

Definice

1 Teoretické základy informatiky a matematika

2 Programové, výpočetní a informační systémy

2.1 Programovací jazyky

- **ordinální datové typy** – pro každý prvek je definován předchůdce a následovník
- **abstraktní datový typ** – implementačně nezávislá specifikace struktury dat s povolenými operacemi (např. zásobník, fronta, seznam, ...)
- **kompilátor** – nástroj sloužící k překladi vyššího programovacího jazyka do strojového kódu
- **interpret** – nástroj umožňující vykonávat zápis programu přímo ze zdrojového kódu

2.2 Objektově orientované programování

- **zapouzdření (encapsulation)** – princip říkájící, že cokoli co nemusí být viditelné, také být nemá (tzn. maximum `private`)
- **dědičnost (inheritance)** – umožňuje vytvořit podtřídu (podtyp) k existující třídě, který bude dědit její atributy a metody; jeho účel je poskytnout způsob modelování hierarchie objektů (generalizace-specializace)
- **podtypový polymorfismus, princip subsumpce** – způsob, jakým lze dosadit namísto očekávané třídy její libovolný podtyp, který se chová podle své implementace
- **událostmi řízené programování (event-drive programming)** – průchod programem řízen událostmi zaregistrovanými senzory (listenery), akcemi uživatelů nebo zprávami od ostatních procesů a vláken

2.3 Základní principy počítačů

-

2.4 Operační systémy

- **monolitické jádro** – veškerý kód jádra běží ve stejném paměťovém prostoru (kernel space) a jeho jednotlivé součásti jsou silně provázány
- **mikrojádru** – jádro velmi malé obsahující pouze nejzákladnější funkce (přerušování, procesy, ...) a vše ostatní (správa FS, ovladače zařízení, síťové služby) jsou nezávislé „moduly“ (tzn. `servery`), které běží v uživatelském prostoru (user space); jejich komunikace je zajišťována jádrem pomocí zasílání zpráv
- **hybridní jádro** – kombinuje vlastnosti monolitického jádra a mikrojádru
- **proces** – spuštěný program zavedený do operační paměti
- **process control block** – struktura určená k ukládání důležitých informací o procesu
- **plánovač (scheduler)** – program řídící přepínání kontextů
- **vlákno** – objekt vznikající v rámci procesu, který je viditelný pouze uvnitř a je charakterizován svým stavem; existují 2 typy – **user-level thread** a **kernel-level thread**
- **race condition** – chyba v systému nebo procesu ve kterém jsou nepředvídatelné výsledky a závisí na pořadí nebo načasování jednotlivých operací (obvykle při přístupu ke zdrojům)
- **kritická sekce** – segment kódu, kde se přistupuje ke sdíleným zdrojům
- **semafor** – způsob kooperace více procesů na základě zasílání jednoduchého signálu

- **monitor** – softwarový modul, který může být naráz používán pouze jedním procesem
- **kritická sekce** – segment kódu, kde se přistupuje ke sdíleným zdrojům
- **uváznutí (deadlock)** – vzniká za předpokladu, že existuje množina blokováných procesů, kde každý proces vlastní nějaký prostředek a čeká na zdroj držený jiným procesem z této množiny (tzn. žádný nemůže pokračovat)
- **dlouhodobý plánovač (scheduler)** – rozhoduje, které programy jsou přijaté systémem na spuštění
- **krátkodobý plánovač (dispatcher)** – rozhoduje, který proces poběží na právě uvolněném CPU
- **střednědobý plánovač** – rozhoduje, který proces se zařadí mezi „odložené“ procesy a naopak
- **FCFS (first come, first served)** – nepreemptivní; procesy řazeny ve FIFO (tzn. první na řadě je spuštěný)
- **round robin** – každý proces dostane na CPU malou jednotku času a po jejím uplynutí je nahrazen nejstarším procesem ve frontě
- **prioritní plánování** – preemptivní/nepreemptivní; každý proces má prioritní číslo a podle toho se přiděluje CPU; **problém stárnutí**
- **Shortest-job-first** – preemptivní/nepreemptivní (**Shortest-remaining-first**); vybírá se proces s nejkratším požadavkem na CPU (tzn. je nutné znát délku příštího požadavku pro každý proces)