

Specifikacija Zahteva za Projektovanje Raspoređivača Procesa

Uvod

Danas su računari koji izvršavaju samo jedan program retki, prosečan desktop računar istovremeno izvršava više desetina različitih procesa, međutim taj isti računar ima tek nekoliko desetina jezgara na kojima se ti procesi izvršavaju. Računar može da izvršava više procesa nego što ima jezgara iz dva razloga, prvi je činjenica da procesi često interaguju sa periferijama, a dok čekaju na odgovor od periferije računar može izvršavati neki drugi proces. Drugi razlog je da kada je broj procesa koji ne čekaju na periferiju i dalje veći od broja jezgara operativni sistem deli procesorsko vreme između procesa koji su spremni za izvršavanje, tako što izvršava jedan proces nekoliko milisekundi a potom izvršava neki drugi proces isto nekoliko milisekundi i tako u krug. Deo operativnog sistema koji određuje koji procesi se kada izvršavaju i na kom procesoru se naziva Raspoređivač Procesa.

Prva faza

Funkcionalni Zahtevi

1. Obezbediti interfejs raspoređivača prema ostatku sistema, ovaj interfejs enkapsulira odabran algoritam za raspoređivanje i pruža bar sledeće usluge:
 - Dodavanje procesa u listu spremnih.
 - Dohvatanje sledećeg procesa za izvršavanje iz liste spremnih.
 - Promenu algoritma za raspoređivanje i strukture podataka na koju se on oslanja, pretpostaviti da će se algoritam i struktura menjati samo kada nema procesa u listi spremnih.
2. Implementirati *FirstComeFirstServed* algoritam raspoređivanja i *ShortestJobFirst* tako da koriste usluge neke strukture podataka kroz definisan interfejs.
 - Voditi računa da dodavanje novih algoritama raspoređivanja, kao i strukture podataka na koju se oni oslanjaju treba biti što lakše.
3. Implementirati naivnu i efikasnu implementaciju strukture podataka koja ispunjava propisan interfejs.
4. Implementirati nit (može biti main nit) koja simulira procesor i reprezentaciju procesa koji naizmenično vrše obradu i čekaju na periferiju, čije se izvršavanje može simulirati implementiranim simulatorom procesora i koji se mogu raspoređivati implementiranim raspoređivačem.
 - Ovo će verovatno zahtevati implementaciju još jednog sloja koji izigrava operativni sistem i podržava pokretanje i završavanje procesa, vođenje računa o vremenu i simulaciju čekanja na periferiju.

- Obratiti pažnju da simulator procesora treba biti dizajniran tako da se može lako skalirati na više niti.

Druga faza

Funkcionalni Zahtevi

1. Omogućiti ispravno funkcionisanje raspoređivača ukoliko postoji više nezavisnih niti koje konkurentno simuliraju procesore, nije potrebno obezbediti ispravno funkcionisanje ukoliko se zahtev za promenu algoritma izvršava konkurentno sa ostalim zahtevima.
2. Omogućiti emulaciju preotimanja i implementirati varijacije implementiranih algoritama tako da koriste preotimanje.
 - Kako preotimanje nije moguće direktno implementirati, njega treba emulirati tako što dok proces vrši obradu sa vremena na vreme proverava da li je istekao vremenski kvant i ukoliko jeste prepušta procesor raspoređivaču.

Nefunkcionalni Zahtevi

1. Sistem treba da bude dizajniran da bude otporan na greške, uključujući neispravne argumente i pogrešnu upotrebu usluga (u granicama razuma, može se smatrati da jedino kod procesa koji se simuliraju i kod koji pokreće simulaciju dolaze od strane korisnika).
2. Sistem treba da vodi računa o performansama, tako da minimalno utiče na vreme izvršavanja programa koji ga koristi.
3. Sistem treba da bude dizajniran tako da bude lako proširiv i da omogući lako dodavanje algoritama za raspoređivanje i afinitet, kao i struktura podataka koje ti algoritmi koriste.