

● Textová informace

- dopisy, dokumenty, texty
- *editování* = práce s textem → program k úpravě textu je *textový editor*
- *textový editor* = psaní, znaky, mazání, ukládání do souborů, otevírání (příkazový řádek)
- vyspělé textové editory → *textové procesory*
- textový procesor (například Word, LibreOffice Writer)
- formáty = způsoby charakteristiky textu

ASCII text – čistý text bez podpory formátu, neomezeně přenositelný mezi libovolnými textovými editory a procesory, různými systémy a mobilními telefony

RTF – vyvinut Microsoftem, nezávislý na platformě, mnoho formátovacích příkazů, podpora všech slušných editorů, při prohledávání zdrojového kodu je možno identifikovat jednotlivé příkazy, umožňuje uchovat obrázky ale za cenu nárůstu velikosti

PDF – (formát pro přenositelné dokumenty) vyvinut firmou Adobe, navržen taky, aby se ve všech počítačích i tiskárnách zobrazoval stejně, speciální prohlížeč = Adobe Reader

- PostScript = programovací jazyk určený ke grafickému popisu tisknutelných dokumentů
vyvinut 1958 Adobe Systems Incorporated
nezávislý na zařízení
standart dražší tiskárny
formát také pro ukládání obrázků

princip : množina mat. příkazů určuje, jak má být strana rozvržena, používá souřadnicový systém k určování jednotlivých objektů (graf. Prvky, t text...) souřadnice x a y, zadána v bodech (points = 1/72 palce), výstupní zařízení disponuje vlastním prostorem = prostorem zařízení → uživatelský prostor lze posouvat nebo natáčet

složeno z – prolog a script

prolog – složen z informací nezbytných pro správný tisk dokumentu, obsahuje i informace o záhlaví a definici jednotlivých procedur

script – složen z aktuálního popisu úlohy na úrovni jedné stránky

Syntax PostScriptu

programy v jazyce PS se píší v postfixové notaci = operátor uveden vždy za operandy, na něž je aplikován

např.: clear – vymaže obsah zásobníku, add – sčítání, div-dělení, findfont – vyhledá příslušný font ve slovníku fontů a výsledek předá překladači...

● Komprimace dat

- = speciální postup při ukládání nebo transportu dat. Úkolem komprese je zmenšit datový tok nebo zmenšit potřebu zdrojů při ukládání informací
- mnoho kompresních metod
- kompresní principy se ale nezabývají tím co nastane, když je použijeme v praxi
- → to ale řeší kompresní metoda, která je spojena s datovým formátem (například .zip, .arj, .rar, atd.)
- dělení :
 - základní dělení
 - ztrátové – transformace pomocí DCT (u JPEG), fraktální komprese, mezisímková komprese (u MPEG)
 - neztrátové – dále dělíme na *primární* (RLE, LZW, LZ77) a *sekundární* (Huffmanova, Aritmetická komprese)
 - dle výpočetní náročnosti
 - symetrické
 - asymetrické
 - dle počtu průchodů
 - jedno-průchodové
 - více-průchodové

◆ Kódování RLE

- bezztrátové, primární, jedno-průchodové, takřkasymetrické
- jednoduché, pro velkou třídu dat efektivní
- vychází z předpokladu, že ve vstupních datech se opakují „sousední“ hodnoty
- Run-Length coding (metoda pro kódování délky runu = skupiny dat majících stejnou hodnotu)
- například: vstupní data - „AAAhoooj“ a výstupní data- „<3A>h<4o>j“
- musíme ale odlišit data od odkazů -pro správnou dekompresi

◆ Kódování LZ77

- bezztrátové, primární, jedno-průchodové, asymetrické
- někdy označována LZSS
- z roku 1977 -Lempel, Ziv
- snaží se nalézt ve vstupních datech opakující se sekvence symbolů a ty na výstup předat ve formě odkazu na předchozí (již odeslaná) data
- důležitá je délka opakující se sekvence
- nutnosti odlišit běžná data od příkazů pro správnou dekompresi

Příklad:

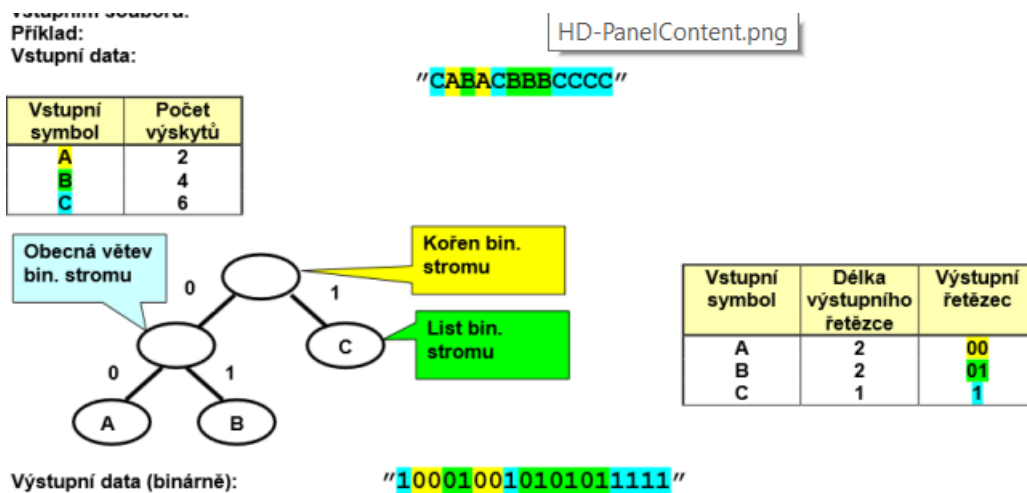
Vstupní data:

```
"the_rain_in_Spain_falls_mainly_in_the_plain"  
".....ain....."  
".....ain....."
```

Výstupní data: (V příkladu lze ušetřit ještě jeden byte, zjistíte kde?)

```
"the_rain_<3,3>Sp<9,4>falls_m<11,3>ly_<16,3><34,4>pl<15,3>"
```

- ◆ Huffmanovo kódování
 - sekundární, neztrátové, mírně asymetrické, nejčastěji dvou-průchodové
 - 1962 Huffmanova
 - použití dvou různě dlouhých bitových kodů pro symboly s různou frekvencí výskytu.
 - Morseovka
 - nejčastěji používané symboly mají nejkratší výstupní kod
 - principem je vytvoření binárního stromu na základě výskytu jednotlivých symbolů ve vstupním souboru



● Značkovací jazyky

- dnes využití v podnikové sféře
- první podnět byl ale standardizace a jednotnost v rámci výměny informací
- velký boom způsobil velký konkurenční boj o platformy, operační systémy, softwarové aplikační vybavení – zejména pro kancelářské použití
- = jazyk, jehož zdrojový text obsahuje současně jak vlastní text, tak instrukce pro jeho zpracování
- například : ASCII → Poznámkový blok
- typický rys jsou znaky se speciálním významem (<, >) text mezi nimi je chápán jako instrukce, kterou zpracuje software
- výhody: nevyžadují speciální programové vybavení pro svou editaci, lze je snadno strojově generovat,
- nevýhody : musíte mít určité znalosti abyste je mohli použít
- dělení :

popisné (deskriptivní)

- konstrukce slouží k popisu, co jsou informace obsažené v dokumentu zač
- například: XML , HTML

výkonné (procedurální)

- obsahují i výkonné instrukce na úrovni programovacího jazyka
- typicky určitou formu paměti či proměnných a nástroje pro přiřazování a využívání jejich hodnot
- umožňují detailně popsat vizuální charakteristiky výstupu

- uživatel může přesně řídit vzhled výsledného dokumentu
- například: TeX, PostScript
- k demonstraci vyjadřovací síly TeX byl vytvořen BASIC → přestože se jedná o typografický program

◆ HTML

- tvoření webových stránek
- = HyperText Markup Language
- → značkovací jazyk pro hypertext
- nyní k dispozici verze 4.01 pracuje se na 5, která přinese výrazné vylepšení
- můžete zobrazit v každém prohlížeči, pokud zadáte příkaz *Zobrazit zdrojový kód stránky*

Zdrojový kód v HTML

```
<html>
  <!-- toto je komentář -->
  <head>
    <meta charset="kódování">
    <title>Titulek stránky</title>
  </head>

  <!-- tělo dokumentu -->
  <body>
    <h1>Nadpis stránky</h1>
    <p>Toto je tělo dokumentu</p>
  </body>
</html>
```

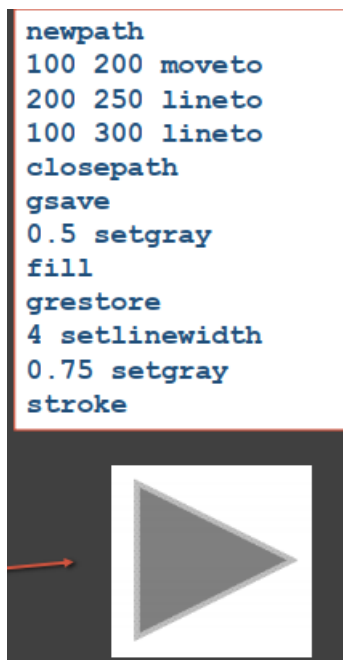
◆ XML

- česky
- použití pro serializaci dat mezi programy a službami
- podporován celou řadou nástrojů a programovacích jazyků
- především pro výměnu dat mezi aplikacemi a pro publikování dokument, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí
- nezabývá se vzhledem
- výhoda: pro většinu programů srozumitelný, můžeme přidávat libovolné množství položek bez programu

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Poznámka je nutné přidat více receptů. -->
<recept jméno="chleba" čas_přípravy="5 minut"
čas_vaření="3 hodiny">
  <titulek>Jednoduchý chleba</titulek>
  <přísada množství="3"
jednotka="šálky">Mouka</přísada>
  <přísada množství="0,25"
jednotka="unce">Kvasnice</přísada>
  <přísada množství="1,5"
jednotka="šálku">Horká voda</přísada>
  <přísada množství="1" jednotka="kávová
lžička">Sůl</přísada>
  <instrukce>
    <krok>Smíchejte všechny přísady dohromady a
dobře prohnětte.</krok>
    <krok>Zakryjte tkaninou a nechejte hodinu v
teplé místnosti.</krok>
    <krok>Znovu prohnětte, umístěte na plech a
pečte v troubě.</krok>
  </instrukce>
</recept>
```

◆ PostScript

- určený ke grafickému popisu tisknutelných dokumentů
- vyvinut 1985 firmou Adobe Systems Incorporated
- výhoda : nezávislý na zařízení, na kterém má dokument vytisknout
- standart pro dražší tiskárny
- později i pro ukládání obrázků
- množina matematických příkazů, která určuje rozvržení stránky
- výsledný dokument může být podstatně větší, ale všechn jeho obsah je převeden na jeho jednoduché příkazy



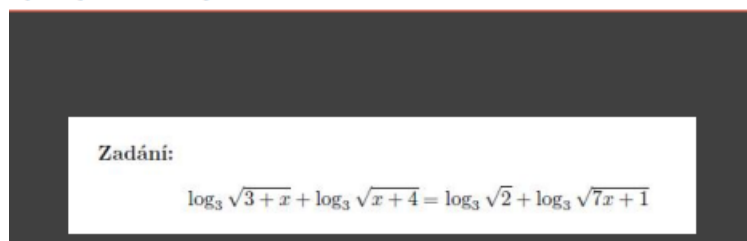
◆ TeX

- využití v matematice, fyzice, informatice
- nejlepší nástroj pro sazbu složitějších vzorců
- existuje několik sad maker (příkazů) například LaTeX také zdarma
- je částečně patrné co bude dokument obsahovat ale není to jako ve wordu – jednotlivé části dokumentu jsou popsány příkazy, které poté překladač přeloží

```
\documentclass{article}
\pagestyle{empty}
\usepackage{czech}
\begin{document}
\bf{Zadání:}

$$\log_3 \sqrt{3+x} + \log_3 \sqrt{x+4} = \log_3 \sqrt{2} + \log_3 \sqrt{7x+1}$$

\end{document}
```



● Digitalizace dat

➤ dělíme :

text

- ASCII tabulka – každý znak má svůj kod

zvuk

- 1s zvuku se rozdělí na 44 100 vzorků
- každý vzorek má určitou výšku / úroveň tonu
- každá úroveň má svůj kod

obrázky – grafika

- rastr x vektor
- rastr – pixel (každý pixel má svoji barvu určenou kodem)

video

- = animace + zvuk
- animace = série obrázku v malých časových úsecích
- počet snímků za sekundu je 24