

# PB001: Úvod do informačních technologií

Luděk Matyska

Fakulta informatiky Masarykovy univerzity

podzim 2012



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

1 Přerušení

2 Systém souborů

# Přerušení

- Operační systémy obecně reagují na asynchronní události (events)
- *Přerušení*: mechanismus, jak přerušit vykonávanou práci na základě externí příčiny (nějaké události)

# Význam přerušení

- Podpora I/O
- Problém v programovém vybavení
  - Neautorizovaný přístup
  - Nelegální instrukce nebo operandy
- Požadavek počítačem řízeného systému
- Zásah operátora
- Výpadek hardware

# Příklady

- Přerušení od časovače (přepřelánování procesů, timeout, ...)
- Přerušení od periferie (klávesnice, myš, síťová karta, ...)
- Přerušení z procesoru (dělení nulou, chybná operace, ...)

# Principy přerušení

- *Přeruš*í běh aktuálního programu
  - Nutno uložit stav
  - a zapamatovat místo návratu
- Více zdrojů a příčin přerušení
  - Nutno rozlišit typy (příčinu) přerušení
  - Nutno zapamatovat zdroj přerušení

# Obsluha přerušeni

- Obsluha přerušeni realizována v kernelu
  - Zajištění serializace
  - Bezpečnost
- Vyvolá tzv. přepnutí kontextu

# Další vlastnosti

- Maskování přerušení
  - dočasné a trvalé
  - možná ztráta přerušení/události
- Priorita přerušení/obsluhy
  - Základní tři úrovně:
    - Nemaskovaná přerušení: vyšší priorita
    - Aktuálně zpracovávané přerušení
    - Maskovaná přerušení: nižší priorita



# Polling

- Polling = opakované dotazování (na stav/událost)
- Možná alternativa pro některá přerušení
  - Zaměstnává procesor
  - Může zůstat v uživatelském prostoru

# Systém souborů

- Základní funkce:
  - Vytvoření souboru
  - Čtení a psaní z/do souboru
  - Odstranění (smazání) souboru
  - Spuštění souboru (soubor=program)
- Podpora na úrovni operačního systému

# Struktura systému souborů

- Hierarchické systémy:
  - Kořen (root)
  - Adresáře jako speciální typ (meta)souboru: drží informace o souborech, nikoliv jejich vlastní data
- Databázové systémy:
  - Soubory (nebo jejich části) jako položka v databázi
  - Bohatší množina operací
  - Složitější implementace

# Struktura souborů

- Posloupnost bytů – vnitřní struktura pro OS neznáma
- Posloupnost záznamů (records)
- Strom – každý uzel má vlastní klíč

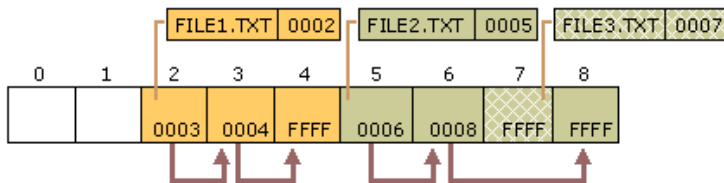
# Typ a přístup

- Typy souborů (v UNIXovém OS)
  - Řádné: běžné soubory
  - Adresáře: udržení hierarchické struktury
  - Speciální: přístup ke konkrétnímu zařízení (`/dev/mouse`, `/dev/audio`, `/dev/lp`); speciální `/proc` systém
  - Blokové: náhodný přístup na základní úrovni (`/dev/hd`, `/dev/kmem`)
- Přístupové metody; příklady:
  - Sekvenční
  - Náhodný (random)
  - Indexsekvenční (není v běžném UNIXu)

# Struktura na disku

- Možné typy
  - Souvislé
    - souvislé posloupnost bloků (složitá alokace, plýtvání místem)
  - Provázaný seznam:
    - každý blok odkazuje na další (může růst, vyšší režie – pro ukazatel, složitý náhodný přístup)
  - Indexové:
    - Např. FAT (File Allocation Table) v MS DOSu
    - Tabulka pro všechny bloky na disku
    - Provázány odkazem na další blok daného souboru
  - inodes

# Soubory ve fat



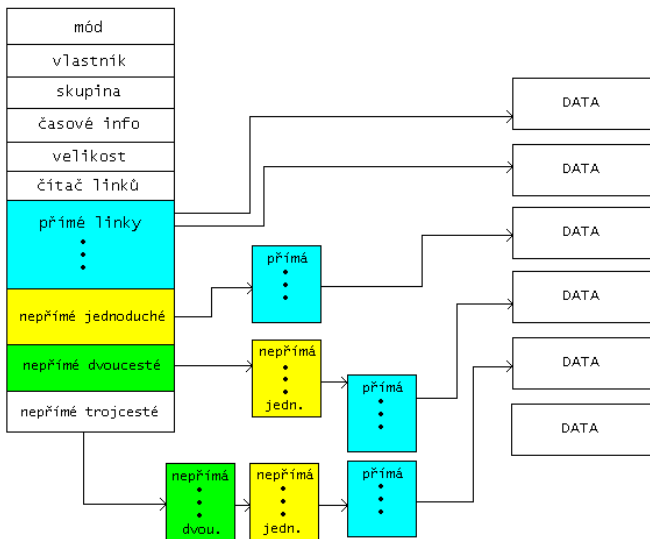
- file1.txt i file2.txt zabírají 3 clustery, file3.txt jeden
- file2.txt je fragmentován

# Struktura – inodes

- Podobné indexovému
- Pevná délka tabulky pro každý soubor
  - Kratší soubory adresovány přímo
  - Pro delší soubory alokována další tabulka
  - Tabulky provázány hierarchicky (1., 2. a 3. úroveň)
- Flexibilní, malá režie



# Struktura – inodes



# Volné bloky

- V tabulce
- Bitový vektor
- Provázaný seznam
- Většinou zpracovávány podle FCFS (First Come First Served)

# Vyrovnávací paměť

- Obecně přístup pro skrytí *zpoždění* (latence)
- Nejčastěji používané bloky/soubory uloženy v paměti
- Pouze pro čtení (snazší) nebo i pro zápis
- Problém: konzistence při přístupech/zápisech z více míst
- Základní typy
  - Write-through: okamžitě po zápisu i na disk
  - Write-back: až po určité době (30 s)