne hodnotu "pravda" ak v tej premennej veľké písmeno nie je a hodnotu "nepravda" inak), budeme postupovať nasledovne: Najprv si napíšeme podmienku, ktorá nám zistí, či v



premennej znak veľké písmeno je. Taká podmienka môže vyzerat' napríklad takto:

znak >= 'A' && znak <= 'Z'. Táto podmienka však funguje presne naopak, než potrebujeme. Ale vďaka operátoru ! z nej vyrobíme podmienku, ktorú potrebujeme nasledovným spôsobom: !(znak >= 'A' && znak <= 'Z')

### 13. Príkaz s podmienkou

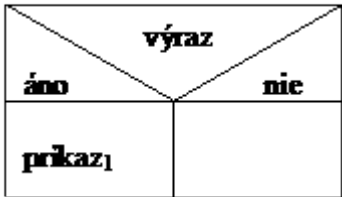
Vysvetlite pojmy: podmienený príkaz if, dokedy trvá počas behu programu

Príkazy vetvenia: Príkaz vetvenia je zložený riadiaci príkaz. Používame ho vtedy, keď potrebujeme, aby sa určitý príkaz alebo príkazy vykonal/vykonal iba vtedy, keď je splnená nejaká podmienka. V programe sa vykoná iba raz!

Príkazy vetvenia sú

- a) **preskok** – neúplný tvar príkazu vetvenia, testuje logickú podmienku – **if** bez else
- b) **alternujúce vetvenie** – úplný tvar príkazu vetvenia, testuje logickú podmienku – **if** s else
- c) **prepínač** – testuje hodnotu premennej alebo funkcie, má aspoň tri vetvy – **switch**

Preskok

<u>Značka v štrukturogramoch:</u> 	<u>Zápis v jazyku C:</u> <b>If(výraz) príkaz1;</b>
--	---

Fungovanie: Najprv sa vyhodnotí logický výraz. Ak je pravdivý, vykoná sa príkaz<sub>1</sub> a potom sa pokračuje za príkazom vetvenia. Ak je nepravdivý, príkaz<sub>1</sub> sa preskočí a pokračuje sa za príkazom vetvenia.

Príklad 1:

Zostavte program, ktorý načíta celé číslo a vypíše jeho absolútnu hodnotu.

Riešenie:

```
#include <stdio.h>
#include <conio.h>
main()
{int x;
 clrscr();
 printf("Zadaj cele cislo: ");
```

```

scanf("%d",&x);
if(x<0) x=x*(-1);
printf("Absolutna hodnota zadaneho cisla je %d",x);
if(getch()=='0') getch();
}

```

#### 14. Cyklus – porovnanie príkazov cyklu

Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?

*Príkaz cyklu* sa používa pre riadenie opakovania nejakého výpočtu. Určuje spôsob a podmienku ukončenia opakovania. Jazyk C pozná tri varianty príkazu cyklu:

- cyklus while
- cyklus do while
- cyklus for

Cyklus *while* má tvar :

```
while (výraz) príkaz;
```

Výraz tvorí podmienku ukončenia cyklu. Cyklus sa vykonáva, kým je výraz nenulový (true). Telo cyklu sa teda nemusí vykonať ani raz.

Cyklus *do while* má tvar :

```
do
príkaz;
while (výraz);
```

Výraz tvorí podmienku ukončenia cyklu. Cyklus sa vykonáva, kým je výraz nenulový (true), najmenej však jedenkrát. Použitie príkazu *do while* je pomerne zriedkavé v porovnaní s ostatnými dvoma príkazmi cyklu.

Cyklus *for* má tvar :

```
for (init ; koniec ; zmena) príkaz;
```

kde *init* je výraz, ktorý sa vyhodnotí raz na začiatku cyklu, obvykle sa tu nastavujú počiatočné hodnoty cyklu.

*Koniec* je podmienka ukončenia cyklu (kým je pravdivá - nenulová).

*Zmena* sa vyčísli po každom vykonaní cyklu, slúži pre zmenu riadiacej premennej cyklu. Táto zmena riadiacej premennej sa nemá uskutočňovať v tele cyklu, ale iba v hlavičke cyklu.

#### 15. Podmienený príkaz - úplné a neúplné vetvenie

Vysvetlite pojem podmienka, vysvetlite rozdiel medzi úplným a neúplným vetvením.

## 16. Cyklus – prerušenie cyklu

Vysvetlite pojmy: cyklus, preskočenie krokov v cykle (skok), predčasné ukončenie cyklu (prerušenie).

- Z cyklu vieme vyskočiť príkazom `break`, prejsť na ďalšiu iteráciu príkazom `continue`

*Príkazy skoku* sa používajú pre ošetrovanie, resp. riešenie zvláštnych či výnimočných situácií v programe. Ide o situácie, ktoré je spravidla potrebné ošetriť jednorázovo. Medzi príkazy skoku možno zaradiť príkazy :

- `break`
- `continue`
- `return`
- `goto`

Príkazy *break* a *continue* sa používajú predovšetkým v spojení s príkazmi cyklu. Príkaz *break* ukončuje najvnútornejšiu neukončenú slučku cyklu a opúšťa okamžite telo cyklu, to znamená, že spôsobuje skok z príkazu cyklu na ďalší príkaz. Ďalšie použitie príkazu `break` je v príkaze `switch`

Príkaz *continue* preruší vykonávanie najbližšej nadradenej štruktúrovanej konštrukcie cyklu (najvnútornejšiu neukončenú slučku cyklu) a spôsobí prechod na ďalší krok cyklu bez toho, aby sa predchádzajúci krok cyklu ukončil. Dôležité je, že nespôsobuje skok z tela cyklu von..

Príkaz *return* spôsobí ukončenie vykonávania nejakej funkcie a súčasne môže exportovať hodnotu výrazu do nadradenej úrovne ako deklarovaný typ hodnoty funkcie. Môže sa vyskytovať v jednom z tvarov :

```
return výraz;  
return (výraz);  
return;
```

Príkaz *goto* umožňuje vykonať skok tak v rámci jednej funkcie, ako aj medzi jednotlivými funkciami. Je však otázka, aký veľký zmysel takého skoky majú. Obecne je možné vykonať skok z bloku aj do bloku, a samozrejme von z cyklu. Všeobecne sa doporučuje používať príkaz `goto` čo najmenej a skutočne len vo výnimočných prípadoch. Príkaz, na ktorý sa má uskutočniť skok, musí byť označený návestím.

## 17. Cyklus – porovnanie cyklov (viz. 26 , viz. 14)

Vysvetlite význam a použitie cyklu `for` a cyklu `while`.

**for**

**while**

## 18. Polia – jednorozmerné pole reálnych čísel

Vysvetlite pojmy: jednorozmerné pole, prvok poľa, index prvku. Akým spôsobom deklarujeme pole reálnych čísel?

Jednorozmerné pole patrí medzi zložené typy údajov. *Zložené typy údajov* sú také, ktorých hodnoty môžu byť zložené z viacerých jednoduchých hodnôt (položiek).

Zložené typy ..... *pole* - pevný počet zložiek rovnakého typu údajov

\_\_\_\_\_ *záznam* - pevný počet zložiek rôznych typov

Pole je pomenovaná skupina premenných **rovnakého** typu, uložených v pamäti bezprostredne za sebou. Tieto premenné označujeme ako **prvky poľa**.

Polia delíme na **jednorozmerné a viacrozmerné**.

Pole **deklarujeme** zadaním typu, za ktorým nasleduje názov poľa a subskript (alebo aj index). Subskript je počet prvkov poľa uzatvorený v hranatých zátvorkách.

*typ názov[n];*

Napríklad:

*long PoleCisel[25];* deklaruje pole 25 celých čísel nazvané *PoleCisel*.

Keď kompilátor vidí túto deklaráciu, vyhradí si dostatok pamäte pre uchovanie všetkých 25 prvkov. Pretože každý prvok typu *long* vyžaduje 4 bajty, táto deklarácia vyhradzuje 100 za sebou nasledujúcich bajtov pamäte.

### Prvky poľa

K jednotlivým prvkom poľa pristupujeme prostredníctvom **indexu**.

Prvky poľa sa počítajú od nuly. Preto je prvým prvkom pola *nazovPola[0]*.

V príklade *PoleCisel[25]* je prvým prvkom poľa *PoleCisel[0]*, druhým prvkom *PoleCisel[1]*, atď.

Pole *NejakePole[3]* má tri prvky. Sú to *NejakePole[0]*, *NejakePole[1]* a *NejakePole[2]*. Všeobecne teda platí, že *NejakePole[n]* má *n* prvkov, ktoré sú očíslované od *NejakePole[0]* až po *NejakePole[n-1]*. Opäť však musíme myslieť na to, že index je odsadením, takže prvý prvok poľa je nula úložných miest od začiatku poľa, druhý je vzdialený jedno úložné miesto, atď. Preto je *PoleCisel[25]* číslované od *PoleCisel[0]* po *PoleCisel[24]*.

## 19. Polia – dvojrozmerné celočíselné pole

Vysvetlite pojmy: dvojrozmerné pole, prvok poľa, indexy prvku. Akým spôsobom deklarujeme celočíselné pole?

### Viacrozmerné pole

**Je možné mať aj pole, ktoré má viac, ako len jeden rozmer.**

**Každý rozmer** je predstavovaný ako určitý (**index**) v poli. Preto má dvojrozmerné pole dva indexy, trojrozmerné pole má tri indexy, atď.

Polia môžu mať ľubovoľný počet rozmerov, ale napriek tomu bude mať väčšina našich polí jeden alebo maximálne dva rozmery. Dobrým príkladom dvojrozmerného poľa je šachovnica. Jeden rozmer predstavuje osem riadkov a druhý rozmer predstavuje osem stĺpcov.

Deklarovanie **dvojrozmerného poľa** bude vyzerat' takto:

*typ názov[n][m];*

Tie isté dáta by sme mohli reprezentovať tiež aj jednorozmerným poľom s rozmerom  $n \times m$  štvorčekmi.

### **Inicializácia viacrozmerných polí**

Viacrozmerné polia je možné aj inicializovať. **Postupne prirad'ujeme zoznam hodnôt prvkom poľa, pričom posledný index poľa sa mení, zatiaľ čo predchádzajúci zostáva nezmenený.**

Ak máme teda pole:

*int Pole[5][3];*

potom sa prvé tri prvky uložia do *Pole[0]*, druhé tri do *Pole[1]*, atď.

Uvedené pole inicializujeme zadaním:

*int Pole[5][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 };*

**Kvôli zreteľnosti** môžeme zoskupiť inicializácie zloženými zátvorkami.

**Napríklad:**

*int Pole[5][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12}, {13, 14, 15} };*

**Kompilátor** síce vnútorné zložené zátvorky **ignoruje**, ale takýto zápis nám pomáha v pochopení rozdelenia čísel vo viacrozmernom poli.

Jednotlivé hodnoty musia byť oddelené čiarkami bez ohľadu na zložené zátvorky. Celá inicializácia musí byť v zložených zátvorkách a musí sa končiť bodkočiarkou.

## **20. Základy – príkaz výstupu.**

Vysvetlite pojmy: príkaz výstupu, textová obrazovka

## **21. : Polia – celočíselné pole, náhodné čísla**

Vysvetlite pojmy: pole, prvok poľa, náhodné číslo, generátor náhodných čísel.

## **22. Reťazce – kódovanie znakov**

Vysvetlite pojmy: znak, znaková premenná, ASCII tabuľka

Premenná, do ktorej môžeme uložiť jeden znak, môže byť iba jediného údajového typu, a to **char**. Je z rozsahu 0 až 255 (binárne kódy znakov) a jeho veľkosť je 1B.

Ako iste viete, každý znak je umiestnený do tabuľky ASCII kódu a jeho miesto v tabuľke určuje hodnotu jeho ASCII kódu.

### Tabuľka ASCII kódu pre stredoeurópske krajiny (kódová stránka 1250):

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL 007F
80	€ 20AC		ı 201A	ı 201C	ı 201D	ı 2022	ı 2026	ı 2020	ı 2021	ı 2030	Š 0160	<	Š 016A	Ť 0164	Ž 017D	Ž 0179
90		ı 2018	ı 2019	ı 201C	ı 201D	ı 2022	ı 2026	ı 2020	ı 2021	ı 2122	š 0161	>	š 016B	ť 0165	ž 017E	ž 017A
A0	NBSP 00A0	ˆ 02C7	ˆ 02D8	Ł 0141	» 00A4	Ł 0104	ı 00A6	Š 00A7	ı 00A8	© 00A9	Š 015E	« 00AB	ı 00AC	ı 00AD	ı 00AE	ı 017B
B0	° 00B0	± 00B1	ı 02DB	ı 0142	ı 00B4	ı 00B5	ı 00B6	ı 00B7	ı 00B8	ı 0105	ı 015F	ı 00BB	ı 013D	ı 02DD	ı 013E	ı 017C
C0	Ř 0154	Á 00C1	Ā 00C2	Ā 0102	Ā 00C4	Ā 0139	Ā 0106	Ā 00C7	Ā 010C	Ā 00C9	Ā 0118	Ā 00CB	Ā 011A	Ā 00CD	Ā 00CE	Ā 010E
D0	Đ 0110	Ň 0143	Ň 0147	Ó 00D3	Ö 00D4	Ö 0150	Ö 00D6	Ö 00D7	Ö 0158	Ö 016E	Ö 00DA	Ö 0170	Ö 00DC	Ö 00DD	Ö 0162	Ö 00DF
E0	ř 0155	á 00E1	ā 00E2	ā 0103	ā 00E4	ā 013A	ā 0107	ā 00E7	ā 010D	ā 00E9	ā 0119	ā 00EB	ā 011B	ā 00ED	ā 00EE	ā 010F
F0	đ 0111	ň 0144	ň 0148	ó 00F3	ö 00F4	ö 0151	ö 00F6	ö 00F7	ö 0159	ö 016F	ö 00FA	ö 0171	ö 00FC	ö 00FD	ö 0163	ö 02D9

Napr. ASCII kód písmena K určité takto: Písmeno **K** sa nachádza v stĺpci s označením 64 a v riadku s označením 11. Jeho ASCII kód je **64+11=75**.

Ak by sme chceli do premennej x typu **char** uložiť písmenko K, tak by sme to zapísali buď takto: **x='K'**; alebo takto: **x=75**;

Ak by sme chceli do premennej x uložiť číslicu 5, tak by sme to zapísali buď takto: **x='5'**; alebo zistíme v tabuľke ASCII kódu kód číslice a zapíšeme to takto: **x=53**;

**V premennej x je vždy uložený ASCII kód znaku a sú pre ňu definované všetky operácie, ktoré môžeme používať s celočíselnou premennou okrem absolútnej hodnoty.**

### 23. Reťazce – práca s reťazcami po znakoch

Vysvetlite pojmy: znak, znaková premenná, reťazec

**Reťazec je postupnosť znakov** uložených za sebou v pamäti. Reťazce ponúkajú vhodný spôsob ukladania textových informácií.

Používame dva spôsoby práce s reťazcami. Prvým, je spôsob, ktorý bol prevzatý od jazyka C, druhý, je využitie knižnice string.

Zatiaľ postupnosť znakov uložených za sebou v pamäti predstavuje pole. Reťazec je pole prvkov typu char, kde má každý znak svoj prvok poľa.

**Reťazce majú ešte jednu zvláštnu vlastnosť, posledný znak každého reťazca je nulový znak. Tento znak má zápis '\0' a má kódovú hodnotu ASCII 0, slúži na ukončenie reťazca.**

Pozrite sa na nasledujúce dve deklarácie:

```
char pes[5] = {'p', 'e', 't', 'k', 'o'}; //nie je retazec
char macka[5] = {'c', 'i', 'c', 'a', '\0'}; //retazec
```

Obidve polia sú typu char, ale iba druhé je reťazcom. Jazyk C++ ponúka **funkcie** pre prácu s reťazcami, ktorá používa objekt **cout**. Všetky funkcie spracúvajú reťazce, až kým nenarazia na nulový znak.

Ak použijeme cout, aby vypísal premennú macka, tak vypíše cica, prečíta nulový znak a skončí. Ale ak mu dáme vypísať premennú pes a on vypíše pole znakov.

Podľa príkladu poľa macka vyzerá inicializácia dosť zložito. Ale existuje aj **jednoduchší spôsob inicializácie reťazcov**. Stačí **jednoducho použiť reťazec v úvodzovkách, ktorý sa nazýva reťazcová konštanta**.

Napr.:

```
char meno[10] = "Pan Peter"; //'\0' sa doplni samo ale musite
mu tam nechat miesto
char jav[] = "Bublinky"; //Velkost si spocita prekladac a
doplni si sam '\0'
```

Za reťazcom v úvodzovkách je **vždy doplnený** nulový znak, tak ho nemusíte vypisovať. Taktiež aj vstupy z klávesnice si samy dopĺňajú nulový znak. Ale musíme si byť istý, že pole je dostatočne veľké pre uloženie všetkých znakov, **vrátane** nulového znaku. Inicializácia znakového poľa reťazcovou konštantou je jeden z prípadov, kedy môže byť bezpečnejšie nechať určenie počtu prvkov na prekladači. Ak vytvoríte pole väčšie než je reťazec, okrem zaberaniu zbytočne pamäte neurobíte nič zlé. Je to preto, lebo funkcie pracujúce s reťazcami, sa neriadia veľkosťou poľa ale hľadajú nulový znak.

**Zapamätajte si: Pri určovaní minimálnej veľkosti poľa pre uloženie reťazca nezabudnite na koncový nulový znak.**

Všimnite si, že reťazcová konštanta je nezameniteľná za znakovú konštantu. Znaková konštanta je „S“ predstavuje zápis kódu znaku. Pre systém so znakovou sadou ASCII je 'S' iba iným zápisom čísla 83. Preto príkaz:

```
char znak = 'S'; //toto je v poriadku
```

priradíme hodnotu 83 premennej znak.

**Ale reťazcová konštanta „S“ reprezentuje reťazec zložený z dvoch znakov S a '\0'**. V skutočnosti predstavuje **adresu pamäte, na ktorej je reťazec uložený**. Takže príkaz:

```
znak = "S"; // nepovolene miesanie typov
```

sa pokúša premennej znak priradiť pamäťovú adresu.

## **24. Podmienený príkaz – viacnásobné vetvenie** vysvetlite pojem vetvenie, viacnásobné vetvenie, prepínač

*Prepínač* umožňuje vykonávať vetvenie programu na viac ako dve vetvy. Vetvenie sa rovnako ako v predchádzajúcom príkaze if vykonáva na základe vyhodnotenia výrazu, ktorý je celočíselného typu. Ak sa nájde zhoda medzi hodnotou výrazu a cestou case,

pokračuje výpočet touto vetvou (až po príkaz break), inak je skok na default vetvu. Tvar príkazu je :

```
switch (výraz){  
case hodn1 : prikaz1;  
...  
case hodnn : prikazn;  
default : prikaz;  
}
```

resp.

```
switch (výraz){  
case hodn1 : prikaz1;break;  
...  
case hodnn : prikazn;break;  
default : prikaz;  
}
```

V prvom prípade sa vykonajú všetky príkazy od zvolenej v

etvy až po default, v druhom prípade je výpočet každej vetvy ukončený príkazom break, to znamená že výpočet sa príkazom switch (v Pascale ekvivalent CASE) rozvetví a po jeho ukončení opäť spojí. Spôsob ošetrenia vetvy default záleží na programátorovi.

## 25. : Aritmetika – celočíselná a reálna premenná

Vysvetlite pojmy celočíselná a reálna premenná a ako sa deklarujú v jazyku C. Vymenujte základné matematické operácie pre oba typy premenných.

Základné objekty s ktorými program, napísaný v jazyku C, narába sú **premenné** a **konštanty**. Na ich pomenovanie sa používajú identifikátory. Používajú sa v algebraických a logických výrazoch ako symboly predstavujúce hodnoty. *Premenné* sú dátové objekty so zmeniteľným obsahom (na priradenie hodnoty používame priradovací výraz). Hodnota *konštant* sa raz určí a potom už nemení. Proces, ktorý premenným prideluje *názvy a typové označenie* sa nazýva **deklarácia**. Ak sa premennej vyhradí v pamäti priestor, prípadne sa aj nastaví počiatočná hodnota hovoríme o **definícii** premennej.

Tretím dôležitým dátovým objektom sú **funkcie**. Zatiaľ čo funkcia v matematike predstavuje matematický výraz ktorý transformuje vstupné atribúty do výsledku tzv. funkčnej hodnoty, v **C jazyku je "funkcia" program**, ktorý spracúva hodnoty vstupných premenných a výsledok umiestňuje do výstupnej premennej. Tu tiež, tak ako v matematike, dochádza k transformácii vstupných hodnôt na výstupnú hodnotu. **Funkcia sa deklaruje**, podobne ako premenná, určením identifikátora výstupnej hodnoty a jej typu. Na rozdiel od premennej sa za identifikátorom názvu funkcie uvádza do malých okrúhlych zátvoriek umiestnená deklarácia vstupných premenných - **formálne parametre** - ktoré používa funkcia v programe (záleží na poradí). Deklarácia funkcie je ako každý príkaz ukončená bodkočiarkou. **Definícia funkcie** má inú úlohu ako definícia premennej. Formálne začína rovnako ako deklarácia funkcie, ale miesto bodkočiarky je na konci prilepený programový blok. Použitie názvu funkcie (t.j. identifikátor funkcie) vo **výrazoch** je podobné ako premenných a konštant. Rozdiel medzi nimi spočíva v okrúhlych zátvorkách za identifikátorom funkcie, v ktorých môžu ale nemusia (v *prípade tzv. prázdnej hodnoty - void*) byť uvedené vstupné hodnoty atribútov funkcie.

Názvy premenných aj funkcií majú vo výrazoch rovnaké použitie - predstavujú hodnoty identifikované ich identifikátormi. Dej ktorý je za tým, je však iný v prípade premennej a iný v prípade funkcie. Zatiaľ čo premenná poskytne svoju hodnotu hneď, lebo jej identifikátor určuje adresu umiestnenia hodnoty, funkcia predtým najprv spustí svoj program. Pri štarte sa hodnoty atribútov uvedených v malých okrúhlych zátvorkách skopírujú do formálnych premenných uvedených v deklarácii a program sa vykoná. Výsledok programu - funkčná hodnota - sa zapíše na pozíciu, ktorú určuje identifikátor funkcie. Ďalej to už prebieha ako s hociktorou inou dátovou premennou.



## Vlastnosti dátového objektu

### Dátový typ

Udáva vlastnosti údajov uloženého v dátovom objekte.

#### základné

**int** celé čísla, pre údaj je vyčlenené jedno pole bajtov

**float, double** reálne číslo jednoduchej a dvojnásobnej presnosti, dve polia pre celú a pre desatinú časť

- **char** jeden bajt schopný uchovať jeden znak zo súboru znakov ( najčastejšie ASCII znakov)

Tabuľka č. 1

Dátový typ	Počet bitov	Význam
<b>char</b> , unsigned char, signed char	8	znak
<b>short</b> , unsigned short, signed short	16	krátke celé číslo
<b>int</b> , unsigned int, signed int	16 alebo 32	celé číslo
<b>long</b> , unsigned long, signed long	32	dlhé celé číslo
<b>enum</b>	8 16 32	výčtový typ
<b>float</b>	32	reálne číslo
<b>double</b>	64	reálne číslo s dvojnásobnou dĺžkou
<b>long double</b>	80	ešte dlhšie re.čí.
pointer	16 32	smerník

Pre bližšie určenie dĺžky celého čísla sa používajú kvalifikátory short a long. Kvalifikátor unsigned ( bez znamienka) pripúšťa len celé kladné čísla no dvojnásobnej dĺžky.

napr.:

```
short    int x;
long     int y;
unsigned int z;
```

V jazyku C chýba typ logický (boolean). Tento je nahradený typom int, kde nulová hodnota je totožná s hodnotou (FALSE) a nenulová s hodnotou true (TRUE).

Výslednou hodnotou relačných operátorov v prípade true je jednotka.

#### odvodené

Odvodené dátové typy sú vždy zložené zo základných dátových typov alebo ich kombinácií.

- pole dátový objekt zložený z dátových objektov rovnakého dátového typu
- štruktúra dátový objekt zložený z dátových objektov rôzneho dátového typu
- smerník reprezentuje adresu daného objektu

#### **Veľkosť**

Každý dátový objekt má určený dátový typ, ktorý má jednoznačne určenú svoju veľkosť ( udáva sa v Bajtoch)

Veľkosti dátových objektov:

- celočíselné 2B až 4B
- reálne 4B až 8B
- znak 1B
- logický 1bit
- pole veľkosť poľa sa určí ak počet prvkov vynásobíme veľkosťou prvku
- štruktúra veľkosť štruktúry určíme ak sčítame veľkosti všetkých prvkov tvoriacich štruktúru

*Pozn.* Viac informácií o veľkosti jednotlivých dátových objektoch vid'. tabuľka č.1

### Adresa

Je to číslo prvej pamäťovej bunky, kde sa v operačnej pamäti začína dátový objekt. Adresa je celé kladné číslo a udáva sa v hexadecimálnej sústave

### Obsah

Hodnota, ktorá sa do dátového objektu ukladá. Je to vždy číslo. Význam tohto čísla určuje dátový typ.

### Obor hodnôt

Veľkosť operačnej pamäte je ohraničená preto aj dátové objekty majú svoje ohraničenie. Hovoríme tomu, že dátový objekt má svoj obor hodnôt.  
napr.

Význam	Zápis	Veľkosť	Obor hodnôt
celé číslo so znamienkom	int	2B	{-32768,...,32767}
celé číslo bez znamienk	unsigned int	2B	{0,.....65535}
znak so znamienkom	signed char	1B	{-128,...,127}
znak bez znamienka	unsigned char	1B	{0,.....255}

### Možnosť meniť obsah

Dátový objekt, ktorý mení svoj obsah počas behu programu nazývame **premenná**.

Dátový objekt, ktorý nemení svoj obsah počas behu programu nazývame **konštanta**.

### Meno

Meno dátového objektu vyjadruje identifikátor. Názov by mal vyjadrovať použitie dátového objektu napr. obvod\_stvorca alebo pocet\_ziakov.

### Rozsah platnosti

Dátový objekt, ktorý má platnosť v celom programe nazývame - globálny.

Dátový objekt, ktorý má platnosť len v určitej časti programu (napr. v prgramovom bloku funkcie) nazývame - lokálny.

- **Globálne premenné** sú definované mimo akejkol'vek funkcie
  - môžu ich používať všetky funkcie definované pod nimi
- **Lokálne premenné** sú definované vo vnútri funkcie
  - môže ich používať iba táto funkcia

## 26.Cyklus [https://spseke.sk/tutor/projekt/c\\_jazyk.htm](https://spseke.sk/tutor/projekt/c_jazyk.htm)

Vysvetlite pojem cyklus, ako sa deklaruje v jazyku C, vysvetlite podmienky behu cyklu

### Príkazy cyklu - iteračné príkazy

Príkazy cyklu slúžia na opakovanie určitej časti programu za splnenia podmienky. Príkazy, ktoré sa majú v cykle opakovať nazývame telo cyklu.

### Poznáme cykly

s neurčitým počtom opakovaní: **WHILE, DO WHILE** sú to cykly u ktorých nevieme koľko krát sa majú príkazy v tele cyklu opakovať s určitým počtom opakovaní: **FOR** je to cyklus kde vieme koľko krát sa majú príkazy v tele cyklu opakovať

Každý z cyklov sa vykonáva dovtedy, pokiaľ je hodnota výrazu **nenulová**.

### Príkaz while

V cykle while sa vyhodnocuje výraz ešte pred vstupom do tela cyklu. To znamená, že príkazy v tele cyklu sa vykonajú iba ak bude mať výraz nenulovú hodnotu.

Pri tomto type príkazu sa môže stať, že cyklus neprebehne ani raz.

syntax:            **while** (podmienka)  
                         príkaz;

alebo

syntax:            **while** (podmienka)  
                         {  
                             príkaz1;  
                             ...  
                             príkazn;  
                             }

Ak je hodnota výrazu nenulová vykonajú sa príkazy alebo príkaz v tele cyklu a znovu sa vyhodnotí výraz. Cyklus sa opakuje dovtedy, pokiaľ výraz nenadobudne nulovú hodnotu.

Potom sa cyklus ukončí a program pokračuje ďalším príkazom.

### Príkaz do while

Príkaz do while je podobný príkazu while s tým rozdielom, že teraz sa podmienka vyhodnocuje až po prvom vykonaní tela cyklu. Príkazy v tele cyklu sa budú vykonávať dovtedy, pokiaľ je podmienka splnená. Pri tomto type príkazu určite cyklus prebehne aspoň raz.

syntax:            **do**  
                         príkaz;  
                         **while** (výraz);

alebo

syntax:            **do**  
                         {  
                             príkaz1;  
                             ...  
                             príkazn;  
                             } **while** (výraz);

Najskôr sa vykonajú príkazy resp. príkaz v tele cyklu a potom sa vyhodnotí výraz. Ak je výraz pravdivý t.j. má nenulovú hodnotu, telo cyklu sa opäť vykoná. Ak je pravdivý, cyklus sa skončí.

Cyklus do while je možné použiť všade tam kde treba telo cyklu určite aspoň raz vykonať.

## Príkaz for

Cyklus **for** je cyklus so známym počtom opakovaní. Cyklus so známym počtom opakovaní je taký cyklus, pri ktorom vieme koľko krát sa majú príkazy v tele cyklu opakovať.

syntax:            **for** (výraz1;výraz2;výraz3)  
                         príkaz;

alebo

syntax:            **for** (výraz1;výraz2;výraz3)  
                         {  
                             príkaz1;  
                             ...  
                             príkazn;  
                             }

výraz1: inicializácia riadiacej premennej cyklu (nastavenie počiatocnej hodnoty premennej napr.(i=0 alebo j=10))

výraz2: je relačný výraz a pokiaľ je jeho hodnota rôzna od nuly, telo cyklu sa bude opakované vykonávať napr. (i>10 alebo j<0)

výraz3: je výraz v ktorom riadiacu premennú cyklu inkrementujeme alebo dekrementujeme napr.(i++ alebo i--)

napr.:

Zápis cyklu **for** ak príkazy v tele cyklu sa majú opakovať 10 - krát.

```
for(i=0;i<10;i++) // po vykonaní príkazov bloku sa "i" zvýši o 1
postinkrement
{
    príkaz1;
    ...
    príkazn;
}
```

Pred vstupom do cyklu sa premenná **i** nastaví na hodnotu **i=0** (výraz1). Príkazy v tele cyklu sa opakované vykonávajú pokiaľ je splnená podmienka **i<10** ( výraz2 má nenulovú hodnotu).

Po každom vykonaní cyklu sa vykoná inkrementácia premennej **i=i+1** (výraz3).

Príkaz **for** môžeme použiť aj pre vytvorenie nekonečného cyklu. To dosiahneme vynechaním výrazu2 v hlavičke.

```
for (výraz1; ; výraz3) { }
```

V tom prípade treba do tela cyklu dať príkaz, ktorý zabezpečí ukončenie cyklu ako **break** alebo **return**. Ak nepotrebujeme v hlavičke žiaden výraz, dvojicu **;;** musíme ponechať.

## 27. Podmieneny príkaz IF

Vysvetlite pojem podmienený príkaz<sup>28</sup>. Cykly – porovnanie cyklov

Vysvetlite rozdiely medzi príkazom cyklu s pevným počtom opakovaní a príkazom cyklu s podmienkou. Ako sa rozhodneme, ktorý z nich použiť?

### Podmieneny cyklus s podmienkou na začiatku

Predstav si situáciu: je pred Vianocami a Ty vykrajuješ z cesta perníčky. V podstate opakuješ činnosť: vykroj perníček, polož ho na plech. Robíš cyklus. Niekedy však musí cyklus skončiť. Keďže nevieš dopredu, koľko perníčkov máš vykrojiť, podmienkou ukončenia Tvojho cyklu je množstvo cesta. Skráteno by sme mohli povedať: „Kým máš cesto, vykrajuj perníčky.“ Dnes sa naučíme zapísať v programovacom jazyku podobné príkazy.

Troška teórie:

### Pokiaľ je podmienka splnená vykonaj činnosť

*while* výraz *do* telo\_cyklu;

Fungovanie: Najprv sa vyhodnotí pravdivosť logického výrazu. Ak je tento výraz pravdivý, vykoná sa telo cyklu (jeden príkaz alebo blok príkazov), a opäť sa testuje jeho pravdivosť. Ak je opäť výsledok pravdivý, pokračuje sa v cykle, ak nie je, pokračuje sa za telom cyklu. **Minimálny počet opakovaní cyklu je 0.**

### Podmieneny cyklus s podmienkou na konci

Sú situácie, v ktorých potrebujeme niekoľkokrát zopakovať nejakú činnosť pričom dopredu nevieme koľkokrát a nie je v nich výhodné použiť cyklus *while*, napr. obedujete dukátové buchtičky, môžete ich zjesť koľko Vám hrdlo ráči. Najprv musíte niečo zjesť, aby ste mohli posúdiť, či chcete jesť ešte ďalej alebo ste dojedli. Pre takéto situácie má programovací jazyk Pascal pripravený cyklus *repeat-until*.

Troška teórie:

### Opakuj činnosť, pokiaľ nie je splnená podmienka

*repeat* telo\_cyklu *until* výraz;

Fungovanie: Najprv sa vykoná telo cyklu (jeden príkaz alebo blok príkazov) a potom sa vyhodnotí pravdivosť logického výrazu. Ak je tento výraz pravdivý, pokračuje sa v cykle, ak je nepravdivý, pokračuje sa za telom cyklu. **Minimálny počet opakovaní cyklu je 1.**

## 28. Cyklus – riadiaca premenná cyklu

## 29. Cykly – porovnanie cyklov

Textový súbor – zápis údajov do nového súboru

*Vysvetlite postupnosť činností počítača a im odpovedajúce príkazy v jazyku C pri zápise údajov do textového súboru.* Ak potrebujeme do počítačového programu dodať zopár údajov, zvládneme to hravo aj z klávesnice. Veľmi často však vytvárame programy za účelom spracovania veľkého množstva informácií a zadávať ich vždy do programu ručne by bolo nemysliteľné.

**Súbor** je pomenovaná skupina údajov. **Textový** súbor pozostáva iba zo znakov. Ak potrebujeme v programe spracúvať väčšie množstvo údajov, je výhodnejšie, presnejšie a rýchlejšie načítať ich z textového súboru. Niekedy je zasa výhodné vypisovať údaje nielen na obrazovku, ale aj do súboru.

**Keďže súbor je uložený na vonkajšom pamäťovom médiu a nie v operačnej pamäti, ako bežiaci program, musíme jeho obsah sprístupniť programu. To sa deje cez vyrovnávaciu pamäť.**

**Po otvorení súboru z disku pre čítanie**, sa do vyrovnávacej pamäte načíta obsah súboru. Ten sa potom operáciami vstupu načítava do jednotlivých premenných. **Aj súbor otvorený pre čítanie je nutné pred ukončením činnosti programu zatvoriť**, inak by nám tento nedostatok mohol neskôr spôsobiť zamrznutie operačného systému.

**Po otvorení súboru pre zápis** je vyrovnávacia pamäť prázdna. Operáciami výstupu do súboru sa zapisujú údaje do vyrovnávacej pamäte. **Obsahom vyrovnávacej pamäte sa prepíše obsah súboru, prípadne sa vytvorí nový súbor, ak takýto súbor neexistuje, až keď v programe súbor zatvoríme!**

Príznak konca súboru - **EOF**. **Ak sme ešte nedošli na koniec súboru má tento parameter hodnotu 0**, v opačnom prípade má hodnotu 1. *EOF je skratka slov end of file.*

## 30. Jednorozmerné pole – triedenie

**TRIEDENIE VÝMENOU (preublávaním)** V usporiadanom poli vieme rýchlejšie nájsť hľadanú položku. Preto je dôležité mať položky v poli usporiadané, teda zotriedené. Triediť môžeme rôznymi spôsobmi. Každý spôsob má svoje výhody i nevýhody. Princíp triedenia výmenou (preublávaním) spočíva v cyklickom porovnávaní dvoch susedných položiek. Ak potrebujeme pole zotriediť vzostupne a ak prvé porovnávané číslo je väčšie ako druhé, musíme

čísla vymeniť. Ak prejdeme celým poľom toľkokrát, koľko je položiek v poli, pole bude utriedené. Malé čísla ako bublinky sa budú hromadiť na jednej strane a veľké čísla na druhej.

Toto triedenie je časovo veľmi náročné a neefektívne, ale zato je principiálne najjednoduchšie.

V jazyku C by sme zoradzovali údaje v poli a takto:

```
for (j=0; j<n; j++)
    for (i=0; i<n-1; i++)
        {if (a[i]>a[i+1])
            {p=a[i];
             a[i]=a[i+1];
             a[i+1]=p;
            } }
}
```

**TRIEDENIE VÝBEROM** Princíp triedenia výberom je nasledovný: na začiatku je celé pole nezotriedené. Prvú položku poľa v nezotriedenej časti postupne porovnávame s ostatnými položkami v tejto časti a ak nájdeme položku menšiu ako je prvá položka, tak ich navzájom vymeníme. Po porovnaní prvej položky so všetkými ostatnými je na prvej pozícii najmenšia položka, preto túto položku zaradíme do zotriedenej časti poľa a nezotriedenú časť poľa zmenšíme o prvú položku. Celý proces opakujeme dovtedy, kým nebudú všetky položky poľa patriť do zotriedenej časti a teda nezotriedená časť poľa bude prázdna. Toto triedenie nie je časovo veľmi náročné a principiálne je pomerne jednoduché.

V jazyku C by sme zoradzovali údaje v poli a takto:

```
for (j=0; j<n-1; j++)
    for (i=j+1; i<n; i++)
        {if (a[i]<a[j])
            {p=a[i];
             a[i]=a[j];
             a[j]=p;
            }
        }
}
```

**TRIEDENIE VSÚVANÍM** Princíp triedenia vsúvaním je nasledovný: na začiatku rozdelíme pole na dve časti do prvej – zotriedenej bude patriť prvá položka, ostatné položky budú patriť do nezotriedenej časti. Zoberieme prvú položku z nezotriedenej časti, uložíme ju do pomocnej premennej a hľadáme pre ňu miesto v zotriedenej časti, tak aby táto časť bola po zaradení tejto položky i naďalej usporiadaná. Po nájdení miesta vyposúvame položky za týmto miestom tak, že na miesto prvej položky v nezotriedenej časti uložíme poslednú položku zo zotriedenej časti, na jej miesto príde predchádzajúca a tento proces robíme dovtedy, kým nedôjdeme k miestu, kde treba uložiť položku z pomocnej premennej, nakoniec ju na uvoľnené miesto v poli uložíme. Zväčšíme počet položiek usporiadanej časti poľa o jedna a počet položiek v neusporiadanej časti zmenšíme o jedna. Celý proces opakujeme, kým nebudú všetky položky poľa patriť do zotriedenej časti a teda nezotriedená časť poľa bude prázdna. Toto triedenie nie je časovo veľmi náročné a principiálne je oproti predošlému náročnejšie.

V jazyku C by sme zoradzovali údaje v poli a takto:

```

for (k=1;k<n;k++)
    {//vyberieme prvý prvok z neusporiadanej casti, ktory treba
zaradit
        x=a[k];
        i=0;
        //hľadame miesto pre tento prvok
        while(x>a[i]) i++;
        j=k-1;
        //posuvame prvky tak, aby sme vytvorili miesto pre prvok
        while (j>=i)
            {a[j+1]=a[j];
              j--;
            }
        //zaradime prvok na spravne miesto
        a[i]=x;
    }

```

**REKURZÍVNE TRIEDENIE** Princíp rekurzívneho triedenia je nasledovný: Najprv budeme deliť pole na menšie úseky, kým sa úsek rozdeliť bude dať a potom úseky spojíme do jedného celku tak, aby údaje v spojených úsekoch boli usporiadané. Napríklad usporiadajme pole s prvkami 5, -3, 8, 0, -7. Delíme pole na dva úseky: 5, -3, 8 a 0, -7. Prvý úsek rozdelíme na dva úseky: 5, -3 a 8. Prvý úsek rozdelíme na úsek 5 a -3. Ďalej sa už prvý úsek rozkladať nedá. Ani druhý úsek sa už rozkladať nedá, tak ich zlúčime do pomocného poľa b, ktorého obsah potom prepíšeme na príslušné miesta v poli a. Po zlúčení bude obsah poľa a vyzeráť takto: -3, 5, 8, 0, -7. Úsek s číslom 8 sa ďalej rozkladať nedá, preto ho zlúčime s usporiadaným úsekom -3, 5. Po zlúčení bude obsah poľa a vyzeráť takto: -3, 5, 8, 0, -7. Keďže prvé tri položky poľa a sú usporiadané, pokračujeme v delení druhého úseku: 0, -7. Rozdelíme ho na úsek 0 a -7. Keďže obidva úseky sa už ďalej rozkladať nedajú, zlúčime ich. Po zlúčení vyzerá obsah poľa a takto: -3, 5, 8, -7, 0. Nakoniec zlúčime dokopy úseky -3, 5, 8 a -7, 0 takto: porovnáme prvé položky z oboch úsekov a do poľa b zapíšeme tú menšiu teda -7, potom porovnáваме -3 s 0, do poľa b sa na druhé miesto dostane -3, ďalej porovnáваме 5 s 0, do poľa b sa na tretiu pozíciu dostane 0. Keďže sme už vyčerpali všetky položky z druhého úseku dopíšeme do poľa b zvyšné položky z prvej časti, teda 5, 8. Po prepise obsahu poľa b do poľa a, vyzerá jeho obsah takto: -7, -3, 0, 5, 8 a je usporiadaný! Toto triedenie je časovo najoptimálnejšie, ale principiálne je pomerne zložité.

V jazyku C by sme zoradzovali údaje v poli a takto:

V hlavnom programe by sme volali procedúru tried **tried(1,n)**; Procedúra tried vyzerá takto:

```

void tried(int m,int n)

{int s;

if (m<n)
    {//delenie pola na dva useky
      s=(n+m)/2;

```



```

    tried(m,s);
    tried(s+1,n);
    zluc(m,s,n);    //zlucenie usekov do usporiadaneho celku
}}

```

Najzložitejšou je procedúra zluc, ktorá vyzerá takto:

```

void zluc(int m,int s,int n)
{float b[20];
 int j,k,l;

 j=m;
 k=s+1;
 l=m;

 //zapis mensich prvkov do pola b z jednotlivych usekov
 while ((j<=s) && (k<=n))
     {if(a[j]<=a[k])
        {b[l]=a[j];
         j++;
        }
     else {b[l]=a[k];
           k++;
        }
     l++;
 }
 //dopisanie prvkov z prveho useku do pola b
 while (j<=s)
     {b[l]=a[j];
      j++;
      l++;
     }
 //dopisanie prvkov z druheho useku do pola b
 while (k<=n)
     {b[l]=a[k];
      k++;
      l++;
     }
 //prepis upravovanej casti pola b do pola a
 for(j=m;j<n;j++) a[j]=b[j];
}

```

## Všeobecné informácie:

Otázky ku skúške ZI\_1 (ŠPZ, KAIA).

### Čo je to informatika?

Informatika je veda o systematickom spracovávaní, uchovávaní a šírení informácií, najmä o automatickom spracovávaní informácií pomocou číslicových počítačov.

### Čím sa principiálne odlišujú generácie počítačov?

Generácie sa odlišujú použitou technológiou spínacích obvodov. zmenšovanie hardware, zväčšovanie kapacity pamäte, rýchlosť vykonávania operácii

### Čo je to informácia?

Informácia je údaj, ktorý sme doteraz nepoznali

### Čo je to kódovanie?

Postup zobrazenia symbolov jednej abecedy do symbolov druhej abecedy.

### Čo je to ASCII-kód?

American Standard Code for Information Interchange – je najrozšírenejším 7-miestnym (bitovým) kódom pre prenos a spracovanie údajov

### Čo je to binárny kód?

dvojkový 0 a 1

### Prečo je binárny kód univerzálny?

je jednoduchý, jednoznačný, dá sa realizovať mechanicky, elektronicky, je ním možné určiť ľubovoľný symbol, ľubovoľnej abecedy záleží len na dĺžke kódu

### Prečo sa v počítačoch používa binárny kód?

z dôvodu spoľahlivosti práce počítača a technickej jednoduchosti jeho konštrukcie

### Čo je najmenšia jednotka informácie? Ako je definovaná?

Bit - Binary Digit - dvojková číslica, najmenšia základná jednotka informácie. Bit môže nadobudnúť iba jednu z hodnôt 0 alebo 1. 1B (Byte - bajt) = 8b (bitov)

### Koľko možností kóduje 5 bitov?

$2 \text{ na } 5 = 32$  možností

### Potrebujem zakódovať 25 rôznych možností. Koľko bitov minimálne potrebujem?

25 do dvojkovej sustavy

$25 = 11001 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$

a spocitat

teda 5 bitov

### Koľko bitov je 1 Byte?

8

**Koľko Bytov je 1 MByte?**

1MB=1 024 kB=1 048 576 B

1GB=1 024MB=1 073 741 824 B

**Prečo je ASCII-kód 8-bitový?**

Vznikol z medzinárodnej telegrafnej abecedy MTA-5, ktorá bola 7 bitová. Neskôr sa ASCII kód rozšíril na 8b pre možnosť kódovanie znakov rôznych národných abecied. V súčasnosti má teda 8 bitov, teda 8 miest. Možnosť zakódovať 256 znakov. Pričom prvá polovica znakov 128 je americká abeceda a druhých 128 znakov je vytvorených podľa jednotlivých národných zvyklostí. Keďže tých je viac museli vzniknúť znakové stránky. Napr. kód kamenských = 859, latin II = 852... -používa kódové stránky, a na tlačiarňach sa niekedy nahovno zobrazia a sú zlé znaky

**Vyjadrite v osmičkovej sústave číslo 567<sub>(10)</sub>.**

567:8=70 zvyšok 7

70:8=8 zvyšok 6

8:8=1 zvyšok 0

1:8=0 zvyšok 1

567 (10)= 1067 (8)

**Dobré materiály k C-čku sú na:**

<http://www.programovanie.kromsat.sk/prog-c/>

[https://spseke.sk/tutor/projekt/c\\_jazyk.htm](https://spseke.sk/tutor/projekt/c_jazyk.htm)