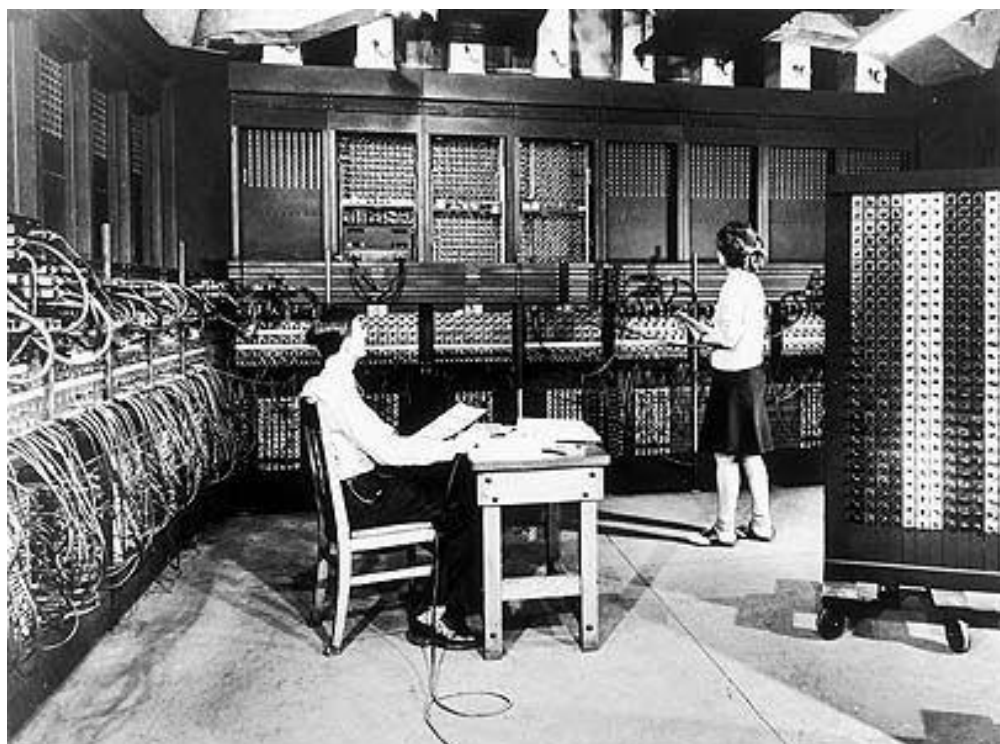
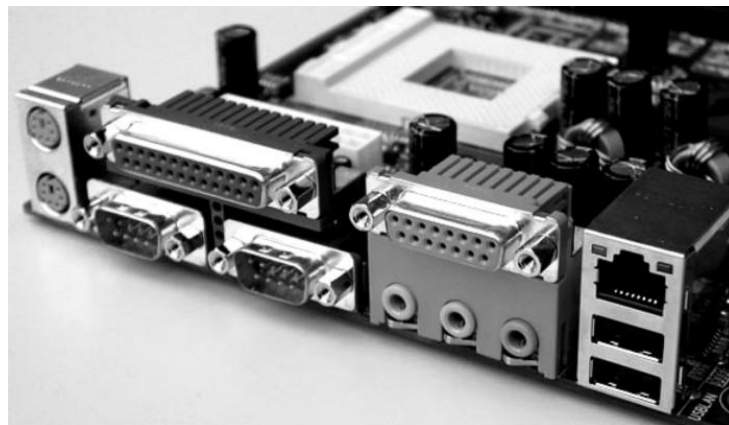


SOŠ a SOU Kuřim, s.r.o.

## ÚVOD DO VÝPOČETNÍ TECHNIKY



Ing. Vojtěch Novotný

**OBSAH:**

1.1.	ZÁKLADNÍ PRAVIDLA UČEBNY .....	3
<b>2.</b>	<b>ČÍSELNÉ SOUSTAVY .....</b>	<b>3</b>
2.1.	ČÍSELNÁ SOUSTAVA DESÍTKOVÁ (DEKADICKÁ) .....	3
2.2.	ČÍSELNÁ SOUSTAVA DVOJKOVÁ (BINÁRNÍ) .....	3
2.3.	ČÍSELNÁ SOUSTAVA ŠESTNÁCTKOVÁ (HEXADECIMÁLNÍ) .....	4
2.4.	ČÍSELNÁ SOUSTAVA OSMIČKOVÁ (OKTALOVÁ) .....	4
2.5.	PŘEVOD ČÍSEL Z DESÍTKOVÉ SOUSTAVY DO OSTATNÍCH .....	4
2.6.	PŘEVOD ČÍSEL Z OSTATNÍCH SOUSTAV DO DESÍTKOVÉ .....	5
2.7.	PŘEVODY MEZI DVOJKOVOU A ŠESTNÁCTKOVOU SOUSTAVOU .....	6
2.8.	PŘEVODY MEZI DVOJKOVOU A OSMIČKOVOU SOUSTAVOU .....	7
<b>3.</b>	<b>HARDWARE .....</b>	<b>8</b>
3.1.	SESTAVA PC .....	8
3.2.	ZÁKLADNÍ DESKA .....	9
3.3.	POČÍTAČOVÉ KONEKTORY .....	9
3.4.	MIKROPROCESORY .....	10
3.5.	OPERAČNÍ PAMĚŤ .....	11
3.6.	MAGNETICKÉ DISKY .....	11
3.6.1.	Pružné disky (floppy disk drive FDD) .....	11
3.6.2.	Pevný disk (harddisk drive HDD) .....	12
3.7.	OPTICKÉ DISKY .....	13
3.7.1.	Dělení CD Disků .....	13
3.7.2.	CD-ROM .....	14
3.7.3.	CD-R .....	14
3.7.4.	CD-RW .....	15
3.7.5.	DVD .....	15
3.8.	TISKÁRNY, DĚLENÍ TISKÁREN .....	16
3.8.1.	Jehličkové tiskárny .....	16
3.8.2.	Inkoustové tiskárny .....	17
3.8.3.	Laserové tiskárny .....	17
3.8.4.	Princip barevného tisku .....	18
3.9.	MONITORY .....	19
3.10.	GRAFICKÁ KARTA .....	20
3.11.	USB ( <i>UNIVERSAL SERIAL BUS</i> ) .....	21
3.12.	ULOŽENÍ INFORMACÍ V POČÍTAČI – TABULKA ASCII .....	21
3.13.	FYZICKÝ A LOGICKÝ DISK .....	22
<b>4.</b>	<b>SOFTWARE .....</b>	<b>24</b>
4.1.	ZÁKLADNÍ DĚLENÍ PROGRAMOVÉHO VYBAVENÍ .....	24
4.2.	OPERAČNÍ SYSTÉMY – ZÁKLADNÍ CHARAKTERISTIKY A FUNKCE .....	24
4.3.	OPERAČNÍ SYSTÉM MS-DOS .....	25
4.4.	STRUKTURA OS MS-DOS .....	25
4.5.	SOUBORY, HIERARCHICKÁ STRUKTURA ADRESÁŘŮ, ADRESÁŘOVÉ CESTY .....	26
4.6.	JEDNOZNAČNÁ SPECIFIKACE SOUBORU .....	26
4.7.	NÁHRADNÍ ZNAKY PRO OZNAČOVÁNÍ SOUBORŮ .....	27
4.8.	ZARÍZENÍ V OS MS-DOS .....	27
<b>5.</b>	<b>PŘÍKAZY MS-DOS .....</b>	<b>28</b>
5.1.	NEJDŮLEŽITĚJŠÍ PŘÍKAZY MS-DOS .....	28
<b>6.</b>	<b>PŘESMĚROVÁNÍ .....</b>	<b>31</b>
<b>7.</b>	<b>FILTRY .....</b>	<b>33</b>
7.1.	POUŽITÍ FILTRU MORE .....	33
7.2.	POUŽITÍ FILTRU FIND .....	33
7.3.	POUŽITÍ FILTRU SORT .....	34
<b>8.</b>	<b>DÁVKOVÉ SOUBORY .....</b>	<b>35</b>
8.1.	PŘÍKAZY PRO DÁVKOVÉ SOUBORY .....	36
8.2.	ZADÁNÍ PŘÍKLADŮ DÁVKOVÝCH SOUBORŮ: .....	40

# 1. Úvod



Tento dokument slouží pouze pro potřeby výuky výpočetní techniky na SOŠ a SOU Kuřim, s.r.o.

## 1.1. Základní pravidla učebny

- 1) Žáci jsou povinni dodržovat v hodinách všechna ustanovení školního řádu a provozního řádu učebny výpočetní techniky, s jehož obsahem budou na začátku školního roku seznámeni.
- 2) Žáci budou poučeni o pravidlech bezpečnosti práce na pracovištích výpočetní techniky.
- 3) Žáci mají zakázáno svévolně manipulovat se zařízením učebny.
- 4) Žáci se budou přihlašovat do sítě jen pod jménem zadáním učitelem
- 5) Žáci mají dovoleno kopírovat si své pracovní soubory na vlastní diskety a po výslovném povolení vyučujícího kopírovat soubory do školního počítače.

Všechny pracovní soubory se budou ukládat na pracovní disk P:, do svého adresáře, který bude pojmenovaný příjmením majitele. Tedy např.: **P:\Novak\**

# 2. Číselné soustavy

## 2.1. Číselná soustava desítková (dekadická)

- Při prvních početních „operacích“ si člověk vypomáhal prsty ruky. Jelikož má deset prstů, začal tak přirozeným vývojem, aniž by to tušil, počítat v **číselné soustavě desítkové**.
- Nejpoužívanějším číselným systémem dneška je arabský systém. Byl vyvinut Hindy přibližně ve 3. století před naším letopočtem. Součástí systému je nula, s jejíž pomocí je využívána množina hodnot rekurentním (opakujícím se) způsobem.
- Tuto číselnou soustavu člověk denně podvědomě používá, a proto ji nemusíme podrobněji rozebírat. **Obecné principy**, které při práci s ní využíváme (např. jednotkový přenos do vyšších řádů při sčítání čísel apod.), **platí i ve všech ostatních číselných soustavách**.
- Číselná soustava desítková ( **$z = 10$** ) (základ soustavy je číslo 10) má deset číslic: **0, 1, 2, 3, 4, 5, 6, 7, 8 a 9**. Pozor, číslo 10 není číslicí desítkové soustavy!
- Desítková soustava je **soustavou poziční**:  
Pro poziční číselnou soustavu platí, že každá číslice má svou **váhu** (záleží na pozici číslice). Všechny dále probírané soustavy jsou poziční. Číselná soustava je jednoznačně určena **základem soustavy, číslicemi soustavy a váhami jednotlivých pozic**. Základem číselné soustavy je celé kladné číslo větší než 1, základ označujeme  $Z$  (nebo  $z$ , to je jedno). Číslice jsou pak v intervalu od 0 do  $(z-1)$ .
- Př.: Zápis čísla v desítkové soustavě např.  $7428_{(10)}$ :

čísllice	7	4	2	8
váhy	$10^3$	$10^2$	$10^1$	10
váhy	1000	100	10	1
pozice	tisíce	sta	desítky	jednotky

## 2.2. Číselná soustava dvojková (binární)

- Pro elektronické prvky počítače je konstrukčně výhodné, když mohou rozlišovat **pouze dva stavy** a to, že daný prvek není/je pod napětím, či daným prvkem (obvodem) neprotéká/protéká elektrický proud. Těmto dvěma stavům jsou přiřazeny hodnoty **0** a **1**, což jsou všechny číslice dvojkové soustavy.
- **Čísla v počítači jsou proto zobrazena v číselné soustavě dvojkové**; podobně všechny ostatní znaky (obecně data) jsou zakódovány a zobrazeny jako **kombinace nul a jedniček**.
- Číselná soustava dvojková ( $z = 2$ ) má dvě číslice: **0** a **1**.
- **Váhy jednotlivých řádů** jsou rovny příslušným mocninám základu soustavy:  
 Naznačená pozice číslic: ... x x x x, x x ...  
 Mocnina základu soustavy: ...  $2^3$   $2^2$   $2^1$   $2^0$ ,  $2^{-1}$   $2^{-2}$  ...  
**Váha daného řádu:** ... **8 4 2 1**,  $\frac{1}{2}$   $\frac{1}{4}$  ...
- Př.: Zápís čísla ve dvojkové soustavě např.  $1011_{(2)}$ :

váhy	8	4	2	1
váhy	$2^3$	$2^2$	$2^1$	$2^0$
čísllice	1	0	1	1

### 2.3. Číselná soustava šestnáctková (hexadecimální)

- Kromě desítkové a dvojkové soustavy se můžeme ve výpočetní technice setkat ještě s dalšími číselnými soustavami. Jsou to **soustavy, jejímž základem je mocnina dvou**, u kterých se využívá zvláštního vztahu mezi nimi (viz dále). Z nich se nejčastěji používá soustava šestnáctková.
- Šestnáctková soustava má šestnáct číslic. Jelikož arabských číslic je pouze deset, musíme pro další šestnáctkové číslice použít jiné **znaky ve významu číselném**. Těmito znaky jsou písmena A až F, přičemž tato písmena odpovídají po řadě číslům 10, 11, 12, 13, 14 a 15.
- Šestnáctková soustava ( $z = 16$ ) má tedy tyto číslice: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (=10), B (=11), C (=12), D (=13), E (=14), F (=15)**
- Př.: Zápís čísla v šestnáctkové soustavě:  $3C2F_{(16)}$ :

váhy	4096	256	16	1
váhy	$16^3$	$16^2$	$16^1$	$16^0$
čísllice	3	C	2	F

### 2.4. Číselná soustava osmičková (oktalová)

- Osmičková soustava ( $z = 8$ ) má tyto číslice: **0, 1, 2, 3, 4, 5, 6, 7**
- Zápís čísla v osmičkové soustavě např.  $3026_{(8)}$ :

váhy	512	64	8	1
váhy	$8^3$	$8^2$	$8^1$	$8^0$
čísllice	3	0	2	6

- Pozor: číslo  $708_{(8)}$  nemůže být číslem osmičkové soustavy, protože 8 není číslicí osmičkové soustavy!

### 2.5. Převod čísel z desítkové soustavy do ostatních



DEFINICE: Převáděné číslo postupně, celočíselně dělíme základem soustavy do které číslo převádíme, tak dlouho, dokud nedostaneme nulový podíl (výsledek). Zbytek po posledním dělení pak je nejvyšším řádem hledaného čísla, zbytek po prvním dělení nejnižším řádem hledaného čísla.

Pozn.: **Celočíselné dělení:** vyjde celočíselný výsledek a zbytek, který je menší než základ soustavy do které převádíme. Např.: převádíme číslo  $158_{(10)}$  z desítkové soustavy do osmičkové. První celočíselné dělení bude  $158 : 8 = 19$ , zbytek 6.

Př.: Převeďte číslo  $274_{(10)}$  ze soustavy desítkové do soustavy dvojkové.

Řešení: **Dělení dvěmi Zbytek**

$274 : 2 = 137$	<b>0</b>	↑
$137 : 2 = 68$	<b>1</b>	↑
$38 : 2 = 34$	<b>0</b>	↑
$34 : 2 = 17$	<b>0</b>	↑
$17 : 2 = 8$	<b>1</b>	↑
$8 : 2 = 4$	<b>0</b>	↑
$4 : 2 = 2$	<b>0</b>	↑
$2 : 2 = 1$	<b>0</b>	↑
$1 : 2 = 0$	<b>1</b>	↑

Nakonec sepíší (ze spodu nahoru) všechny zbytky po dělení (zleva doprava). Mohu psát, že  $274_{(10)} = \underline{100010010}_{(2)}$  Zkouška:  $1 \cdot 2^8 + 1 \cdot 2^4 + 1 \cdot 2^1 = 256 + 16 + 2 = \underline{274}_{(10)}$

Př.: Převeďte číslo  $163_{(10)}$  ze soustavy desítkové do soustavy šestnáctkové.

Řešení: **Dělení šestnácti Zbytek**

$163 : 16 = 10$	<b>3</b>	↑
$10 : 16 = 0$	<b>A</b>	( $A_{(16)} = 10_{(10)}$ )

Pozn.: Je potřeba dávat si pozor, protože v šestnáctkové soustavě číslice 10 neexistuje, místo ní správně napíšeme číslici A. Výsledek je  $163_{(10)} = \underline{A3}_{(16)}$

Př.: Převeďte číslo  $163_{(10)}$  ze soustavy desítkové do soustavy osmičkové.

Řešení: **Dělení šestnácti Zbytek**

$163 : 16 = 20$	<b>3</b>	↑
$20 : 8 = 2$	<b>4</b>	↑
$2 : 8 = 0$	<b>2</b>	↑

Výsledek je  $163_{(10)} = \underline{243}_{(8)}$

## 2.6. Převod čísel z ostatních soustav do desítkové



DEFINICE: Provádí se rozkladem převáděného čísla na součet n násobku mocnin základu číselné soustavy a jeho vyhodnocení.

Př.:  $10111,11_{(2)}$  je číslo v soustavě dvojkové. Jaká je hodnota tohoto čísla v soustavě desítkové?

Řešení: Vyjdeme-li z obecného zápisu čísla, můžeme přímo psát:

váhy	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$
čísllice	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b> ,	<b>1</b>	<b>1</b>
	$1 \cdot 2^4$	$0 \cdot 2^3$	$1 \cdot 2^2$	$1 \cdot 2^1$	$1 \cdot 2^0$	$1 \cdot 2^{-1}$	$1 \cdot 2^{-2}$
hodnota	16	0	4	2	2	0,5	0,25

$10111,11_{(2)} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = \underline{(23,75)}_{10}$

nebo můžeme přímo sčítat váhy jednotlivých řádů (pamatujeme-li si je):

$16 + 4 + 2 + 1 + 0,5 + 0,25 = \underline{(23,75)}_{10}$

Pro efektivní převádění čísel z libovolné soustavy do soustavy desítkové je vhodné naučit se základní mocniny některých soustav. Uvedeny jsou v tabulce č. 1.

Tab. 1.: Důležité mocniny používané při převodu

dvojková soustava		osmičková	šestnáctková
$2^0 = 1$	$2^8 = 256$	$8^0 = 1$	$16^0 = 1$
$2^1 = 2$	$2^9 = 512$	$8^1 = 8$	$16^1 = 16$
$2^2 = 4$	$2^{10} = 1\ 024$	$8^2 = 64$	$16^2 = 256$
$2^3 = 8$	$2^{11} = 2\ 048$	$8^3 = 512$	$16^3 = 4\ 096$
$2^4 = 16$	$2^{12} = 4\ 096$	$8^4 = 4\ 096$	$16^4 = 65\ 536$
$2^5 = 32$	$2^{13} = 8\ 192$	$8^5 = 32\ 768$	
$2^6 = 64$	$2^{14} = 16\ 384$		
$2^7 = 128$	$2^{15} = 32\ 768$		

Další vzorové příklady pro převod čísel do desítkové soustavy:

Př.:  $100011101_{(2)} = 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 256 + 0 + 0 + 16 + 8 + 4 + 0 + 1 = \underline{285}_{(10)}$

Př.:  $217_{(8)} = 2 \cdot 8^2 + 1 \cdot 8^1 + 7 \cdot 8^0 = 128 + 8 + 7 = \underline{143}_{(10)}$

Př.:  $64_{(8)} = 6 \cdot 8^1 + 4 \cdot 8^0 = 48 + 4 = \underline{52}_{(10)}$

## 2.7. Převody mezi dvojkovou a šestnáctkovou soustavou

- Ve výpočetní technice je často využíván převod mezi dvojkovou a šestnáctkovou soustavou a naopak. Tento převod je možné provést pomocí mezipřevodu přes desítkovou soustavu. Je to však zbytečně zdlouhavý způsob, při kterém je zvýšené pravděpodobnost výskytu chyby. Naštěstí lze celý převod provést podstatně jednodušeji. Nejlépe si to vysvětlíme na příkladech:

Př.: Převeďte číslo  $10110011101_{(2)}$  ze soustavy dvojkové do soustavy šestnáctkové.

Řešení:

Převáděné číslo rozdělíme na čtveřice (odzadu)	1 0 1	1 0 0 1	1 1 0 1
Pod každou číslici dané čtveřice napíšeme jeho dvojkovou váhu	$2^3 2^2 2^1 2^0$	$2^3 2^2 2^1 2^0$	$2^3 2^2 2^1 2^0$
Sečteme všechna váhová čísla, nad kterými je 1	8 4 2 1	8 4 2 1	8 4 2 1
V našem případě je ještě potřeba převést 13 do šestnáctkové soustavy. Konečný výsledek je:	4+1=5	8+1=9	8+4+1=13
	5	9	D
	$10110011101_{(2)} = \underline{59D}_{(16)}$		

Pozn.: na čtveřice rozdělujeme dvojkové číslo proto, že 4 číslicemi dvojkové soustavy lze rozlišit 16 různých kombinací, tj. jednu číslici šestnáctkové soustavy

**Opačný převod** ze soustavy šestnáctkové do soustavy dvojkové je velice podobný

Př.: Převeďte číslo  $59D_{(16)}$  ze soustavy šestnáctkové do soustavy dvojkové

Řešení:

Převáděné číslo rozdělíme na jednotlivé číslice	5	9	D
Nyní napíšeme čtveřice dvojkových váhových čísel	$2^3 2^2 2^1 2^0$	$2^3 2^2 2^1 2^0$	$2^3 2^2 2^1 2^0$
Sčítáme váhová čísla postupně, abychom dostali požadovanou hodnotu nacházející se nad nimi. Každé číslo mohou použít maximálně jednou.	8 4 2 1	8 4 2 1	8 4 2 1
Pod každé číslo, které při součtu použijí napíší 1, pod to, které nepoužijí, napíší 0.	Potřebuji 5, použiji 4 a 1	Potřebuji 9, použiji 8 a 1	Potřebuji D, použiji 8 a 4 a 1
Nyní jen stačí sepsat výsledek předchozího kroku. Konečný výsledek (první nula se vynechává) je:	0 1 0 1	1 0 0 1	1 1 0 1
	$59D_{(16)} = \underline{10110011101}_{(2)}$		

## 2.8. Převody mezi dvojkovou a osmičkovou soustavou

- Tento převod je velice podobný převodu mezi dvojkovou a šestnáctkovou soustavou.

**Př.:** Převeďte číslo  $11001101_{(2)}$  ze soustavy dvojkové do soustavy osmičkové.

**Řešení:**

Převáděné číslo rozdělíme na <b>trojice</b> (odzadu)	1 1	0 0 1	1 0 1
Pod každou číslici dané čtveřice napíšeme jeho dvojkovou váhu	$2^2$ $2^1$ $2^0$	$2^2$ $2^1$ $2^0$	$2^2$ $2^1$ $2^0$
	4 2 1	4 2 1	4 2 1
Sečteme všechna váhová čísla, nad kterými je 1	$2 + 1 = 3$	$1 = 1$	$4 + 1 = 5$
Konečný výsledek je:	$11001101_{(2)} = \underline{\underline{315}}_{(8)}$		

**Pozn.:** na trojice rozdělujeme dvojkové číslo proto, že 3 číslicemi dvojkové soustavy lze rozlišit 8 různých kombinací, tj. jednu číslici osmičkové soustavy

### Opačný převod

**Př.:** Převeďte číslo  $315_{(8)}$  ze soustavy osmičkové do soustavy dvojkové

**Řešení:**

Převáděné číslo rozdělíme na jednotlivé číslice	3	1	5
Nyní napíšeme trojice dvojkových váhových čísel	$2^2$ $2^1$ $2^0$	$2^2$ $2^1$ $2^0$	$2^2$ $2^1$ $2^0$
	4 2 1	4 2 1	4 2 1
Sčítáme váhová čísla postupně, abychom dostali požadovanou hodnotu nacházející se nad nimi. Každé číslo mohou použít maximálně jednou.	Potřebuji 3, použiji 2 a 1	Potřebuji 1, použiji 1	Potřebuji 5, použiji 4 a 1
Pod každé číslo, které při součtu použiji napíši 1, pod to, které nepoužiji, napíši 0.	0 1 1	0 0 1	1 0 1
Sepíši výsledek předchozího kroku. Konečný výsledek tedy je:	$315_{(8)} = \underline{\underline{11001101}}_{(2)}$		

Tab. 2.: Zobrazení prvních 17 přirozených čísel v soustavách desítkové, dvojkové, osmičkové a šestnáctkové

desítková ( $X_{10}$ )	dvojková ( $X_2$ )	osmičková ( $X_8$ )	šestnáctková ( $X_{16}$ )
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

## 3. Hardware

**HARDWARE** – je technické vybavení počítače. Patří sem veškeré hmotné vybavení počítače. Laicky řečeno, je to všechno, co si můžeme na počítači „osahat“. Např. monitor, klávesnice, disketová mechanika, reproduktory,...

**SOFTWARE** – je programové vybavení. Je to nehmotné vybavení počítače, operační systémy, programy, které provádí úkoly zadané uživatelem a tím se snaží ulehčit komunikaci a práci uživatele s počítačem. Software je uložený na nosičích dat – disky.



Hardware by bez softwaru nemohl správně fungovat a naopak.



### 3.1. Sestava PC

V dnešní době existuje několik různých standardů pro počítače. Nejznámějším standardem je typ IBM PC (Personal Computer), o kterém se budeme učit. Kromě něj ovšem postupem času vznikaly a zanikaly i další standardy jako např. Apple, IQ 151, SGI,... Některé pořád existují, ale jsou využívány spíše jen pro specifické úkoly. Například jako servery nebo výkonné grafické stanice.

Základními částmi počítače typu PC jsou:

- **Základní jednotka** – jsou v ní obsaženy obvody a součásti nezbytné pro provoz počítače.

**Základní jednotka musí obsahovat:** základní desku, procesor, paměť RAM, grafický adaptér (grafickou kartu), mechaniky pružných disků (FDD), pevný disk (HDD), napájecí zdroj. Dále může obsahovat zvukovou kartu, mechaniku CD-ROM, DVD-ROM, interní faxmodem, zip drive, streamer, ....

- **Klávesnice** – standardní vstupní zařízení (data tudy do počítače „vstupují“)
- **Monitor** – standardní výstupní zařízení (data tudy z počítače „vystupují“)
- **Myš** – volitelné vstupní zařízení
- **Ostatní zařízení** – tiskárna, scanner, reproduktory, modem, čtečka karet, joystick, webkamera,... – volitelné výstupní, vstupní či vstup-výstupní zařízení

Počítače jsou konstruovány pro různé účely jako stolní nebo přenosné:

- **notebook** – přenosný počítač určený pro práci „v terénu“ neboli mimo kancelář
- **PDA** – (personal digital asistent) – organizér, počítač do dlaně
- **stolní počítač** – klasické domácí PC
- **server** – speciální výkonný počítač, většinou se speciálním softwarem, který má za úkol obsluhovat více uživatelů najednou.

Přenosné počítače jsou lehké a mají speciální zobrazovací jednotky s nízkou spotřebou energie, takže vydrží několik hodin pracovat na akumulátor. Naproti tomu jsou mnohem dražší.



Obr. 1.: Různé typy počítače PC. Notebook, stolní počítač, PDA.



### 3.2. Základní deska

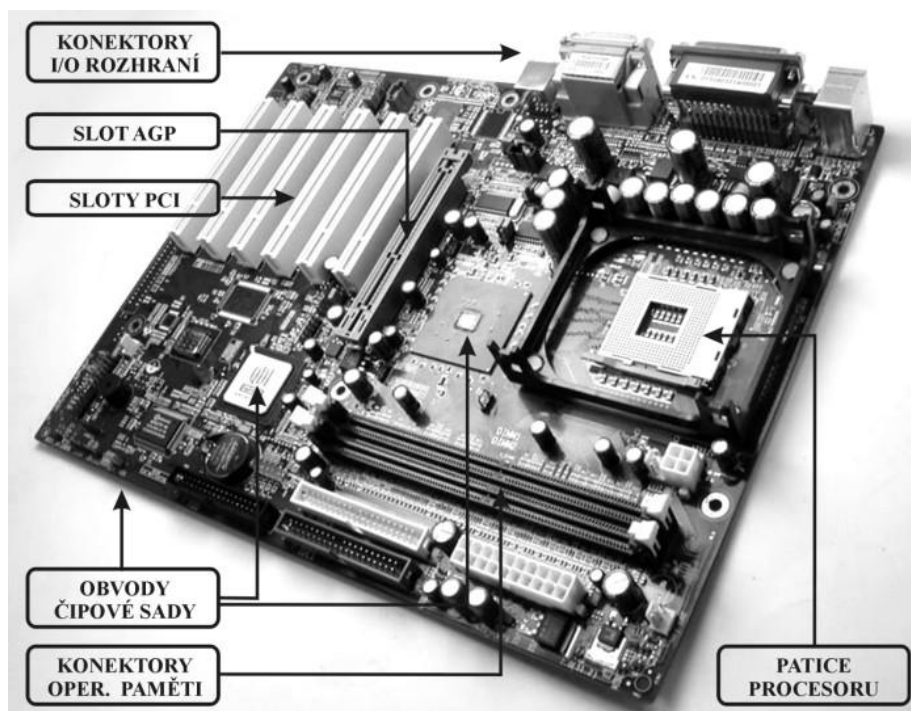
**Základní deska** – (motherboard) je nejdůležitější součást HW, určuje možnosti připojení komponent a jejich upgrade (inovaci). Obsahuje sloty pro přídavné karty, operační paměť RAM a procesor(y). Ty se osazují pro provoz důležitými elektronickými součástmi, jako jsou např. procesor, operační paměť, zvuková karta, grafická karta, modem.

Dále základní deska obsahuje řadič a obvody pro komunikaci s okolím a obvody čipové sady (Chipset). Chipset má dvě části North Bridge a South Bridge.

**North Bridge** – zprostředkovává komunikaci CPU s cache a operační paměti a napojuje tyto komponenty na sběrnice.

**South Bridge** – řídí sběrnice, obsahuje řadič pevného disku, obsahuje IO porty – USB, IrD, řídí správu energie.

Zařízení jako jsou procesor, řadič cache paměti, paměti a obvody čipové sady jsou společně propojeny pomocí sběrnic, které umožňují jejich rychlou vzájemnou komunikaci.



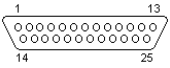







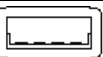






Obr. 2.: Základní deska

### 3.3. Počítačové konektory

Aby mohl počítač komunikovat s okolními zařízeními musí být vybaven konektory. Každé zařízení má jiné požadavky na způsob komunikace. Proto vznikly v průběhu vývoje počítačů různé, specifické typy konektorů. Zde je ukázka těch, se kterými se dnes můžete setkat:

Tab. 3.: Počítačové konektory

	<b>Napájecí konektor počítače</b>
	<b>Výstupní napájecí konektor.</b> Sem se obvykle připojuje monitor. U některých zdrojů se již tento způsob napájení nepoužívá.
	<b>Sériový port 25 pinů.</b> (COM port). Svého času býval nejrozšířenějším komunikačním konektorem. Jeho nevýhodou je, že dokáže data přenášet dnes již pomalou rychlostí 115kb/s. Na jeden port lze připojit jen jedno zařízení.

	<b>Sériový port (COM port)</b> 9 pinů. Menší verze sériového portu. Rozdíl je pouze ve velikosti.
	<b>Paralelní port.</b> (LPT port) Je určen pro připojování tiskáren, využívají ho také např. skenery a externí mechaniky ZIP. Dnes se využívá jen ojediněle.
	<b>Port PS/2.</b> Konektory určené výhradně pro připojení myši a klávesnice. (Port pro klávesnici bývá označován fialově, port pro myš zeleně.)
	<b>Konektor DIN.</b> Dnes se vyskytuje jen ojediněle. Používal se pro připojování klávesnice
	Konektor <b>D-Sub (VGA konektor)</b> . Slouží jako výstup pro připojení analogového monitoru.
	<b>USB A Host.</b> USB 1.1 / 2.0 Port USB. Více, v kapitole 3.11.
	<b>USB B Device.</b> Druhý konektor typu B je určen pro periferní zařízení.
	<b>Fire Wire port.</b> Někdy označován jako IEEE-1394 nebo iLink. Používá se zejména pro připojení kamer pro zpracování videa na PC, nebo externího HDD. Rozhraní je velice rychlé, přenos dat až 400Mb/s. Podporuje Plug'n'Play, teoreticky možnost připojit až 64 000 zařízení.
	<b>Stereo Sound In &amp; Out.</b> Výstupy a vstupy ze zvukové karty, pro připojení reproduktoru, sluchátek,...
	<b>TV výstup.</b> Slouží pro připojení televize.
	<b>Konektor RJ-11.</b> Konektor modemu. Slouží pro připojení do počítačové sítě
	<b>Konektor RJ-45.</b> Konektor síťové karty. Slouží pro připojení modemu k telefonní lince.

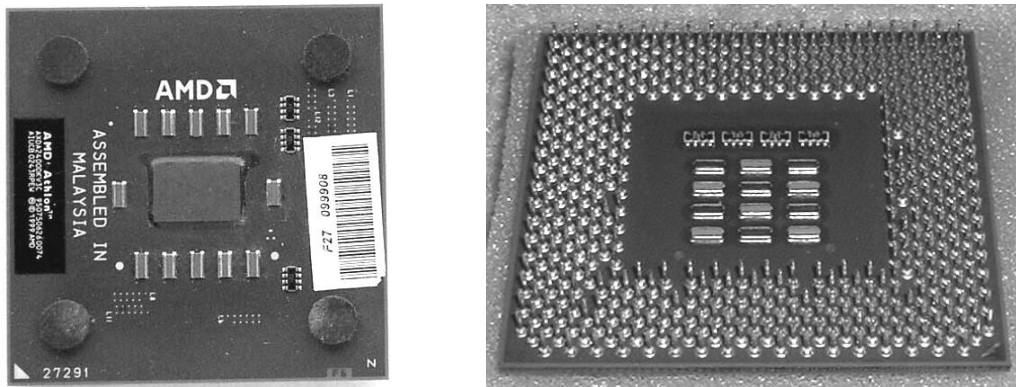
### 3.4. Mikroprocesory

**Mikroprocesor** – je základní elektronický obvod počítače, řídí činnost počítače a provádí výpočty. Nejvýznamnější výrobci mikroprocesorů na světě jsou **firmy Intel a AMD**, existují i další např. Motorola, VIA, Transmeta, IBM. U počítačů typu PC se používají procesory typů Intel 80x86 kompatibilní. Např.: od firmy Intel jsou to: 8086, 80286, 80386 SX, 80486 DX4, Pentium, Pentium MMX, Pentium III, Pentium 4, Core, Core2 Duo, Core Quad, Xeon,... Od firmy AMD: K5, K6, K6 3D, Duron, Athlon, Athlon XP, Athlon64, Athlon x2, ...

**Pracovní kmitočet procesoru:** je to frekvence, na které procesor pracuje [Hz]. Nastavuje se v Biosu nebo na základní desce přes jumper (drátové propojky) či přepínače. Nastavuje se jako frekvence sběrnice FSB x tzv. multiplikátor. Pracovní kmitočet (nepřímo) udává kolik operací je schopen procesor provést za jednotku času. Platí, že čím větší kmitočet procesor má tím je „rychlejší“. Celkový výkon procesoru ovšem ovlivňují ještě další věci, jako je velikost cache paměti, typ jádra, rychlost použité sběrnice FSB.



Samotný výkon počítače však nezávisí jen na mikroprocesoru a na jeho kmitočtu, ale i na operační paměti, grafickém akcelérátoru, HDD...



Obr. 3.: Procesor pohled shora a zdola.

### 3.5. Operační paměť

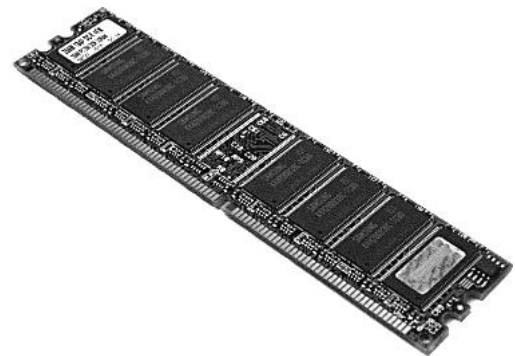
**Operační paměť** je realizována elektronickým obvodem nebo skupinou elektronických obvodů. Do operační paměti se po spuštění počítače načte operační systém, dále je určena k ukládání spuštěných programů a zpracovávaných dat (slouží tedy k dočasnému uchování informací). Je závislá na napájení a po vypnutí počítače se vymaže. Vlastní paměť se přidává do speciálních patič na základní desce.

Základní charakteristikou všech pamětí je **kapacita**. Udává velikost paměti. To je množství informací, které můžeme do operační paměti uložit. Jednotkou označení kapacity je 1 Byte nebo 1 bit.

Běžné hodnoty velikostí pamětí – 64 MB, 128 MB, 256 MB, 512 MB, 1024 MB, 2048 MB, 4096 MB

Tab. 4.: Jednotky velikosti informace

bit	b	bit může nabývat dvou stavů 0 a 1
bajt (Byte)	B	8 bitů; může nabývat $2^8=256$ stavů, je takové množství informací, v kterém lze uchovávat jeden znak
kilobajt	KB	$2^{10}=1\,024$ bajtů
megabajt	MB	$2^{20}=1\,048\,576$ bajtů
gigabajt	GB	$2^{30}=1\,073\,741\,824$ bajtů



Obr. 4.: Modul operační paměti DDR

### 3.6. Magnetické disky

Jsou disky pro trvalé uložení informací. Je-li informace uložena na magnetický disk, pak není závislá na napájení, po vypnutí počítače informace na magnetickém disku zůstává (na rozdíl od operační paměti). Čtení i zápis informace funguje na magnetickém principu, informace se ukládají na kotouč, na jehož povrchu je magnetická vrstva.

#### 3.6.1. Pružné disky (floppy disk drive FDD)

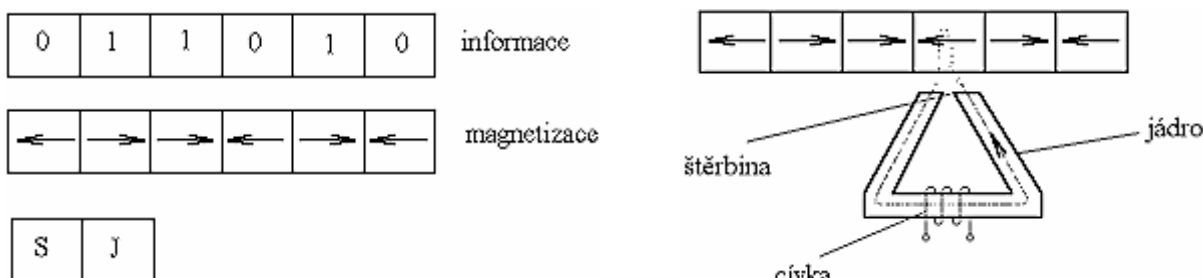
**Pružné disky** – jsou mechaniky pružných disků (**disketová mechanika**) a vlastní záznamová media (**diskety**). Slouží hlavně k přenosu dat, protože jsou bez větších problémů přenosné, výměnné. Základem diskety je kotouč z plastické hmoty, na kterém je nanášena magnetická vrstva.

Informace jsou zaznamenávány v soustředných kružnicích, kterým říkáme **stopy**. Každá stopa je rozdělena při formátování na určitý počet **sektorů** (1 sektor = velikost 512 B).

Diskety měly dříve různou kapacitu, dnes se již používají jen 3,5“ (3,5 palcové) která má kapacitu 1,44MB (dvě strany, 80 stop na stranu, 18 sektorů na stopu, každý sektor má 512B).

### Princip zápisu:

Diskety patří mezi magnetické disky. Pro zápis využívají principu magnetizace materiálu a pracují se dvěma stavy informace - 0, 1.



Obr. 5.: Princip zápisu na magnetický disk

Cívka, kterou prochází proud, vytváří magnetické pole, které je vedeno přes jádro zapisovací hlavy v magnetické vrstvě. V místě štěrbin prochází magnetické pole na materiál, **zmagnetizuje** ho. Směr magnetizace závisí na směru zápisového proudu.

### 3.6.2. Pevný disk (harddisk drive HDD)

**Pevný disk** – záznamové médium a mechanika tvoří jeden celek. Pevný disk je naproti pružným diskům zabudován přímo v počítači a slouží jako základní médium pro uchování dat a programů, pro instalaci systému i jako pracovní záznamové médium. Pevný disk je záznamové médium se stejnou organizací jak pružné disky.

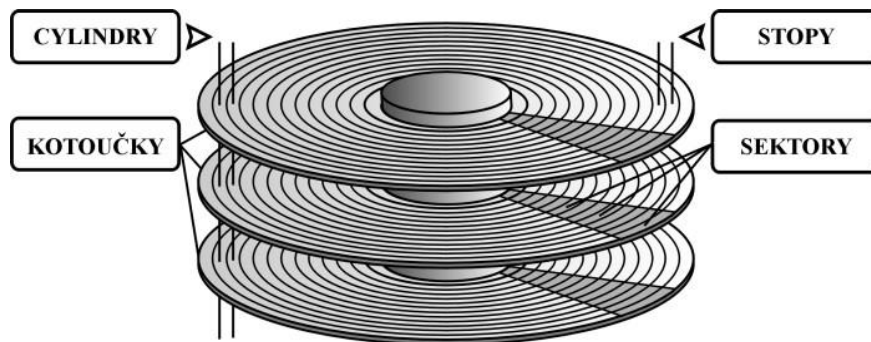
Na jednom vřetení je uloženo více magnetických kotoučků. Mezi jeho výhody patří daleko **větší kapacita** než pružný disk a podstatně **větší rychlost** práce s daty. Pevný disk tvoří neoddelitelnou část počítačové sestavy. Používá se pro uchování programů a dat, se kterými se denně v počítači pracuje.

### Základní pojmy:

- **Kapacita** – parametr stejný jako u operační paměti, udává kolik informací je schopen disk pojmout. Velikost (kapacita) dnešních disků je od jednotek GB až po stovky GB. U pevného disku (HDD) se obecně vypočítá: počet stran (povrchů) x počet stop (na každé straně) x počet sektorů (na každé stopě) x velikost sektoru.  
Pozn.: Počet sektorů na stopě nemusí být konstantní, vnitřní (kratší) stopy mají obvykle menší počet sektorů než vnější (delší) stopy.
- **Rychlost otáčení ploten** – udává, jak rychle dokáže motorek otáčet s plotnami disku. Tento parametr nejvíce vypovídá a rychlosti přenosu dat z disku. Dříve se vyráběly disky z 5400 ot/min, dnes převážně 7200 ot/min. Pro výkonné servery se používají disky s 10, 15 – 20 000 ot/min.
- **Diskové rozhraní** – dohoda mezi diskovým řadičem a pevným diskem, jak mezi sebou budou komunikovat. Ve starších počítačích nalezneme převážně rozhraní ATA (IDE) (rychlost přenosu dat 33 – 133 MB/s), v servech SCSI (rychlost až 320MB/s), v novějších počítačích

pak rozhraní SerialATA (SATA) (150MB/s) nebo SATA II (300 MB/s). Tyto rozhraní se navzájem liší hlavně kabeláží a přenosovou rychlostí.

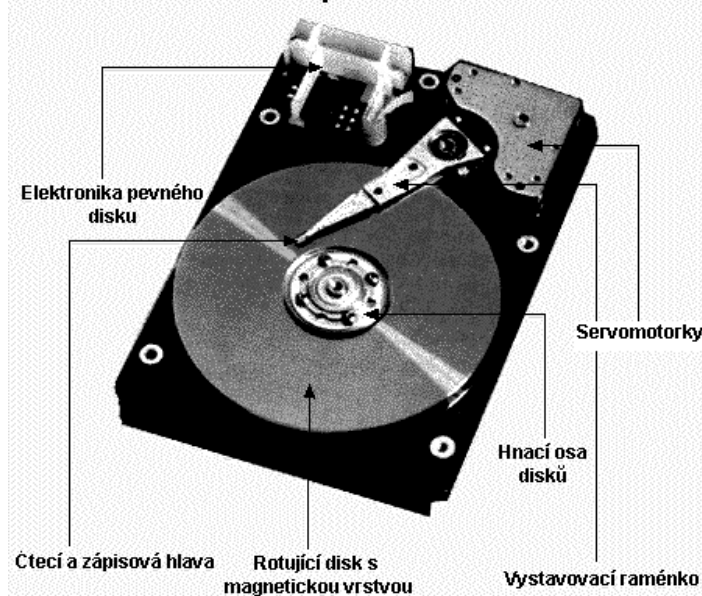
- **Přenosová rychlost** – rychlost komunikace mezi diskem a řadičem. Záleží především na použitém rozhraní a rychlosti otáčení. Přenosová rychlost se udává v Mb/s. (30 – 150 MB/s)
- **Velikost mezipaměti cache** – elektronická, rychlá paměť mezi diskem a řadičem. Slouží pro urychlení práce s malými, častěji se opakujícími objemy dat. Velikost 2 – 32 MB.
- **Přístupová doba** - doba, kterou je nutné čekat od zadání požadavku, než disk zpřístupní požadovanou informaci, typicky 5 – 30 ms.



Obr. 6.: Schéma základního naformátování harddisku.

Pozn.: **Cylindr** – je souhrn všech stop daného čísla na všech jednotlivých plotnách. (stopy pod sebou)

### Součásti pevného disku



Obr. 7.: Otevřené pouzdro harddisku

## 3.7. Optické disky

### 3.7.1. Dělení CD Disků

„Klasické“ kompaktní disky CD můžeme podle jejich vlastností rozdělit do těchto tří skupin:

- **CD-ROM** (Read Only Memory) Paměti, z nichž lze pouze číst. Informace se na disk zapisuje u výrobce při výrobě média (obvykle lisováním).
- **CD-R** (Write Once – Read Mostly) Paměti, do nichž lze prvotní informaci nahrát u uživatele. Pak se disk chová jako pevná paměť (ROM).
- **CD-RW** (Read Write Memory) Optické paměti s možností smazání a opětovného záznamu.

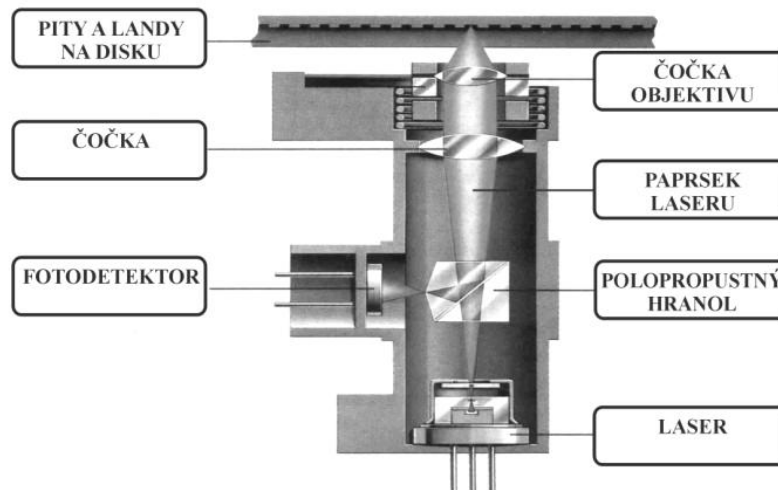
### 3.7.2. CD-ROM

Je plastový kulatý disk o průměru 8 cm (kapacita cca 200MB) nebo 12 cm (kapacita dříve 650 MB, nyní 700 až 870 MB. (74, 80, 99 minut))

**Princip záznamu:** Data v jedné spirálové stopě jsou ve formě malých prohlubní (**pity**), proloženými rovnými oblastmi nazývanými pole (**land** – ploška). Čtecí hlava (optické zařízení) s laserem a fotocitlivým prvkem (fotodetektem) (viz obr. 8) načítá pity a pole. Zařízení zaměřuje laserový paprsek přes polopropustné zrcadlo na povrch disku. Pole laserový paprsek odráží zpět, pity jej rozptylují. Zrcadlo přeměruje vracející se paprsek na fotodiodu, kde je snímáné světlo převáděno na elektrický signál. Klasická CD jsou vyráběna lisováním

**Záznamová stopa** má tvar **spirály**. Spirála začíná u středu média a rozvíjí se postupně až k jeho okraji. Záznam (spirála dat) je pouze na spodní straně disku, tj. záznam na CD-ROM disku je jednostranný. Délka celé spirály je zhruba 6 km a hustota dat je konstantní. Spirála je tvořena různě dlouhými prohlubněmi (pits) dlouhými 0,8  $\mu\text{m}$ .

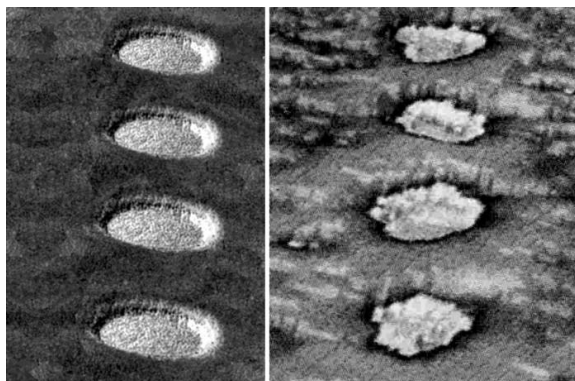
**Rychlosti čtení:** Základní rychlost přenosu je odvozována od hodnoty 150 kB/s a jejich násobků. Rychlost CD-ROM se pak udává jako násobek této rychlosti. Takže např. CD-ROM 40x má přenosovou rychlost 6000 kB/s (40 x 150kB/s), zatímco 8x CD-ROM má přenosovou rychlost pouze 1200 kB/s (8 x 150 kB/s). Této rychlosti lze ovšem dosáhnout jen na okraji disku.



Obr. 8.: Princip optické snímací hlavy

### 3.7.3. CD-R

**Způsob záznamu:** Základem technologie pro **vypalování** je vrstva s barvivem (a speciální laser používaný ve vypalovačkách). Bylo vybráno barvivo, které se při ozáření specifickým laserem o určité intenzitě velmi rychle zahřeje a nevratně změní své chemické složení. (To, že lidé hovoří o vytváření CD-R jako o *vypalování*, je tedy v podstatě správné.) Výsledkem chemické změny barviva je, že ozářená plocha odráží méně světla než plocha, která nebyla laserem zasažena. CD-R je vytvořeno z ozářených (*pit*) a neozářených (*land*) plošek, podobně jako je klasické CD vytvořeno pomocí pitů a lands. Výsledkem je, že vypálené CD funguje ve většině případů správně i na obyčejných domácích CD přehrávačích. (Jakmile již je médium jednou fyzicky změněno působením tepelných a chemických jevů, je změna **trvalá a nevratná**.)



Obr. 9.: Porovnání pitů na lisovaném CD-ROM (vlevo) a vypálených pitů na CD-R (vpravo) při velkém zvětšení

### 3.7.4. CD-RW

U tohoto média je dána možnost **přepisu**. Jde o vratné procesy záznamu, ale mazat jde pouze celý disk najednou.

#### Způsoby záznamu

U CD-RW médií je vrstva barviva nahrazena speciální vrstvou umožňující fázové změny jejich vlastností. Tato vrstva je tvořena slitinou kovů, která může změnit svůj stav při dodání energie a může se také vrátit do původního stavu. Podobně, jako se voda může měnit na páru nebo led v závislosti na teplotě, existují chemické látky, které změní svůj stav pouze pokud jsou zahřáty nebo v nebo v případě jiných podmínek, a dokáží ve změněném stavu zůstat i o zániku inicializačních podmínek. Později mohou být vráceny do svého původního stavu dalším, odlišným procesem.

Materiál použitý v CD-RW discích má tu vlastnost, že po zahřátí na jistou teplotu a po svém zchladnutí zkrystalizuje. Pokud se materiál zahřeje na vyšší teplotu a poté je ochlazen, bude mít amorfni strukturu. Když je materiál v krystalické formě, odráží mnohem více světla, než když je amorfni; tedy krystalická podoba je *land* a amorfni stav působí jako *pit*. **Použitím laseru s dvěma různými výkony je možné měnit materiál z jednoho stavu do druhého, což umožňuje přepisovat disk.**

### 3.7.5. DVD

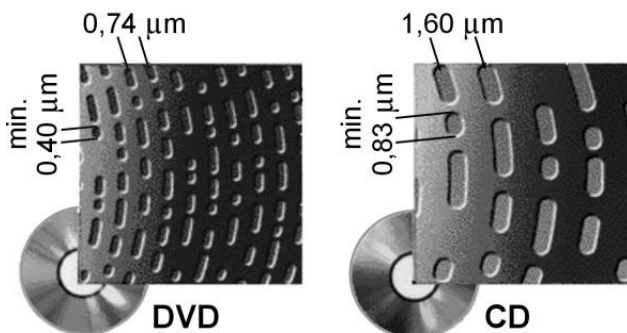
Jako **médium** se používá stříbrný disk o průměru 12 cm, tedy na první pohled je nerozlišitelný od CD. Obdobně jako u CD jsou data na disk zaznamenána ve formě spirálové stopy, sestávající z nepatrných prohlubní, a snímána laserovým paprskem.

DVD může být nahrán **jednostranně nebo oboustranně**, přičemž každá strana může být **jednovrstvá či dvojevrstvá**. Z toho vyplývá existence čtyř variant DVD:

- jednostranně nahané, jednovrstvé, s kapacitou 4,7 GB
- oboustranně nahané, jednovrstvé, s kapacitou 8,5 GB
- jednostranně nahané, dvojevrstvé, s kapacitou 9,4 GB
- oboustranně nahané, dvojevrstvé, s kapacitou 17 GB

Horní **informační vrstva disku** je polopropustná a umožňuje tak snímání horní i dolní odrazové vrstvy pomocí stejné laserové optiky, která je však vždy zaostřena na požadovanou vrstvu.

**Kapacita DVD** Zvýšená kapacita nespočívá výlučně na počtu vrstev nosiče, ale i na hustotě, se kterou mohou být informace uloženy. Je-li u obvyklých CD rozestup stop 1,6  $\mu\text{m}$ , u DVD je to pouhých 0,74  $\mu\text{m}$ . Stejně důležité je i zmenšení minimální délky prohlubně z 0,8  $\mu\text{m}$  u běžného CD na cca 0,4  $\mu\text{m}$  u DVD. Díky tomu je na jednom disku uloženo více dat.



Obr. 10.: Porovnání velikosti a hustoty pitů na disku DVD (vlevo) a na disku CD (vpravo)

**Požadavek kompatibility** směrem dolů (tj. s klasickými CD) znamená, že mechanika DVD musí umět číst oba typy médií. Problém snímání různých rozestupů stop byl vyřešen pomocí dvoučočkových systémů.

#### Varianty DVD:

- Stejně jako CD je klasické **DVD** pouze pro čtení
- **DVD-R** (recordable), které umožňují uživateli jednou zapsat data (stejně jako u CD-R)
- **DVD-RAM** (přepisovatelné), které dovolují uživateli libovolně číst, zapisovat a mazat data
- **DVD-RW** (přepisovatelné), ale mazat lze jen celý disk najednou (jako u CD-RW)

### 3.8. Tiskárny, dělení tiskáren

#### Tiskárny

Tiskárny slouží k trvalému zápisu informace na papír, plastovou fólii apod.

#### Rozdělení tiskáren

V současné době se u PC setkáváme nejčastěji s těmito **typy tiskáren**:

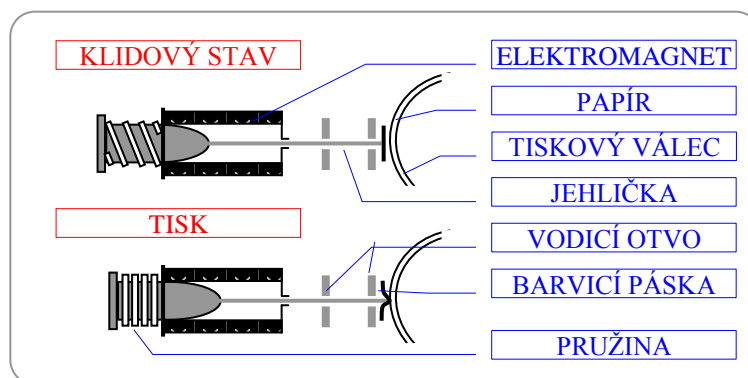
- jehličkové tiskárny,
- inkoustové tiskárny,
- laserové tiskárny,
- tepelné tiskárny – s počítači se běžně nepoužívají. Využívají se zejména jako tiskárny v supermarketech nebo např. pro tisk jízdenek v autobusu.

#### 3.8.1. Jehličkové tiskárny

Jehličkové tiskárny, které se též označují jako bodové, maticové či mozaikové, patří mezi úderové tiskárny.

Tyto tiskárny mají **v tiskové hlavě sloupec jehliček**. Znaky na papíře se vytváří sérií úderů těchto jehliček přes barvicí pásku, uloženou zpravidla v kazetě. Jednotlivé jehličky jsou ovládány elektromagneticky.

Nejčastěji se můžeme setkat s tiskárnami s **9 či 24** jehličkami, které mají vyšší kvalitu tisku.



Obr. 11.: Princip tisku jehličkové tiskárny



### ***Vlastnosti jehličkových tiskáren***

**Rychlost tisku** běžných jehličkových tiskáren je 50 až 150 zn./s, u kvalitnějších a dražších typů 200 až 300 zn./s. Současná technologická mez je 700 zn./s.

**Hustota tisku:** obvykle 200 až 300 DPI. **DPI** (dot per inch – bodů na palec) – označuje rozlišení tiskáren. Udává, kolik jednotlivých bodů je schopná tiskárna vytisknout na jeden palec (2,54 cm). Platí že čím více DPI, tím je tisk jemnější a lepší.

**Nevýhodou** úderových tiskáren je jejich velká hlučnost a nevalná kvalita tisku.

**Výhodou** těchto tiskáren je velmi levný provoz a např. možnost tisknout více kopií najednou přes „kopírák“.

### **3.8.2. Inkoustové tiskárny**

**Znak** u těchto tiskáren vzniká též z bodů, jen místo jehliček tiskací hlavy dopadají na papír kapky rychleschnoucího inkoustu, které jsou vystřelovány z jemných trysek v tiskací hlavě.

#### ***Vlastnosti inkoustových tiskáren***

**Kvalita tisku** je srovnatelná s laserovými tiskárnami a je – mimo jiné – závislá na kvalitě použitého papíru (má vliv na rozpíjení inkoustu). **Rozlišení** inkoustových tiskáren bývá zpravidla od 360 do 2400 DPI.

**Rychlost tisku** je řádově srovnatelná s jehličkovými tiskárnami, liší se však velmi model od modelu.

- rychlost černobílého tisku se pohybuje přibližně od 3,4 po 8,0 str./min,
- rychlost barevného tisku od 0,25 do 3,0 str./min.

Výhodou je téměř **bezhluchý provoz**, nízké pořizovací náklady a bezkonkurenčně kvalitní barevný tisk.

Největší **nevýhodou** jsou nejvyšší provozní náklady.

### ***Způsoby tisku inkoustových tiskáren***

#### **Technologie Bubble Jet**

- **Tryskové komůrky** tiskové hlavičky se plní inkoustem automaticky kapilárními silami.
- Má-li se tisknout, zapne se na krátkou dobu (asi 2  $\mu$ s) **topné tělísko**, které zahřeje inkoust na dně komůrky na teplotu zhruba 300 °C.
- Inkoust se vlivem této teploty odpaří a vzniklá **parní bublinka** vytlačí inkoust do těla trysky, kterou opouští ve formě malé kapičky o objemu přibližně 10 pl (picolitru) rychlostí přibližně 100 km/h a dopadá na list papíru.

#### **Technologie Piezo**

- K vystřelení kapičky inkoustu se používá piezoelektrický měnič (destička, která se po přivedení elektrického napětí deformuje – **prohne**).
- Deformací piezoelektrického měniče vznikají v kanálku s inkoustem tlakové vlny, které **vystřelují** kapičky inkoustu.
- Výhoda oproti Bubble Jet je ta, že elektrické napětí je přímo převedeno na mechanický pohyb  $\Rightarrow$  vyšší rychlost tisku.

### **3.8.3. Laserové tiskárny**

Tyto tiskárny využívají k tisku obdobné principy jako kopírky.

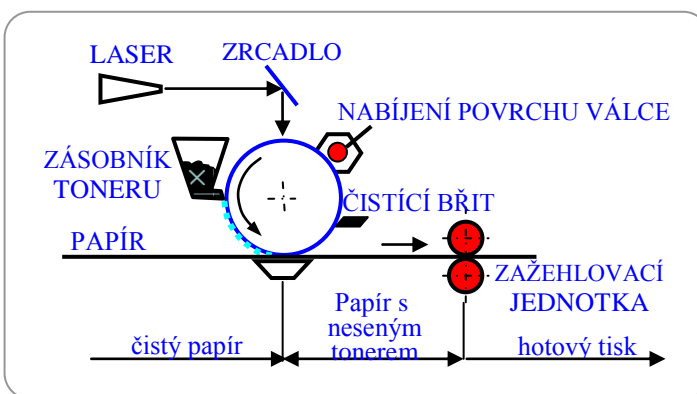
## Technologie tisku

Tiskárna využívá fotoelektrických vlastností polovodičů (selenu), který je nanesen na tiskacím (fotocitlivém) válci. Neosvětlený selen se chová jako izolátor  $\Rightarrow$  povrch fotocitlivého válce lze nabít elektrostatickým nábojem.

Fotocitlivý válec je tedy nejprve nabit. Na takto připravený povrch válce se pak laserovým paprskem postupně přenesou (překreslí) obraz, který má být vytištěn. Místa, která byla laserem osvětlena, se vybijí. V dalším kroku se na místa, která jsou vybita, přichytí barvicí prášek – toner.

Z válce se toner přenesou na papír, na kterém je tepelně fixován průchodem papíru přes vyhřívané válce.

V poslední fázi tisku se mechanicky odstraní zbytky toneru z válce a válec se před dalším nabíjením osvítlí.



Obr. 12.: Princip tisku laserové tiskárny

## Vlastnosti laserových tiskáren

- Vysoká **kvalita tisku**; rozlišení bývá obvykle 600 DPI, u kvalitnějších tiskáren 1200 DPI.
- Vysoká **rychlost tisku**; u „osobních tiskáren“ zpravidla 6/8/12 str./min, u dražších a výkonnějších „síťových“ modelů 17 až 40 str./min.
- Téměř **bezhluchý provoz**.
- **Vyšší pořizovací cena**, ale **lacinější provoz**, než u inkoustových tiskáren.

Tab. 5.: Porovnání základních vlastností jednotlivých typů tiskáren

	Pořizovací cena	Cena za stránku	Rychlost tisku	Kvalita tisku
Jehličková tiskárna	4 000 a více	Velmi nízká	1 – 4 stránky za minutu	Špatná
Inkoustová tiskárna (černobíle)	2 000 a více	80 – 100 hal	1 – 12 stránek za minutu	S dobrým papírem vynikající
Inkoustová tiskárna (barevně)		1 – 15 Kč	0,5 – 2 stránky za minutu	
Laserová tiskárna (černobíle)	4 000 a více	10 – 40 hal	4 – 40 stránek za minutu	Vynikající
Laserová tiskárna (barevně)	20 000 a více	50 – 100 hal	4 – 24 stránek za minutu	Dobrá

### 3.8.4. Princip barevného tisku

- Tiskárny tisknou zpravidla na bílý papír. Proto skládáním barev potřebujeme získat barvu černou. Takovým mechanismem je model CMY.
- **Model CMY** obsahují tři základní barvy **Cyan-Magenta-Yellow** ~ azurová-purpurová-žlutá.  
**Platí:** Azurová + purpurová + modrá = černá. To platí teoreticky. V praxi však většinou získáme „špinavě“ šedohnědou barvu. Proto se u tiskáren používá model CMYK.
- **Model CMYK** Cyan-Magenta-Yellow-Key ~ azurová-purpurová-žlutá-černá: K uvedeným barvám je přidána ještě barva černá, kterou se tiskne černý text. Ostatní barvy se skládají z barev CMY.

- **Př. :** Purpurová + žlutá ~ červená  
Azurová + žlutá ~ zelená  
Azurová + purpurová ~ modrá

### **Konkrétní realizace barevného tisku**

- **Jehličkové tiskárny:** čtyřbarevná barvicí páska – není však schopna tisknout různé odstíny
- **Laserové tiskárny:** čtyři zásobníky toneru – použitelnost pro fotografie je špatná, protože špatně zvládá tisk jednotlivých odstínů. Hodí se zejména pro grafiku.
- **Inkoustové tiskárny:** čtyři nádržky na inkoust. Kvalita barevného tisku velmi vysoká. Zaleží ovšem hodně na použitém papíru.

## **3.9. Monitory**

### **Parametry monitorů:**

- **Snímková frekvence** – Počet snímků, který je schopen monitor vykreslit za sekundu, nazýváme **snímková frekvence**. Běžná snímková frekvence displejů se pohybuje od 70 do 150 Hz, u kvalitních displejů dosahuje až 130 Hz. Ergonomická obnovovací frekvence je 72 Hz. To je hodnota, při které již lidské oko nevnímá „blikání“ monitoru a je tím pádem méně unavováno.
- **Rozlišení** – (velikost obrazu) se pohybuje od 320 x 200 bodů (tj. 25 řádků po 40 znacích), až po rozlišení 2096 x 2096 bodů u nejlepších displejů.
- **Rozteč bodů** – Ta nám říká, jak je každý bod velký (či spíše úzký). Čím užší, tím lépe (znaky jsou ostřejší). V dnešní době výrobce nabízejí monitory s roztečí bodů 0,28 mm a lepší (běžně již 0,25 mm, dobré grafické monitory mívají rozteč bodu i 0,23 nebo 0,21 mm). Větší monitory mají obecně větší bodovou rozteč, i když jsou velmi kvalitní. Obraz je ostřejší, když vzdálenost bodů je menší.
- **Úhlopříčka (velikost monitoru)** – je udávána v palcích úhlopříčky: **15“, 17“, 19“, 20“, 21“**. Jeden palec = 2,54 cm. Většinou platí, že čím větší úhlopříčka, tím větší je maximální rozlišení.
- **Bezpečnost vyzařování** – Postupem času byly definovány normy pro povolené normy vyzařování. Každá další norma je přísnější a tedy pro oči méně zatěžující. Chronologicky vznikly tyto normy: MPR I, MPR II, TCO 92, TCO 95, TCO 99, TCO 02, TCO 04.



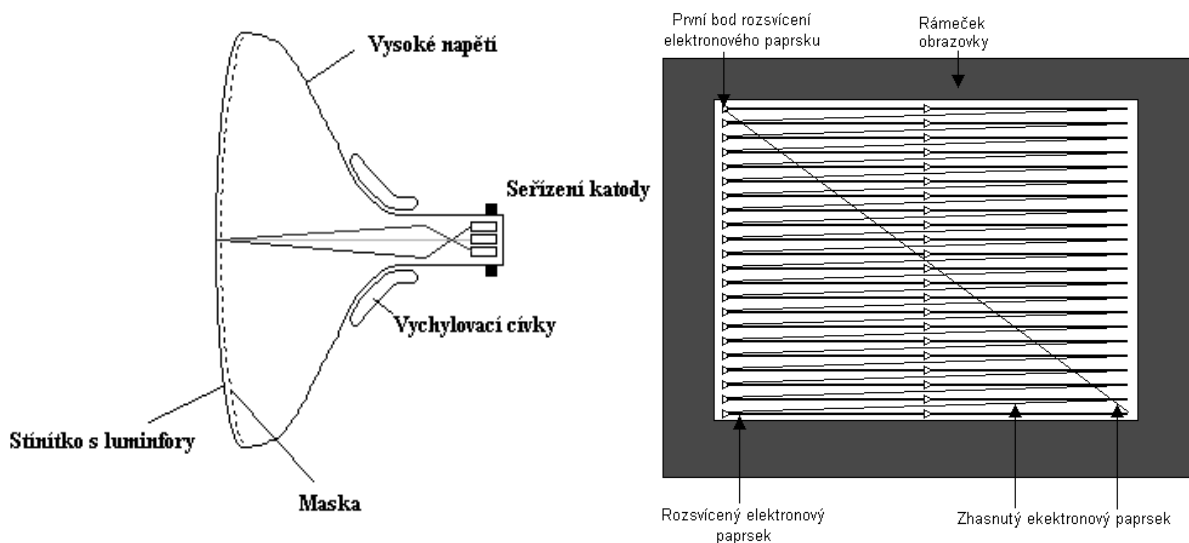
Obr. 13.: CRT monitor



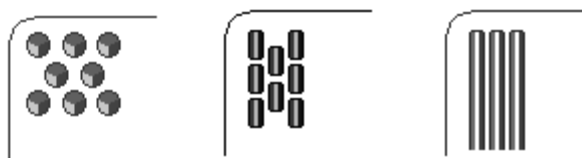
Obr. 14.: TFT display

## Princip CRT monitoru:

Při práci barevné obrazovky jsou ze tří rozžhavených katod (viz obr. 15.) emitovány elektronové svazky, které jsou pomocí jednotlivých mřížek taženy až na stínítko obrazovky. Na zadní stěně stínítka obrazovky jsou naneseny vrstvy tzv. **luminoforů** (luminofor = látka přeměňující kinetickou energii na energii světelnou). Tyto luminofory jsou ve třech základních barvách - Red (červená), Green (zelená), Blue (modrá) - pro aditivní **RGB** model skládání barev. Vlastní elektronové svazky jsou bezbarvé, ale po dopadu na příslušné luminofory dojde k **rozsvícení** bodu odpovídající barvy. Kombinováním jejich intenzity lze vytvořit různé barevné odstíny. Pro přesný výběr luminoforu se v rastrových displejích používají stínící masky (viz obr. 16.). Každý bod na luminiscenční vrstvě stínítka tedy emituje malé množství záření, jehož intenzita rychle slábne. Proto se musí obraz rychlým směřováním elektronového paprsku opakovaně vykreslovat.



Obr. 15.: Princip vykreslování na CRT monitoru (pohled shora a zepředu)



Obr. 16.: Různé typy masek: Delta, Inline, Trinitron

### 3.10. Grafická karta

**Grafická karta** (grafický adaptér) – je souhrnné označení pro sadu integrovaných obvodů, které mají na starost zobrazení obrazových dat na monitoru. Srdcem grafické karty je grafický čip. Tento čip se stará o vykonání příkazů, které přicházejí z mikroprocesoru. Následně předá hotová data do převodníku, a ten je pošle na výstup grafické karty, kde je připojen kabel, vedoucí k monitoru. Na monitoru se zobrazí výsledný obraz.

**Základní parametry grafické karty jsou**

- **Velikost grafické paměti** (4 – 1024 MB).
- **Typ a frekvence grafické paměti** – vypovídá o rychlosti a provedení (VRAM, SDRAM, DDR, SDR...)
- **Typ a frekvence grafického procesoru.**
- **Použití rozhraní** (AGP, AGP 8x, PCIe...).

- **Rychlost** – počet pixelů (bodů), které videokarta dokáže vykreslit za jednotku času.
- **Barevná hloubka** – určuje maximální počet současně zobrazitelných barev. Typická 24-bitová barva RGB (tzn. 8-bit na barvu) je označována jako True Color (Pravé barvy), jedná se o fotorealistické zobrazení barev, ve kterém lze současně zobrazit až 16,7 miliónů barev.

### 3.11. USB (*Universal Serial Bus*)

USB rozhraní se během posledních let stalo zcela běžnou součástí spotřební elektroniky připojitelné k počítači a již téměř vytlačilo klasický sériový port a postupně i paralelní port.

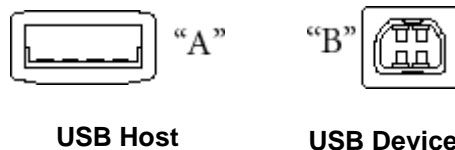
#### Základní parametry rozhraní USB :

- Komunikační rychlost USB 1.1 max 1.5MB/s (12Mb/s), USB 2.0 max 60MB/s (480Mb/s)
- Komunikační vzdálenost do 5 m
- Max. 127 fyzických zařízení připojitelných současně do hvězdicové topologie
- USB zajišťuje správné přidělení prostředků (IRQ, DMA, ...).
- Programové vybavení pro ovládání rozhraní USB odpovídá plně standardu Plug & Play
- Dynamické přidávání a ubírání zařízení při jejich připojení/odpojení
- Autoidentifikace periférií, automatické mapování funkce a konfigurace ovladače (driveru)
- Dynamicky připojitelné a překonfigurovatelné periférie
- Připojení periférií požadujících přenosové pásmo od několika kb/s do několika Mb/s
- Distribuce napájení 5V pro zařízení s malým odběrem, jako je myš či klávesnice



USB rozhraní používá dva typy konektorů. Plochý konektor „typ A“ je obsažen na prakticky každém PC v minimálně 2 konektorech (některé základní desky mají integrován rovnou USB hub, který obsahuje až 8 portů).

Druhý konektor „typ B“ je určen pro periferní zařízení, čímž je zároveň definován standard propojovacího kabelu.



### 3.12. Uložení informací v počítači – tabulka ASCII

Znaky se v počítači často ukládají tak, že jeden znak zabírá jeden bajt. Bajt je 8 bitů a může být zaplněn  $2^8=256$  způsoby. Jaký znak odpovídá určitému rozložení nul a jedniček v bajtu určuje ASCII tabulka (*American Standard Code for Information Interchange*). Například  $0011111_2=63_{10}$  je znak **?** a  $01100001_2=97_{10}$  je znak **a**.

Prvních 32 znaků (od 0 do 31) jsou vyhrazeny pro **řídící kódy** a představují proto tzv. **netisknutelné znaky**. Znaky s kódy 32 až 127 jsou velká a malá anglická abeceda, číslice, běžná interpunkční a matematická znaménka. Znaky 128 až 255 patří do **rozšířené ASCII tabulky**, jsou to speciální grafické symboly, například pro psaní rámečků a znaky národních abeced. Př.:  $\perp \perp \perp \perp$   
 Ā ā Ļ ĺ Ž ž.

Tab. 6.: Standardní ASCII tabulka se 128 znaky

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Protože tento počet znaků nestačí pro všechny národní abecedy (neobsahuje například naše čárky a háčky), bývají tyto znaky programově měněny na znaky národní abecedy. Pro češtinu se používá hned několik systémů, což problém značně komplikuje. Jde zejména o češtinu v kódu Kamenických, kód Latin 2 a kódovou stránku CP1250 používanou ve Windows. Podle toho, v jaké češtině je text napsán, je třeba použít to správné kódování.

Při použití češtiny je třeba rozlišit dvě věci. Samotné instalování češtiny, kdy se na obrazovce správně zobrazují české znaky, a vypínání a zapínání české klávesnice. To se často v MS-DOS děje pomocí kláves CTRL + ALT + F1. K převodu mezi kódy lze použít některé textové editory nebo speciální programy.



**TIP:** Kombinací kláves levý ALT + číselný kód napsaný na numerické klávesnici se zapíše do souboru i znak, který momentálně není na klávesnici k dispozici. (Např.: ö-148, ü-129, \ - 92, @-64 a pod.)

Ve Windows je situace komplikována ještě různými fonty. (Pro češtinu jsou určeny fonty s příponou CE.) K orientaci a získání méně používaných znaků slouží aplikace Mapa znaků.

Textové editory postupně přecházejí na systém **Unicode**, kdy jeden znak zabírá 2 bajty. Potom máme k dispozici bez problémů  $256 \times 256 = 65\,536$  symbolů, což už je plně dostačující. Délka souborů je však dvojnásobná.

### 3.13. Fyzický a logický disk

**Fyzický disk** je vlastní technické zařízení počítače–hardware. Aby s diskem mohl operační systém pracovat (číst a zapisovat soubory), musí se disk pro použití v operačním systému naformátovat do příslušného **souborového systému**. Různé operační systémy pracují s různými souborovými systémy (FAT, FAT32, NTFS, Ext2, ...)

Před formátováním je třeba na disku vytvořit (jednu nebo více) oblast (oddíl, **partition**). Oblasti se dělí na **primární a rozšířené** (v rozšířené oblasti je třeba vytvořit logickou jednotku).

Pokud vytvoříme na jednom fyzickém disku více než jednu oblast, pak každá tato jednotka se chová jako samostatný disk a hovoříme o **logickém disku**. Jeden fyzický disk tedy může (ale nemusí) být **rozdělen na více disků logických**. Každý takový logický disk se pak „chová“ jako **samostatný disk**, lze ho např. formátovat, atd.

#### *Jména disků*

Každý logický disk má v OS MS-DOS své jméno, které se skládá z písmene (může psát malými i velkými písmeny) a dvojtečky (např. C:). Přitom písmena A a B jsou vyhrazena pro označování pružných disků, další písmena počínaje C slouží pro postupné označování ostatních disků.

Tab. 7.: Posloupnost přiřazování jmen pro značení disků

Označení disku	Význam
A:	Vždy první pružný disk
B:	Vždy druhý pružný disk (je-li instalován)
C:	Je první primární oblast (oddíl, partition) prvního pevného disku
D: až Z:	Je-li v počítači jeden fyzický disk, pak další primární oblast nebo logická jednotka v rozšířené oblasti Jsou-li dva fyzické disky, pak první primární oblast druhého disku Nakonec jednotka CD ROM nebo DVD ROM, USB disk nebo jiné zařízení

## 4. Software

### 4.1. Základní dělení programového vybavení

Programové vybavení počítačů PC můžeme rozdělit na dvě skupiny:

1. **Základní programové vybavení – OPERAČNÍ SYSTÉM**
  - operační systém se po zapnutí počítače automaticky ukládá do operační paměti  
**„Zajišťuje komunikaci s uživatelem a řídí činnost počítače.“**
2. **Uživatelské programové vybavení**
  - je to programové vybavení, které umožňuje uživateli řešit jeho problémy a úkoly (např. textové editory – práce s textem a tisk)

### 4.2. Operační systémy – základní charakteristiky a funkce



Je množina programů, které umožňují využívat technické a programové vybavení počítače, řídí jeho činnost, zajišťuje efektivní zpracování aplikačních programů. Operační systém ovládá řízení výpočetního systému - procesor, operační paměť, periférie atd.

Tab. 8.: Základní rozdělení operačních systémů:

Typ OS	Uživatelů	Multitasking	Přístupová práva	Příklad
Jednouživatelské a jednoúlohové	1	není	nejsou	MSDOS
Jednouživatelské víceúlohové	1 nebo více	ano	nejsou	WIN3.x, WIN95,98
Víceuživatelské (síťové)	Více (skupiny)	ano	ano	WIN NT, UNIX

#### Jednouživatelské a jednoúlohové O.S.

Nejjednodušší typ O.S. Předpokládá se práce jednoho uživatele na daném počítači. Neobsahuje tedy žádné prostředky pro zajištění ochrany souborů a disků před poškozením jiným uživatelem. V nejjednodušším případě neumožňuje běh více procesů (programů) a nedisponuje tedy ani prostředky pro sdílení systémových zdrojů mezi současně běžícími procesy nebo jsou tyto prostředky omezeny. Krach programu = krach celého operačního systému.

Př: jednoduché O.S. - MS-DOS

#### Jednouživatelské víceúlohové O.S.

Vyspělejší systémy umožňují tzv. **multitasking**. **Multitaskingem** v tomto smyslu rozumíme **možnost paralelního** (současného) **běhu** několika **procesů (programů)**. Musí se zajistit bezchybné sdílení systémových zdrojů jednotlivými procesy tak aby nedocházelo ke konfliktům. To se řeší postupným přidělováním potřebných zdrojů (např. času procesoru, paměti apod.) procesům. Jde tedy pouze o kvaziparalelní zpracování procesů – je jen jeden procesor a jeho čas musí být postupně rozdělován mezi procesy. Toto přidělování může probíhat buď v režii samotných procesů (po určité době sám předá řízení jinému) pak se jedná o **kooperativní** multitasking nebo zcela v režii O.S. což je **preemptivní** multitasking.

**Preemptivní** multitasking je méně zranitelný – havárie jednoho procesu neznamená ukončení práce celého systému. *(Havárii se rozumí, že v průběhu zpracování daného procesu dojde k přechodu na nesprávnou adresu v paměti. Obsah této adresy je interpretován jako instrukce (ačkoliv obsahuje něco úplně jiného) a její zpracování způsobí zcela nesmyslné chování aplikace – obvykle dojde k tzv. zatumnutí – aplikace přestane komunikovat).* Vzhledem k tomu, že u



preemptivního multitaskingu řídí přidělování času procesoru O.S., po určité době je řízení předáno jinému (funkčnímu) procesu. OS také nabízí uživateli prostředky k násilnému ukončení nekomunikující aplikace. Všechny ostatní běžící procesy fungují normálně dál.

Postupné přidělování zdrojů je buď **cyklické** (bez ohledu na charakter procesů má každý daný dobu po kterou může procesor využívat) v určitých časových intervalech nebo se děje podle **priority** procesu. **Př:** Windows 3.x, Windows XP

### Víceuživatelské (síťové) O.S.

Kromě zajišťování multitaskingu umožňují sdílení prostředků mezi více uživateli. Musí tedy počítat s ochranou disků a souborů a dalších prostředků před neoprávněnými zásahy uživatelů.

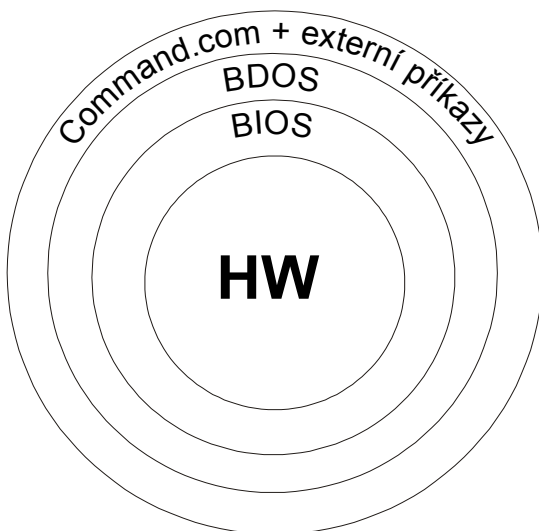
Při zahájení práce se systémem je nutné provést tzv. přihlášení. Ověří se, zda uživatel daného jména a hesla existuje v seznamu povolených uživatelů (databáze uživatelských účtů). Tento seznam vytváří a modifikuje obvykle správce operačního systému (pověřená osoba – supervisor, administrátor). Na základě přihlášení jsou potom odvozována přístupová práva daného uživatele k jednotlivým prostředkům systému. **Př:** Windows NT, Unix, Novell

## 4.3. Operační systém MS-DOS

Je to diskově orientovaný OS. Instrukce tohoto systému umožňují základní operace s daty (kopírování, mazání, tisk, apod.). Množina příkazů OS MS DOS se skládá z příkazů interních a externích, které jsou samostatnými programy.

Interní příkazy jsou příkazy uloženy v operačním systému a jsou nedílnou součástí příkazového interpretu command.com (např. del, md, rd, ren, dir, atd.). Operační systém MS DOS je jednoúlohový a jednouživatelský operační systém.

## 4.4. Struktura OS MS-DOS



Obr. 17.: struktura OS MS-DOS

**BDOS** (jádro MS-DOS) – je nezávislý na technickém vybavení (HW), je uložen v souboru **MSDOS.SYS**

**COMMAND.COM** – příkazový interpret. Komunikuje s uživatelem a zjišťuje vykonání jeho příkazů. Jeho součástí jsou tzv. **interní příkazy OS**

**BIOS** (Basic Input-Output System) – tato vrstva závisí na HW a slouží pro přímou spolupráci s technickým vybavením. Skládá se ze dvou částí. První je uložena v paměti ROM na základní desce počítače a jmenuje se **ROM BIOS**. Druhá část je uložena na disku v souboru **IO.SYS**

BIOS dále obsahuje část zajišťující konfiguraci počítače (**SETUP**), která je uložena v paměti CMOS napájené po dobu vypnutí počítače z akumulátoru. V této části se nastavuje velikost operační paměti, připojení pružných a pevných disků aj.

Přístup do **SETUPu BIOSu** je možný ihned po zapnutí (resp. **RESETu**) počítače pomocí určité klávesy (**DEL, F1,..**)

**EXTERNÍ PŘÍKAZY** – mají své vlastní soubory (např. SYS, MEM, FORMAT, TREE, ...). reprezentované soubory s příponami COM a EXE. Aby tyto příkazy fungovaly, musí být v počítači příslušný soubor a musí být po zadání příkazu nalezen.

## 4.5. Soubory, hierarchická struktura adresářů

### Soubor

Jsou data, která spolu souvisí nebo k sobě patří, se na disku uchovávají ve formě souboru. Soubor jsou tedy určitá data nebo informace, které logicky patří k sobě.

### Adresář

Aby se usnadnila orientace v záznamech na discích, mohou se soubory organizovat do adresářů (podskupin). Operační systém MS DOS se vyznačuje možností pracovat s tzv. stromovou strukturou adresářů.

Adresář je pomyslná část disku, má svoje jméno, ale není nositelem informací na disku. Nositelem informací jsou soubory. Do adresáře se ukládají informace o uložených souborech.

### Hierarchická struktura adresářů

Každý disk obsahuje tzv. hlavní adresář (kořenový). Hlavní adresář (ROOT) se vytvoří automaticky po naformátování disků příkazem FORMAT. Ostatní podadresáře již uživatel může vytvářet sám dle svých potřeb.

Tento hlavní adresář může obsahovat podřízené adresáře a soubory. Každý z podřízených adresářů může obsahovat podřízené adresáře a soubory. Hlavní adresář označujeme znakem zpětné lomítko \ .

### Název souboru

Každý soubor musí mít své **jméno a příponu**. Jméno a přípona se uvádějí vždy *spolu a oddělují se tečkou*. **Jméno může obsahovat 1 – 8 znaků, přípona 0 – 3 znaky**. Totéž platí i u adresářů. K vytvoření jména a přípony se používají **písmena a číslice**, některé znaky se nesmí použít: **čárka, tečka, závorky, znaménka** apod. Kombinací až 8 znaků lze vytvořit jméno souboru. Při volbě typu souboru tolik možností nemáme zejména proto, že přípony souborů jsou určitým způsobem zadáné. **Například:** TXT jsou soubory obsahující text, EXE a COM jsou soubory obsahující spustitelné programy, BAK jsou záložní kopie,...



Pozn.: Ve Windows mohou být jméno i přípona dlouhé dohromady maximálně 255 znaků. Jsou také navíc povoleny některé znaky zakázané v MS-DOS, jako je mezera nebo čárka aj.

## 4.6. Jednoznačná specifikace souboru

### Jednoznačná specifikace souboru

slouží k rozlišení souborů se stejnými názvy, které jsou uloženy v různých adresářích popř. na různých discích. **V jednom adresáři nemohou být uvedeny dvě položky se stejnými jmény.**

**Složení:** - jméno disku, na kterém je soubor uložen (např. C:\)

- adresářová cesta (windows\programxy\)
- vlastní název souboru (office.txt)

### Adresářová cesta

je posloupnost adresářů, kterými musíme projít, abychom se dostali k hledanému cílovému adresáři. Jednotlivé adresáře přitom oddělujeme zpětným lomítkem.

**Dělení:** a) **Relativní** – je cesta **vztahená k pracovnímu adresáři**. Nezačíná zpětným lomítkem. Příklad: novak\data

- b) **Absolutní** – je cesta **vztažená ke kořenovému adresáři** disku. Začíná zpětným lomítkem (\). Příklad: \uzivatele\novak\data  
Příklad: C: \uzivatele\novak\data

**Pracovní adresář** – je pouze 1 z celé množiny adresářů na disku a jeho jméno vidíme ve výzvě OS za jménem pracovního disku.

## 4.7. Náhradní znaky pro označování souborů

Pro současnou práci s celou skupinou souborů, jsou zavedeny v tzv. “**Náhradní znaky**”. Náhradní znaky v operačním systému MS-DOS jsou dva:

**Znak \*** – znamená, že na jeho místě a každém z následujících míst může stát libovolný znak nebo nic. Na jeho místě nemusí být nic nebo tam může stát libovolná skupina znaků a zbylé znaky se ignorují.

**Například:** \*.\* - všechny soubory v pracovním adresáři

\*.TXT - všechny soubory z příponou TXT v pracovním adresáři

**Znak ?** – znamená, že na jeho místě může stát právě jeden libovolný znak.

**Například:** ZPRÁVA?.TXT může znamenat ZPRÁVA1.TXT i ZPRÁVA2.TXT

?.TXT – všechny soubory co mají jeden znak a příponu txt. Tedy například A.TXT, X.TXT, ale nepatří sem třeba soubor AA.TXT protože v názvu má již dva znaky.

## 4.8. Zařízení v OS MS-DOS

V operačním systému MS-DOS se používají různá vstupní, výstupní a vstup-výstupní zařízení

standardní vstupní zařízení – **klávesnice**

standardní výstupní zařízení – **monitor**

volitelné výstupní zařízení – **tiskárna**

volitelné vstupní zařízení – **paralelní port**

Každému z těchto zařízení je přiřazeno určité jméno. Jména zařízení jsou tzv. *vyhrazená*. Vyhrazená jména nemůžeme používat pro označování vlastních souborů. Nesmí být nic uloženo pod stejným jménem.

**CON** – označuje standardní vstup-výstupní zařízení: pro vstup je to tedy klávesnice a pro výstup monitor

**COM1 – COM4** – označují sériová komunikační rozhraní, na která připojujeme například myš

**AUX = COM1** – ekvivalentní označení prvního sériového rozhraní

**LPT1 – LPT3** – označuje paralelní rozhraní počítače, na které připojujeme například tiskárny


**PRN = LPT1** – ekvivalentní označení prvního paralelního portu

**NUL** – označuje neexistující (prázdné) zařízení

## 5. Příkazy MS-DOS

### Vysvětlení formátování použitého v následujících kapitolách

Formát	Význam
<i>Kurzíva</i>	Informace, které by měl doplnit uživatel.
<b>Tučný text</b>	Části, které musí uživatel zadat přesně podle příkladu.
Tři tečky (...)	Parametr, který může být v příkazovém řádku několikrát opakován.
Hranaté závorky ([ ])	Volitelné položky
Složené závorky ( { } ); Volby oddělené svislou čarou (   ) Příklad: {even odd}	Sada voleb, z nichž musí uživatel vybrat pouze jednu.
Písmo Courier	Kód nebo výstup programu

 **Pozn:** Příkazy se v MS-DOS zadávají do **příkazového řádku**. **Syntaxe** určuje pořadí, ve kterém je třeba zadávat příkazy a všechny jejich parametry. V následujícím textu jsou uvedeny nejpoužívanější příkazy.

V šedém rámečku je jejich syntaxe. Slovní popis syntaxe je však velmi zkrácen. K tomu, abyste mohli zjistit co dělá který přepínač je důležité naučit se používat vestavěnou nápovědu pro každý příkaz. Nápověda se vyvolá tím, že zadáte příkaz a jako jeho jediný parametr použijete **/?**. Např. tedy zadáte **dir /?**, na obrazovku se následně vypíše nápověda se všemi parametry, které můžete pro daný příkaz použít a jejich popis.

### 5.1. Nejdůležitější příkazy MS-DOS

#### Dir (directory)

Zobrazí seznam souborů a podadresářů, které adresář obsahuje. Jestliže použijete příkaz **dir** bez parametrů, dojde k zobrazení jmenovky svazku a sériového čísla disku. Zobrazí se také seznam adresářů a souborů na disku, včetně jejich názvů a data včetně času posledních úprav. U souborů se zobrazí přípony názvu a velikost v bajtech. Příkaz **dir** také zobrazí celkový počet souborů a adresářů v seznamu, jejich velikost a zbývající volné místo na disku (v bajtech).

```
dir [Jednotka:][Cesta][Název_souboru] [/p] [/q] [/w] [/d] [/a[:atributy]] [/o[:Pořadí_řazení]] [/s] [/b] [/l] [/n] [/x] [/c]
```

**/p** Zobrazí vždy jen jednu obrazovku seznamu. Další obrazovku zobrazíte stisknutím jakékoli klávesy.

**/s** Vypíše soubory v určeném adresáři a ve všech podadresářích.

#### Příklady

Chcete-li zobrazit všechny adresáře jeden za druhým v abecedním pořadí, v širokém formátu a s pozastavením vždy po zaplnění obrazovky, zkontrolujte, zda je aktuálním adresářem kořenový adresář, a zadejte:

```
dir /s/w/o/p
```

Chcete-li zobrazit seznam všech názvů souborů s příponou TXT, které se nacházejí ve všech adresářích na jednotce C:, zadejte:

```
dir c:\*.txt /w/o/s
```

#### CD (Change Directory)

Zobrazí název pracovního adresáře nebo změní pracovní adresář. Pokud příkaz **CD** použijete pouze s písmenem jednotky (například **CD C:**), zobrazí se pouze název aktuálního adresáře a jednotky. Při použití bez parametrů zobrazí příkaz **CD** aktuální jednotku a adresář.

```
cd [/d] [Jednotka:][Cesta] [..] [/d] [Jednotka:][Cesta] [..]
```

### Příklady

Chcete-li přejít z jakéhokoli podadresáře do nadřazeného adresáře, zadejte:

```
cd ..
```

## Tree

Graficky zobrazí strukturu adresářů cesty nebo disku v jednotce.

```
tree [Jednotka:][Cesta] [/f] [/a]
```

**/f** Zobrazí názvy souborů v jednotlivých adresářích.

### Příklady

Chcete-li zobrazit názvy všech podadresářů aktuálního pracovního adresáře zadejte:

```
tree
```

Chcete-li zobrazit názvy všech podadresářů kořenového adresáře na aktuální jednotce, zadejte:

```
tree \
```

## Copy

Zkopíruje jeden nebo více souborů z daného umístění do jiného.

```
copy [d] [/v] [/n] Zdroj Cíl
```

### Příklady

Chcete-li zkopírovat soubor Drozd.typ z aktuální jednotky a adresáře do existujícího adresáře Ptaci, který se nachází na jednotce C:, zadejte:

```
copy drozd.typ c:\ptaci
```

## Ren (Rename)

Změní název jednoho nebo více souborů.

```
ren [Jednotka:][Cesta] název_souboru1 název_souboru2
```

### Příklady

Předpokládejme, že chcete změnit příponu všech názvů souborů v aktuálním adresáři, které mají příponu TXT, na příponu DOC. Tuto změnu lze provést pomocí následujícího příkazu:

```
ren *.txt *.doc
```

Jestliže chcete změnit název souboru nebo adresáře Kap10 na Dil10, zadejte:

```
ren kap10 dil10
```

## Del (Delete)

Slouží k odstranění zadaných souborů.

```
del [Jednotka:][Cesta] Název_souboru [ ...] [/p] [/f] [/s] [/q] [/a:atributy]
```

## Cls (Clear Screen)

Vymaže obsah obrazovky.

```
cls
```

## MD (Make Dir)

Vytvoří adresář nebo podadresář.

```
md [Jednotka:]Cesta
```

### Příklady

Chcete-li vytvořit v kořenovém adresáři adresář Daně a jeho podadresář s názvem Nemovitosti, který obsahuje další podadresář pojmenovaný Aktuální, zadejte:

```
md \Dane\Nemov\Aktual
```

Výsledek bude stejný jako při zadání následující sekvence příkazů při zakázaném rozšíření příkazů:

```
md \Dane
```

```
cd \Dane
```

```
md Nemov
```

```
cd Nemov
```

```
md Aktual
```

## RD (Remove Directory)

Odebere (tj. odstraní) adresář. Adresář musí být prázdný.

```
rd [Jednotka:]Cesta [/s] [/q]
```

**/s** Odstraní zadaný adresář včetně všech podadresářů a souborů. Pomocí parametru **/s** lze odstranit adresářový strom.

## Xcopy

Zkopíruje soubory a adresáře včetně podadresářů.

```
xcopy Zdroj [Cíl] [/w] [/p] [/c] [/v] [/q] [/f] [/l] [/g] [/d:mm-dd-rrrr] [/u] [/i] [/s] [/e] [/t] [/k] [/r] [/h] [/a/m] [/n] [/o] [/x] [/exclude:soubor1][+[soubor2]][+[soubor3]] [/y][/-y] [/z]
```

**/y** Potlačí výzvy k potvrzení přepsání existujícího cílového souboru.

**/-y** Zobrazí výzvu k potvrzení přepsání existujícího cílového souboru.

**/p** Vyzve uživatele k potvrzení, zda chce vytvořit všechny cílové soubory.

**/s** Zkopíruje adresáře a podadresáře, pokud nejsou prázdné.

**/t** Zkopíruje pouze strukturu podadresáře (tj. adresářový strom), nikoli soubory. Chcete-li zkopírovat i prázdné adresáře, použijte parametr příkazového řádku **/e**.

## Příklady

Chcete-li zkopírovat všechny soubory a podadresáře (včetně prázdných) z jednotky A: na jednotku C:, zadejte následující příkaz:

**xcopy a: c: /s /e**

Chcete-li do předchozího příkladu zahrnout systémové a skryté soubory, přidejte parametr příkazového řádku /h:

**xcopy a: c: /s /e /h**

## Attrib

Zobrazí, nastaví nebo odebere atributy přiřazené souborům nebo adresářům. Atributy určují některé vlastnosti souborů (adresářů), případně to, jak se s nimi má zacházet. V MS-DOS existují čtyři atributy: READ ONLY - soubor je pouze na čtení, ARCHIVE - soubor vyžaduje archivaci, HIDDEN - skrytý soubor, SYSTEM - systémový soubor.

Pokud použijete příkaz attrib bez parametrů, zobrazí se atributy všech souborů v aktuálním adresáři.

**attrib** [{+r|-r}] [{+a|-a}] [{+s|-s}] [{+h|-h}] [[Jednotka:][Cesta] *Název\_souboru*] [/s/[d]]

### Parametry

- +r** Nastaví atribut souboru Jen pro čtení.
- r** Odstraní atribut souboru Jen pro čtení.
- +a** Nastaví atribut souboru Archivovat.
- a** Odstraní atribut souboru Archivovat.
- +s** Nastaví atribut souboru Systémový.
- s** Odstraní atribut souboru Systémový.
- +h** Nastaví atribut souboru Skrytý.
- h** Odstraní atribut souboru Skrytý.

### Příklady

Chcete-li zobrazit atributy souboru s názvem Zprávy86, který je umístěn v aktuální jednotce, zadejte:

**attrib zprávy86**

Chcete-li přiřadit souboru Zpráva.txt atribut Jen pro čtení, zadejte:

**attrib +r zpráva.txt**

## 6. Přesměrování

Pomocí operátorů přesměrování příkazů lze přesměrovat vstup a výstup; do (ze) zařízení nebo souboru. Standardní výstupní zařízení je monitor, standardní vstupní klávesnice. Chceme-li tedy použít pro vstup nebo výstup jiná zařízení nebo soubor použijeme některý z operátorů přesměrování.

V následující tabulce jsou uvedeny operátory, které lze použít k přesměrování vstupních a výstupních datových proudů příkazů.

Tab. 9.: Operátory, přesměrování a propojení

Operátor přesměrování	Popis
>	Místo na obrazovku zapíše výstup příkazu do souboru nebo do zařízení, například do tiskárny.
<	Načte vstup příkazu ze souboru namísto z klávesnice.

>>	Připojí výstup příkazu na konec souboru bez odstranění informací, které jsou již v souboru obsaženy.
	Načte výstup z jednoho příkazu a zapíše jej na vstup jiného příkazu. Nazývá se také kanál.

### **Přesměrování vstupu příkazu (<)**

Chcete-li přesměrovat vstup příkazu z klávesnice na soubor nebo zařízení, použijte operátor <.

Př.: Chcete-li například získat vstup příkazu **sort** ze souboru soubor.txt, zadejte:

```
sort<soubor.txt
```

Obsah souboru soubor.txt se zobrazí na obrazovce a bude uspořádán podle abecedy.

### **Přesměrování výstupu příkazu (>)**

Téměř všechny příkazy odesílají výstup standardně na obrazovku (standardní výstupní zařízení).

Chcete-li přesměrovat výstup příkazu z obrazovky do souboru nebo zařízení, použijte operátor >. Tento operátor lze používat u většiny příkazů.

Př.: Chcete-li například přesměrovat výstup příkazu **dir** do souboru seznam.txt, zadejte:

```
dir>seznam.txt
```

Pokud soubor seznam.txt neexistuje, pak se vytvoří.

### **Použití operátoru přesměrování (>>) k připojení výstupu**

Chcete-li přidat výstup příkazu na konec souboru beze ztráty informací, které jsou již v souboru obsaženy, použijte dvojici po sobě následujících znaků větší než (>>).

Př.: Následující příkaz například připojí seznam adresářů vytvořený příkazem **dir** k souboru seznam.txt:

```
dir>>seznam.txt
```

### **Použití operátoru propojení (|)**

Operátor propojení (|) načte výstup jednoho příkazu a směruje jej na vstup jiného příkazu.

Př.: Následující příkaz například seřadí obsah adresáře:

```
dir | sort
```

Oba příkazy se v tomto příkladu spustí současně, ale příkaz **sort** je ihned pozastaven do okamžiku, kdy přijme výstup příkazu **dir**.



## 7. Filtry

Příkazové filtry se používají v kombinaci s operátorem propojení příkazu svislá čára (|), filtry lze použít i s operátory přesměrování (<, >, >>). Filtry jsou speciální typy příkazů, které načtou výstup jiného příkazu, provedou transformaci takto získaného vstupu a zapíší výsledný výstup. Příkazy filtrů pomáhají řadit, zobrazovat a vybírat části výstupu příkazu. V MS-DOS existují tři filtry **More**, **Find** a **Sort**.

Pokud chcete jako vstup příkazu filtru použít výstup ze souboru, použijte operátor přesměrování < (je menší). Chcete-li, aby příkaz filtru získal vstupní hodnoty z jiného příkazu, použijte operátor propojení svislá čára (|).

### 7.1. Použití filtru more

Filtr **more** zobrazí obsah souboru nebo výstup příkazu na obrazovce postupně po jednotlivých obrazovkách. Příkaz **more** tedy můžete použít při práci s příkazem, jehož výstup zabírá více než jednu obrazovku.

```
příkaz | more [/c] [/p] [/s] [/tn] [+n]
```

```
more [[/c] [/p] [/s] [/tn] [+n]] < [Jednotka:] [Cesta] Název_souboru
```

Př.: Předpokládejme například, že chcete zobrazit strom adresářů na pevném disku. Pokud je na pevném disku více adresářů, než lze zobrazit na obrazovce, můžete použít příkaz **tree** se operátorem svislá čára (|) a s příkazem **more**:

```
tree c:\ | more
```

Př.: Chcete-li zobrazit obsah souboru nazvaného list.txt na obrazovce po jednotlivých obrazovkách, zadejte:

```
more < list.txt
```

Obrazovka se zaplní informacemi a v jeho dolní části se objeví výzva -- Další --. Chcete-li na obrazovce zobrazit další část informací, stiskněte libovolnou klávesu kromě klávesy PAUSE. Chcete-li příkaz přerušit bez zobrazení dalších informací, stiskněte kombinaci kláves CTRL+C.

### 7.2. Použití filtru find

Filtr **find** vyhledá v souborech zadaný text. Filtr Find zobrazí všechny řádky obsahující zadaný text na obrazovce. Příkaz **find** lze použít jako filtrovací příkaz nebo jako standardní příkaz.

```
find [/v] [/c] [/n] [/i] "řetězec" [[Jednotka:][Cesta]Název_souboru[...]]
```

**/v** Zobrazí všechny řádky, které hledaný *řetězec* neobsahují.

**/c** Zjistí, kolik řádků obsahuje zadaný *řetězec*, a zjištěný počet zobrazí.

**/i** Určuje, že při hledání nebudou rozlišována malá a velká písmena.

Chcete-li použít příkaz **find** jako příkaz filtru, musíte spolu s ním použít operátor přesměrování < (menší než) a hledaný *řetězec* nebo text. Ve výchozím stavu se při hledání pomocí příkazu **find** rozlišují velká a malá písmena.

Př.: Následující příkaz například vyhledá výskyty řetězce Velká Británie v souboru Obchod.txt:

**find "Velká Británie" < obchod.txt**

Ve výstupu se neobjeví žádné výskyty řetězce "velká británie". Budou nalezeny pouze výskyty řetězce "Velká Británie" s prvními počátečními písmeny.

Pokud nechcete výstup příkazu **find** zobrazit na obrazovce, ale chcete jej uložit, zadejte operátor přesměrování > (je větší) a název souboru, do kterého chcete výstup uložit.

Př.: Následující příkaz například vyhledá výskyty řetězce Velká Británie v souboru obchod.txt a uloží výsledek do souboru obchod2.txt:

```
find "Velká Británie" < obchod.txt > obchod2.txt
```

### 7.3. Použití filtru sort

Filtr **sort** načte vstup, seřadí data a zobrazí výsledky na obrazovce nebo je zapíše do souboru či na jiné zařízení.

```
sort [/r] [/+n] [/m kilobajty] [/l národní_prostředí] [/rec znaky] [jednotka1:]
[cesta1]název_souboru1] [/t [jednotka2:][cesta2]] [/o [jednotka3:][cesta3]název_souboru3]
[příkaz |] sort [/r] [/+n] [/m kilobajty] [/l národní_prostředí] [/rec znaky] [[jednotka1:]
[cesta1]název_souboru1][t [jednotka2:][cesta2]] [/o [jednotka3:][cesta3]název_souboru3]
```

#### Parametry

/r Změní pořadí řazení (řazení se provádí od Z do A a od 9 k 0).

#### Příklady

##### Řazení souboru

Př.: Následující příkaz přečte soubor Náklady.txt, seřadí jej v obráceném pořadí a zobrazí na obrazovce:

```
sort /r náklady.txt
```

##### Řazení výstupu příkazu

Př.: Chcete-li ve velkém souboru nazvaném Seznam.txt vyhledat text Novák a seřadit výsledky hledání, přeměrujte pomocí operátoru propojení svislé čáry (|) výstup příkazu **find** do příkazu **sort**, jak ukazuje následující příklad:

```
find "Novák" seznam.txt | sort
```

Tento příkaz vytvoří seřazený seznam řádků obsahujících zadaný text.

Filtr **sort** seřadí textový soubor nebo výstup příkazu podle abecedy.

Př.: Následující příkaz například seřadí obsah souboru Seznam.txt a výsledek zobrazí na obrazovce.

```
sort < seznam.txt
```

Př.: V tomto příkladu příkaz **sort** seřadí řádky souboru Seznam.txt podle abecedy a zobrazí výsledek, aniž by došlo ke změně souboru. Pokud nechcete výstup příkazu **sort** zobrazit, ale chcete jej uložit, zadejte operátor > (je větší) a název souboru. Následující příklad například seřadí řádky souboru seznam.txt podle abecedy a výsledek uloží do souboru abseznam.txt:

```
sort < seznam.txt > abseznam.txt
```

Př.: Chcete-li seřadit výstup příkazu, zadejte požadovaný příkaz, operátor propojení (|) a příkaz **sort** (tedy *příkaz | sort*). Následující příkaz například seřadí řádky obsahující řetězec Novák (tj. výstup příkazu **find**) podle abecedy:

```
find "Novák" seznam.txt | sort
```

## 8. Dávkové soubory

Dávkové soubory (dávký) se používají pro často se opakující úlohy. Obsahují posloupnost příkazů MS-DOSu, která se při spuštění dávky vykoná.

*Pozn.: Místo aby uživatel pokaždé zadával příkazy z klávesnice, vytvoří si dávku kterou pak dle potřeby spouští (volá).*

Dávkový soubor je spustitelný, textový soubor s příponou BAT. Obsahuje příkazy operačního systému MS-DOS, které se provedou při jeho volání.

Dávkový soubor (dávká) může obsahovat kromě interních a externích příkazů, spouštění uživatelských programů a volání jiných dávkových souborů i speciální příkazy použitelné v dávkových souborech.

Dávkový soubor lze vytvořit:

- a) COPY CON davka.bat
- b) v textovém editoru (např. EDIT), který ukládá tzv. čistý ASCII text (bez formátování)

### Parametry dávkových souborů

V dávkových souborech lze používat parametry, kterými se dá ovlivňovat zpracování dávky. Parametry dělíme na:

- a) **formální** – jsou použity při definici dávky (zapsány v dávce), při volání dávky jsou do nich umístěny skutečné parametry
- b) **skutečné** – jsou použity při volání dávky

Formální parametry zapsané v dávce začínají znakem % následované číslicí 0 až 9 (tj. %0, %1, ..., %9). Formální parametr %0 má speciální význam – obsahuje jméno právě prováděného dávkového souboru.

Vysvětlení parametrů dávkových souborů

- A. Vytvoříme dávkový soubor NOVYADR.BAT. Tento dávkový soubor bude používat jeden **formální parametr %1**, který se použije jako parametr příkazu MD. Dávkový soubor NOVYADR.BAT bude vypadat následovně:

```
MD %1
```

Jestliže budeme chtít vytvořit adresář MUJ1, pak zavoláme dávku se **skutečným parametrem MUJ1**.

```
P:\PRAC\>NOVYADR MUJ1
```

Při volání dávkového souboru NOVYADR se formálnímu parametru %1 (je zapsaný v dávce) přiřadí první skutečný parametr MUJ1 (s tímto parametrem je dávka volána) a provede se příkaz MD MUJ1.

- B. Vytvoříme dávkový soubor NAJDI.BAT. Tento dávkový soubor bude používat dva **formální parametry %1 a %2**, které se použijí jako parametry příkazu FIND. První parametr bude hledaný řetězec a druhý parametr bude název souboru ve kterém se bude řetězec hledat. Dávkový soubor NAJDI.BAT bude vypadat následovně:

```
FIND %1 %2
```

Jestliže budeme chtít najít řetězec „Pepa“ v souboru LIDE.TXT, pak zavoláme dávku se **skutečnými parametry Pepa a LIDE.TXT**.

```
P:\PRAC\>NAJDI „Pepa“ LIDE.TXT
```

Při volání dávkového souboru NAJDI se formálnímu parametru %1 přiřadí první skutečný parametr „Pepa“, formálnímu parametru %2 se přiřadí druhý skutečný parametr LIDE.TXT a provede se příkaz FIND „Pepa“ LIDE.TXT

## 8.1. Příkazy pro dávkové soubory

### ECHO

```
echo [{on|off}]
```

```
echo [text]
```

Zapne (ON) nebo vypne (OFF) vypisování prováděných příkazů (v dávce) na obrazovku, tento příkaz platí pro právě prováděnou dávku.

ECHO text zobrazí zadaný text na obrazovce. Před příkaz lze zadat znak zavináč @ který zabrání vypsání příkazu na obrazovku. Chcete-li na obrazovku vypsát prázdný řádek, zadejte **echo**. (echo s tečkou na konci).

Př.: Následuje příklad dávkového programu, který vypne vypisování odezvy příkazu, potom zobrazí prázdný řádek, nakonec zprávu *SPŠ a SOU Kuřim, s.r.o.*

```
ECHO OFF
```

```
ECHO.
```

```
ECHO SPŠ a SOU Kuřim, s.r.o.
```

Jestliže chcete vypnout příkaz echo a nechcete, aby se příkaz **echo** zobrazil na obrazovce, zadejte před ním znak @:

```
@ECHO OFF
```

### CALL

```
call [jednotka:][cesta] dávka2 [parametry dávky2]
```

Z právě prováděné dávky zavolá jinou dávku, po jejím ukončení se pokračuje v provádění příkazů původní dávky za příkazem CALL.

Př.: Dávka1.bat smaže obrazovku. Poté vypíše *Provádíme dávku1*, zavolá se a spustí dávka2.bat. Dávka2.bat vypíše dva řádky. Po jejím provedení se pokračuje v provádění příkazů dávky1.bat. Pokračuje se tedy vypsáním řádku *Toto je opět v dávka1*.

```
dávka1.bat
```

```
CLS
```

```
ECHO Provádíme dávku1
```

```
CALL dávka2.bat
```

```
ECHO Toto je opět v dávka1
```

```
dávka2.bat
```

```
ECHO Provádíme dávku2
```

```
ECHO Po skončení dávky se vrátíme do původní
```

### PAUSE

**pause**

Pozastaví provádění dávky a čeká na stisk klávesy. Po stisknutí klávesy dávka pokračuje.

**REM****rem** [*komentář*]

Příkaz REM uvádí v dávce komentář, tj. cokoli za REM až do konce řádku se považuje za komentář.

Př.: Dávka, která nic neprovede:

**REM** *tahle dávka nedělá nic*

**REM** CLS

**REM** ECHO Dobrý den

**GOTO****goto** *návěští*

Pomocí GOTO lze měnit pořadí provádění příkazů v dávce; v dávce se provede skok na příkaz za návěštím. Návěští je na samostatném řádku, začíná dvojtečkou a první znak nesmí být číslice.

Př.: Pořadí příkazů provedených v této dávce je CLS, ECHO Zpráva1, GOTO skok, ECHO Zpráva3 (příkaz ECHO Zpráva2 se neprovede).

**CSL**

**ECHO** Zpráva1

**GOTO** skok

**ECHO** Zpráva2

**:SKOK**

**ECHO** Zpráva3

**FOR****for** {*%proměnná*}%*%proměnná*} **in** (*množina*) **do** příkaz [*Parametry\_příkazu*]

Pro každou proměnnou z množiny hodnot proved' příkaz

Pozn.: Příkaz **for** můžete použít v rámci dávkového souboru nebo přímo z příkazového interpretu. Při provádění z příkazového interpretu použijte parametr *%proměnná*. V dávkovém souboru se místo *%proměnná* zapíše *%%proměnná*.

Klíčová slova **in** a **do** nejsou parametry, jejich použití je však pro příkaz **for** povinné. Pokud některé z těchto slov vynecháte, zobrazí se chybová zpráva.

Parametr *množina* může představovat jednu nebo více skupin souborů. K zadání množiny souborů můžete použít náhradní znaky (\* a ?). Následující příklady představují platné zápisy množin souborů: (**\*.doc**) nebo (**\*.doc \*.txt \*.dat**) nebo např. (**zprava?.\* dopis??.\***)

Funkce: použití příkazu **for** nahradí první hodnota uvedená v parametru *množina*

proměnnou *%proměnná* nebo proměnnou *%%proměnná* a zadaný příkaz tuto hodnotu zpracuje.

Tento postup se opakuje, dokud nejsou zpracovány všechny soubory (nebo skupiny souborů), které odpovídají hodnotě parametru *množina*.

**Př.:** Chcete-li zobrazit obsah všech souborů v aktuálním adresáři, které mají příponu DOC nebo TXT, pomocí nahraditelné proměnné %%f, zadejte:

**FOT %%f IN (\*.doc \*.txt) DO TYPE %%f**

Ve výše uvedeném příkladu budou za proměnnou %%f postupně dosazeny všechny soubory s příponou DOC nebo TXT v aktuálním adresáři a obsah každého takového souboru bude zobrazen.

**Př.:** Chcete-li všechny soubory s příponou BAK v pracovním adresáři zkopírovat do adresáře ZALOHA (podadresář kořenového adresáře) zadejte:

**FOR %%f IN (\*.bak) DO COPY %%f \ZALOHA**

## SHIFT

### shift

Příkaz SHIFT provádí posunutí obsahu formálních parametrů, tj. do %0 se zapíše obsah %1, do %1 se zapíše obsah %2 atd.

formální parametr	obsahuje	SHIFT provede	formální parametr	nyní obsahuje
%1	123	do %1 přiřadí %2	%1	ahoj
%2	ahoj	do %2 přiřadí %3	%2	DATA
%3	DATA	do %3 přiřadí %4	%3	ano.txt
%4	ano.txt	do %4 přiřadí %5	%4	obr.bmp
%5	obr.bmp		%5	

## IF

**if [not]** podmínka *příkaz*

**if [not] errorlevel** číslo *příkaz*

**if [not] řetězec1==řetězec2** *příkaz*

**if [not] exist** *Název\_souboru* *příkaz*

Příkaz **IF** je podmíněný příkaz. Provádí podmíněné zpracování v dávkových programech.

**IF** podmínka *příkaz* – pokud platí podmínka (je pravdivá), pak provede příkaz.

**IF NOT** podmínka *příkaz* – pokud podmínka neplatí (je nepravdivá), pak provede příkaz.

Jako podmínku lze použít

**ERRORLEVEL** číslo; pokud je hodnota proměnné ERRORLEVEL větší nebo rovna zadanému číslu (proměnná ERRORLEVEL obsahuje návratový kód programu nebo příkazu, např. CHOICE)

**řetězec1==řetězec2**; pokud se dva řetězce rovnají (tj. jsou shodné)

**EXIST** soubor; pokud soubor existuje

**EXIST** *Název\_souboru*; pokud existuje soubor

**Př.:** Pokud nelze nalézt soubor Produkt.dat, zobraz zprávu Datový soubor nebyl nalezen.:

**if not exist produkt.dat, echo Datový soubor nebyl nalezen.**

**Př.:** Příkaz **if** nelze přímo použít ke zjištění, zda existuje určitý adresář. V každém adresáři však existuje nulové zařízení (NUL) a existenci adresáře je tedy možné ověřit otestováním existence nulového zařízení.

Následující příklad ověřuje existenci adresáře mujadr:

**if exist c:mujadr\nul echo Adresář existuje.**

Další příklady použití:

```

IF ERRORLEVEL 3 GOTO Volba3
IF %1=="" GOTO Chybi
IF %1==%2 ECHO Parametry jsou shodné
IF NOT EXIST \ZALOHA\nul MD \ZALOHA
IF EXIST soubor.bak DEL soubor.bak

```

## CHOICE

**choice** [/C:volby] [/N] [/S] [/T:c,nn] [text]

Příkaz čeká, dokud uživatel nezvolí jednu z přípustných voleb. Po stisku povolené klávesy vloží do proměnné ERRORLEVEL pořadí stisknuté klávesy (ze seznamu přípustných voleb).

/C:volby	určuje přípustné klávesy
/S	rozlišuje malé a velké znaky
/T:c,nn	c je výchozí volba po uplynutí nn sekund
text	text výzvy, která se vypíše

Př.: Uživatel může stisknout klávesu A nebo N. V případě A se do proměnné ERRORLEVEL uloží hodnota 1, v případě N hodnota 2.

**CHOICE /C:AN Pokračovat v dávce?**

## SET

**set** [proměnná=hodnota]

Vypíše, nastaví nebo odstraní proměnné MS-DOSu. Příkaz SET bez parametrů vypíše aktuální nastavení proměnných.

## PATH

**path** [[jednotka:]cesta[;...]]

Vypíše nebo nastaví cestu pro vyhledání spustitelných souborů (\*.COM, \*.EXE, \*.BAT). Při spouštění příkazů postupuje MS-DOS následovně: nejprve ověří, zda se jedná o interní příkaz, pokud ne, nejprve prohledá pracovní adresář. Pokud v něm nenajde příslušný spustitelný soubor, začne procházet adresáře definované v nastavení PATH.

*MS-DOS má následující prioritu spouštění příkazů: 1. COM, 2. EXE, 3. BAT. Tj. existují-li v pracovním adresáři soubory edit.com, edit.exe a edit.bat, pak zadáním příkazu edit se spustí edit.com. Zadáme-li jako příkaz edit.bat, pak se spustí edit.bat*

Př.: Následující příkaz určuje, že OS má externí příkazy vyhledávat ve třech adresářích. Cesty těchto tří adresářů jsou: C:\Uzivatel\Dane, D:\Uzivatel\Investice a C:\dos.

**PATH c:\uzivatel\dane;d:\uzivatel\investice;c:\dos**

## PROMPT

**prompt** [text]

Mění formát výzvy MS-DOSu. Text určuje tvar systémové výzvy na příkazovém řádku.

## 8.2. Zadání příkladů dávkových souborů:

- 1) Vytvoř dávkový soubor JA.BAT, který smaže obrazovku a vypíše tvoje jméno a příjmení.
- 2) Vytvoř dávkový soubor TRIDA.BAT, který smaže obrazovku a vypíše na jeden řádek třídu a na druhý řádek tvoje jméno a příjmení.
- 3) Vytvoř dávkový soubor SKOLA.BAT, který smaže obrazovku a vypíše na 1. řádek název školy, na 2. řádek ulici a č.p., na 3. řádek obec a na č. řádek PSČ.
- 4) Vytvoř dávkový soubor SKOLA2BAT, který smaže obrazovku a vypíše na 1. řádek název školy, počká na stisk klávesy, vypíše na 2. řádek ulici a č.p., počká na stisk klávesy, vypíše na 3. řádek obec, počká na stisk klávesy, vypíše na 4. řádek PSČ.
- 5) Vytvoř kopii dávky SKOLA2.BAT, pojmenuj ji SKOLA3.BAT. Uprav dávku SKOLA3.BAT tak, aby se nevypisovaly prováděné příkazy.
- 6) Vytvoř dávku MUJTYPE2.BAT (opiš příkazy dle zadání), dávku doplň o komentáře.

```
@echo MUJDIR, syntaxe: mujdir pripona
dir *.*%1 /b
```

```
@echo MUJTYPE, syntaxe: MUJTYPE pripona
for %%f in (*.%1) do type %%f
```

```
@echo off
echo TESTIF, testovani prikazu IF
if exist *.*%1 echo Existuje soubor s priponou %1
```

```
@echo off
echo MUJTYPE2, syntaxe: MUJTYPE2 pripona [pripona[,pripona...]]
:cyklus
if "%1" == "" goto konec
for %%f in (*.%1) do type %%f
shift
goto cyklus
:konec
```

```
@echo off
echo MUJREN, syntaxe: MUJREN stary novy
if not exist %1 goto nemam
echo Chcete prejmenovat soubor s nazvem %1 na novy nazev %2
choice /c:an ano, ne?
if errorlevel 2 goto nechci
if errorlevel 1 goto chci
:chci
ren %1 %2
echo Provedeno!
goto total
:nechci
echo Takze nakonec zustane vse pri starem
goto total
:nemam
echo Pozadovany soubor nebyl nalezen ...
:total
```

- 7) Vytvoř dávku JMENO.BAT, která vypíše svoje jméno (dávka musí pracovat správně i po přejmenování!).



- 8) Vytvoř dávku TESTSOUB.BAT, které uživatel při volání předá jako skutečný parametr název souboru, dávka otestuje zda zadaný soubor existuje v pracovním adresáři a vypíše na obrazovku výsledek testu (existuje/neexistuje).
- 9) Vytvoř dávku TEST888.BAT, které uživatel předá skutečné parametry; dávka otestuje, zda se některý rovná řetězci „888“ a vypíše výsledek (888 nalezeno/nenalezeno).
- 10) Vytvoř dávku NAJDIVIC.BAT, které uživatel při volání předá 3 skutečné parametry, 1. a 2. bude hledaný řetězec, 3. parametr bude název souboru ve kterém se bude hledat.
- 11) Vytvoř dávku PARAM.BAT, která pomocí cyklu vypíše všechny skutečné parametry (max. 9), které jí uživatel předal při jejím volání.
- 12) Vytvoř dávku COSNIM, které uživatel zadá název souboru a dávka zobrazí volbu pro vypsání obsahu tohoto souboru nebo pro jeho smazání, na základě stisku klávesy uživatelem dávka provede příslušný příkaz.