

# PB002 – Základy informačních technologií

## Operační systémy

6. října 2010

- 1 HCILAB - Laboratoř interakce člověka s počítačem
- 2 Laboratoř konstrukce a architektury číslicových systémů
- 3 Laboratoř optické mikroskopie
- 4 Laboratoř znalostních a informačních robotů
- 5 NLP – Laboratoř zpracování přirozeného jazyka
- 6 ParaDiSe
- 7 LSD – Laboratoř vyhledávání a dialogu
- 8 Laboratoř pokročilých síťových technologií
- 9 Laboratoř vyhledávání znalostí
- 10 Laboratoř kvantového zpracování informace a kryptografie
- 11 LaBAK – Laboratoř bezpečnosti a aplikované kryptografie
- 12 SYBILA – Laboratoř systémové biologie

- 1 Dle základu: dvojková, osmičková, šestnáctková, desítková
- 2 Volně mezi sebou převoditelné (celá čísla bez ztráty přesnosti)
- 3 Celá čísla a zlomky
- 4 Reálná čísla
- 5 První počítač v desítkové soustavě
- 6 Konečné reprezentace

- ① Základem je číslo dvě
  - ① pouze dvě číslice – dva stavy
  - ② vhodná pro reprezentaci v číslicových systémech

## 1 Pro reálná čísla

- 1 Rozlišitelnost (nejmenší zobrazitelné číslo):  $X + \epsilon > X$   
a  $X + \epsilon/2 = X$
- 2 Přesnost (rozsah)
- 3 Zobrazení: mantisa  $m$  a exponent  $e$   
 $0 \leq m \leq 1 \wedge x = m \cdot 2^e$

## 2 Záporná čísla

- 1 Přímý kód
- 2 Inverzní kód (binární negace)
- 3 Dvojkový doplňkový kód (binární negace + 1))

## 1 Přímý kód

- 1 přidáno znaménko - znaménkový bit
- 2 Dvě nuly:  $+0$  a  $-0$  ( $10\dots00$ )
- 3 Rozsah:  $\langle -MAX, -0 \rangle$  a  $\langle +0, +MAX \rangle$

## 2 Inverzní kód:

- 1 Přidáme znaménko
- 2 Dvě nuly:  $+0$  a  $-0$  ( $11\dots11$ )
- 3 Rozsah:  $\langle -MAX, -0$  a  $\langle +0, +MAX \rangle$

## ① Dvojkový doplňkový kód

- ① Inverze bitu a přičtení jedničky
- ② Pouze jedna nula ( $11\dots 11$  je  $-1$ )
- ③ Nesymetrický rozsah:  $\langle -MAX - 1, -1 \rangle$  a  $\langle +0, +MAX \rangle$
- ④ skutečně používán v počítačích

## 1 Podle počtu bitů

- 1 Byte: 8 bitů, tj.  $\langle 0, 255 \rangle$  nebo  $\langle -128, 127 \rangle$
- 2 Půl slovo, 2 byte: 16 bitů, tj.  $\langle 0, 65535 \rangle$  nebo  $\langle -32768, 32767 \rangle$
- 3 Slovo, 4 byte: 32 bitů, tj. přibližně  $\langle -2 \cdot 10^9, 2 \cdot 10^9 \rangle$
- 4 Dvojslovo (nebo dlouhé slovo), 8 byte: 64 bitů, tj. přibližně  $\langle -9 \cdot 10^{18}, 9 \cdot 10^{18} \rangle$



- 1 Formát dle IEEE 754
- 2 Součásti
  - 1 Znaménko
  - 2 Mantisa (přímý kód, normalizace)
  - 3 Exponent (v kódu posunutě nuly)

- 1 Normalizace mantisy:
  - 1 Nejvyšší bit vždy jedna: 1.aaaaaa; 1 nezapíšeme
  - 2 Nejmenší kladné číslo:  $1.0 \times 2^{-2^{n-1}+1}$  ( $2^{-127}$  pro  $n = 8$ )
  - 3 Největší číslo:  $1.0 \times 2^{2^{n-1}}$  ( $2^{128}$ )
- 2 Exponent (n bitů, dvojková soustava)
  - 1 Přičteme  $2^{n-1} - 1$  ( $=127$  pro  $n = 8$ ), abychom získali správnou hodnotu pro uložení
  - 2 00000000 je  $-127$
  - 3 11111111 je  $128$
- 3 Zvláštní a nenormalizovaná čísla

- 1 Rozsah zobrazení: nejmenší záporné, největší kladné
- 2 Přesnost zobrazení: počet bitů mantisy + 1
- 3 Rozlišitelnost: nejmenší nenulové číslo
  - 1 Normalizované vs. nenormalizované ( $2^m$ krát menší,  $m$  počet bitů mantisy)

① Osmičková

①  $001\ 101\ 101\ 111_2 = 1557_8 = 879_{10}$

② Šestnáctková

①  $0011\ 0110\ 1111_2 = 36F_{16} = 879_{10}$

③ Používány především pro hutný zápis binárních čísel

- 1 Bootstrap loader
- 2 Spooling
  - 1 Nezávislé zavádění programu a jeho vykonávání
  - 2 Vyžaduje DMA (Direct Memory Access)
  - 3 Zavedlo *multiprogramování*
  - 4 Stále zpracování *dávek* (batch processing)
- 3 Timesharing
  - 1 Virtualizace počítače/CPU
  - 2 Zpracování *interaktivních* úloh
  - 3 Souvisí se zavedení *disků* (Direct Access Storage Device, DASD od IBM, 60tá léta)

## ① *Zkrásnění:*

### ① Zjednodušení práce s počítačem

- ① Práce s pamětí
- ② Práce se soubory
- ③ Přístup k perifériím

## *Sdílení:*

- 1 Zajistit sdílení vzácných zdrojů
- 2 Musí zajistit:
  - 1 Aby to vůbec fungovalo
  - 2 Aby to fungovalo účinně (využití, propustnost, rychlost odezvy)
  - 3 Aby to fungovalo správně
    - 1 Omezení následků chyb (avšak pozor na chyby v samotném operačním systému)
    - 2 Oprávnění k přístupu (autentizace a autorizace)

- 1 Periferie výrazně pomalejší než procesor
- 2 Příklad
  - 1 1 GHz Pentium IV:  $1 \cdot 10^9$  operací za sekundu
  - 2 Běžný disk: 10 ms pro přečtení 1 byte
  - 3 Poměr 1 : 10 000 000
  - 4 Stejně zpomalení člověka: 1 úhoz na klávesnici cca 20 dní.
- 3 Možné řešení: prokládání I/O a výpočtu
  - 1 Spuštění diskovou operaci  
Provádějí instrukce nad jinými daty  
(alespoň 1~milión instrukcí)  
Počkej na dokončení
  - 2 Příliš těžkopádné a složité



```
Proces 1 {  
Spust' diskovou operaci  
Počkej na dokončení  
Zpracuj získaná data  
}
```

```
Proces 2 {  
Nějaká jiná aplikace  
}
```

- 1 Přehlednější
- 2 OS musí „přepínat“ mezi procesy (*priorita*)

- 1 Většina paměti nevyužita
  - 1 Zpracování cyklu (zbytek programu)
  - 2 Zpracování konkrétních dat (ostatní neaktivní)
  - 3 Čekání na I/O
- 2 *Virtualizace* paměti
  - 1 Data a programy na disku
  - 2 Do paměti *na žádost*
  - 3 Umožňuje
    - 1 Každý program má „celou“ paměť
    - 2 Program může adresovat více jak rozsah fyzické paměti
- 3 Ochrana paměti

- 1 Procesy a jejich správa
- 2 Paměť a její správa
- 3 Periferie a jejich správa
- 4 Systém souborů
- 5 Ochrana a bezpečnost

- 1 Proces je abstrakce průchodu programem
  - 1 Sekvenční model: program = 1 proces
  - 2 Paralelní model: program > 1 proces
- 2 Proces má *interní stav*, charakterizovaný
  - 1 programovým čítačem (program counter)
  - 2 zásobníkem (volání funkcí a procedur)
  - 3 vlastní paměť pro data

- 1 Klasické (heavy-weight) procesy (např. UNIX)
  - 1 Všechna data privátní
  - 2 Sdílen pouze program (read-only)
- 2 *Lehké* (light-weight) procesy či Vlákna (threads)
  - 1 Minimum vlastní paměti
  - 2 Většina dat sdílena

## ① Vytvoření procesu

- ① `fork()` a jeho varianty
- ② *Přesná* kopie původního procesu
- ③ *Rodič* a *potomek*
- ④ První proces v OS vytvářen jinak (`init` v Unixu)

## ② Stavy

- ① Start/vytvoření, připraven (`ready`), běží (`running`), je blokováno (čeká), skončil