Lisp 9-1

Tento týdem je 2. test z Lispu !!

Ovládání paměti

U vyšších programovací jazyků zpravidla není potřeba explicitně uvolňovat pamět po již nepoužívaných objektech. Mluvíme o tzv. garbage collection. Lisp je historickým průkopníkem těchto metod.

- společným rysem různých garbage collection algoritmů je přidělování paměti, tak dlouho jak je to jenom možné (vyčerpání disponibilní paměti)
- v případě, že se žádost nepovede uspokojit dojde z určité kořenové množiny objektů k nalezení aktivních (dosažitelných) objektů a zbytek paměti je uvolněn pro další použití

Algoritmus Stop & Copy
- *používají se dvě oblasti* - nová oblast a kopírovací oblast (stejně velké)
- v okamžiku zaplnění nové oblasti se zavolá garbage collector a aktivní objekty překopíruje do kopírovací oblasti
- po tomto okamžiku kopírovací oblast označujeme jako novou oblast a probíhají v ní další alokace tak dlouho jak je to možné
- problém aby se opakovaně nekopírovali sdílené objekty dosažitelné z více kořenových míst -> dopředný ukazatel

výhody:
efektivní práce s virtuální pamětí - procházejí se pouze aktivní objekty, nikoliv celá virtuální pamět; procházení celé virtuální paměti by znamenalo značnou zátěž na stránkování

zcelování paměti - do kopírovací oblasti se data ukládají sekvenčně -> defragmentace paměti (snížení nároků na stránkování) ; protože procházení aktivních objektů je "do hloubky" dostanou se za sebe v paměti související objekty

nevýhody:
plýtvání pamětí - fakticky se vždy používá jenom polovina dostupné paměti
doba trvání - v mnoho případech je neúnostné aby se prostě zastavil výpočet a pokračoval až po dokončení kopírování
kopírování i statických objektů

Inkrementální Stop & Copy
- *používá 3 oblasti*, zátěž způsobená kopírováním je rozložena na celou dobu výpočtu

Metoda stratifikace objektů
- modifikace stop & copy vycházející z *různé doby životnosti jednotlivých objektů*
- vytvoří se několik přechodných oblastí (efemérní oblasti)
- do první oblasti jsou ukládány nově vzniklé objekty; v okamžiku zaplnění se provede stop & copy do druhé oblasti
- do první oblasti se opět ukládají nově vzniklé objekty
- dlouho žijící objekty postupně propadávají do hlubších oblastí až se nakonec dostanou do dynamické oblasti, ve které se provádí klasický stop & copy, pokud je to vůbec potřeba
- výhodou algoritmu je soustředění na krátce žijící objekty, dlouhodobě žijící objekty propadnou až do dynamické paměti, kde na stejném místě klidně vydrží až do konce výpočtu
- problém odhadnou vhodný počet přechodných oblastí

Lisp 9-2

Metoda značkování paměti

- angl. Mark & Sweep
- hlavní výhodou relativní jednoduchost a úplné využití přidělené paměti
- algoritmus má dvě fáze - *značkování a sbírání*
- během značkování se označí všechny aktivní cons buňky; začne se od buněk z kořenové oblasti a rekurzivně se projdou podřízené struktury
-fáze sbírání probíhá tak, že se projde celá pamět a neoznačené cons buňky se spojí do seznamu volné paměti

nevýhody:
nároky na zásobník - rekurivní provedení značkovací fáze vyžaduje místo na zásobníku v okamžiku kdy zrovna pamět není
nároky na značky - cons buňka musí mít flag na nastavení značky
sekvenční průchod celou pamětí při sbírání -> velká zátěž na stránkování u virtuální paměti
nedochází k defragmentaci paměti