STRATEGIJE OPTIMIZACIJE KODA

Stefan Bačević 1061/2014

Optimizacija koda

- Istorijski značajan problem 60-ih, 70-ih je uočeno koliko optimizacija koda šteti čitljivosti i održavanju koda, ponovo značajan problem 80-ih pojavom PC-ja i početkom 2000-ih pojavom telefona, PDA uređaja itd.
- Kontraverzna tema
- Performanse programa su labavo povezane sa brzinom izvršavanja koda

Kako poboljšati efikasnost programa?

- Zahtevi programa
- Dizajn programa
- Dizajn klasa i funkcija
- Interakcije sa operativnim sistemom
- Kompilacija koda
- Hardver
- Optimizacija koda

Uvod u optimizaciju koda

- Mane
- Prednosti
- Paretov princip – pravilo 80/20

Mitovi u vezi sa optimizacijom koda

- Kraci kod – brži program
- Neke operacije su verovatno brže od drugih
- Treba optimizovati u toku kodiranja
- Brzina programa je važna koliko i njegova tačnost

Izvori neefikasnosti

- Ulazno/Izlazne operacije
- Straničenje
- Sistemski pozivi
- Interpretirani jezici
- Greške

Operacije koje su obično relativno skupe

- Pozivi polimorfnih metoda
- Deljenje celih brojeva
- Deljenje brojeva sa pokretnim zarezom
- Narocito su skupe transcendentalne funkcije: sqrt(x), sin(x), log(x), e^x

Merenje vremena izvršavanja

- Merenje vremena je neophodno u optimizaciji koda
- Ne treba se oslanjati na iskustvo
- Merenje treba biti precizno
- Koristiti posebne alate za to ili meriti vreme tokom kog procesor izvršava samo taj program

Iteracije

- Jedna promena obično nije dovoljna
- Nikada ne menjati više stvari od jednom
- Meriti vreme nakon svake promene
- Većina promena neće skratiti vreme izvršavanja

Koraci koje treba slediti prilikom optimizacije koda

1. Razviti softver koristeći dobre prakse dizajna
2. Ako su performanse loše
   - Sačuvati poslednje “dobro” stanje
   - Meriti vreme radi nalaženja uskih grla
   - Utvrditi da li se problem može rešiti bez optimizacije koda, ako može vratiti se na korak 1
   - Optimizovati uska grla
   - Meriti vreme nakon svake promene
   - Ako promena ne poboljšava vreme izvršavanja vratiti se na poslednje “dobro” stanje
3. Ponoviti korak 2

Zaključak

- Brzina izvršavanja je samo jedna od karakteristika dobrog programa i obično nije najbitnija. Optimizovan kod nije najbitniji faktor koji utiče na performanse programa
- Merenje vremena i nalaženje uskih grla programa su najbitniji prilikom optimizacije koda
- Većina programa najveći deo vremena troši na izvršavanje jako malog dela koda
- Više promena daju optimizovan kod a ne samo jedna
- Najbolja priprema za optimizovanje koda je pisanje urednog i čitljivog koda

Hvala na pažnji!