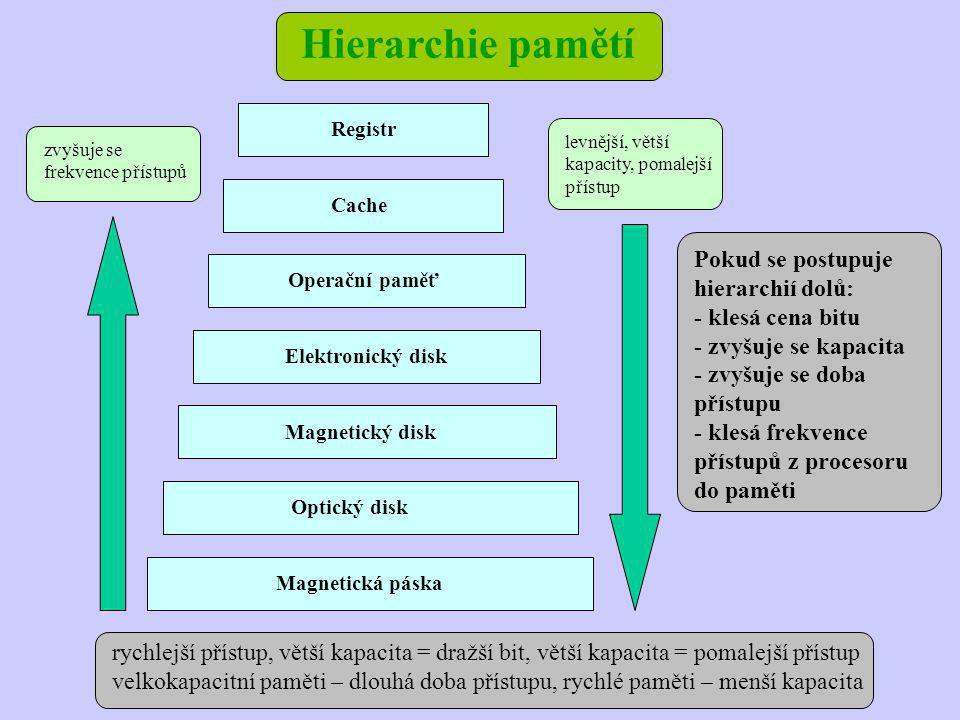
Paměti – obecný přehled



Paměť

* Součástka v počítači

Hierarchie

Polovodiče

* Energiticky závislé
  + Při vypojení, paměť se smaže
* Energeticky nezávislé
  + Při vypojení, paměť zůstane
* Ram
* SRAM
* DRAM
* Flash Paměť

NePolovodičové

* DVD
* CD
* Blue-Ray

Nezávislé

* EROM
* ROM
* Může se z nich jenom číst
  + Data jsou do nich zapsaná předem
  + Dřív BIOS

EEPROM

* Readonly
* Jdou smazat pomocí UV světla

Operační paměť RAM

* Data, uživána v běhu
* Hlavní pamět
* Protiklad HARDDISK
* Částo se mění, často se mažou
* Random asccess memorry
  + Random – může se k prvkům přistoupit jakýmoliv způsobem
    - Např. POLE – pomocí indexu, ví se kam se přistupuje

DRAM

* Nejrrozšířenější v Operačních Pamětích

SRAM

* Statická
* Hlavně na cache
* Dražší
* Rychlý přístup k datům

MRAM

* Magnoresistive
* Kombinace DRAM a SRAm

SSD

* Ukládá data systmaticky

HDD

* Kotouče
* Drobounka část ZMAGNETIZOVANÁ/NENÍ
  + 1/0
* Kmytání páčky

Virtuální paměť

* Není to paměť
* Způsob, jak ukládat paměť
* Hází data pomocí sběrnice
* Může snížit rychlost
  + RAM je rychlá ale HDD je pomalejší
* Možnost hodit data z RAM do HDD a následně je číst

Adresování a správa paměťi v JAVA



JAVA řeší za nás

* Spouští GC sama
  + Můžeme zavlat manualne

PAMĚŤ FUNGUJE JAK STACK

* Heap
* Stack
* Register

Method area

* Část paměti, kde se ukládají informace
* Binární kód
* String literals
* Stejné stringy budou uloženy na stejné paměti

Heap

* Ukládájí všechny objeky který se během programu vytvářejí
* Young generation
  + Volá se na nej MINOR GC
  + EDEN
    - Nově vytvořený objekty
    - Spustí se následně GB
      * Jestli mají přežít nebo ne
      * Posunou do Survivor space
      * Pokud přežijou 8 tak se posubou do Old Generation
  + Survivor space
* Old generation
  + Hozený objekty, který pro
  + Volá se na nej MAJOR GC
* Perm gen
  + Uchovává informace o metadatech
    - Data která odkazuji na další data
  + JAVA vyžaduje na popis tříd a metod
* Native meta stack
  + Spouští programy psaný v jiných jazycích
* Meta space
  + Nahradil Perm gen
  + Nemá žádný limit
  + Omezuje OUTOFMEMORYERROR

Stack (halda)

* Zásobník
* LIFO
* Jsou uloženy FRAME
  + Odkazují na různé proměnné a reference v HEAPU
  + REFERENČNÍ PROMĚNNÁ
    - ….
* Primitivní proměnný
  + Float
  + Int
  + …

PC Registy (Program counter)

* Instrukce jak mají být za sebou spuštěny příkazy
* Instrukce pro procesor
* Uloženy v pořadí
  + PODMÍNKA
  + CYKLUS
* Jak vypadají ty instrukce?

Garbage collecting

* Proces, co maže nepotřebný data
* Volá automaticky
* Maže prázdné
* Vytvoříte objekt
  + Bude na heapu
  + Objekt TŘÍDA = Další třída
  + Pokud 2 třída bude null
  + První se smaže GC protože bude odkazovat na NIC
* Každy objekt v EDENU se zkontroluje jestli je null nebo ne
  + Když není tak ho hodí do SURVIVOR
  + Když projde jedným GC
    - Znova GC
    - Projde – SURVIVOR 2
    - SURVIVOR 1 SE SMAŽE
    - Takto se opakuje v JAVA 8 krát aby opravdu přežil

Reference a pointery

* Při vytvoření – objekt na HEAPU
* Pomocí reference se tam dostanete
* Existuje reference na zdrojový kód