
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2 (13E113OS2, 13S113OS2)

Nastavnik: prof. dr Dragan Milićev

Školska godina: 2021/2022. (Zadatak važi počev od januarskog roka 2022.)

Projekat za domaći rad

- Projektni zadatak –

Verzija dokumenta: 1.0

Važne napomene: Pre čitanja ovog teksta, **obavezno** pročitati opšta pravila predmeta i pravila vezana za izradu domaćih zadataka! Pročitati potom ovaj tekst **u celini i pažljivo**, pre započinjanja realizacije ili traženja pomoći. Ukoliko u zadatku nešto nije dovoljno precizno definisano ili su postavljeni kontradiktorni zahtevi, student treba da uvede razumne pretpostavke, da ih temeljno obrazloži i da nastavi da izgrađuje preostali deo svog rešenja na temeljima uvedenih pretpostavki. Zahtevi su namerno nedovoljno detaljni, jer se od studenata očekuje kreativnost i profesionalni pristup u rešavanju praktičnih problema!

Uvod

Cilj ovog zadatka jeste implementacija raspoređivača procesa (engl. *scheduler*) za operativni sistem. Raspoređivač treba da obezbedi smeštanje procesa u skup spremnih procesa i odabir narednog procesa za izvršavanje. Operativni sistem ima podršku za izvršavanje procesa na jednom ili više procesora, shodno tome raspoređivač treba da obezbedi podršku za odabir procesa za izvršavanje na bilo kom procesoru.

Zadatak se sastoji od dva dela. Svaki deo definiše jedan algoritam koji raspoređivač treba da implementira. Svaki algoritam nosi po 15 poena. Za uspešnu odbranu projektnog zadatka student mora da uradi bar jedan deo, po slobodnom izboru.

Opšti zahtevi

Odnos projekta i korisničke aplikacije

Tražene podsisteme treba realizovati na jeziku C. Korisničku aplikaciju, koja sadrži test primere, prevesti nezavisno u konzolni program. U datoj aplikaciji biće prisutna i funkcija *main*. Korisnička aplikacija će napraviti određeni broj procesa korišćenjem sistemskog poziva *fork*. Korisnička aplikacija pristupa uslugama operativnog sistema jedino putem sistemskih poziva.

Odnos projekta i ostatka operativnog sistema

Dat je operativni sistem xv6 (programski kod se nalazi na sajtu predmeta). Zadatak studenta je da izmeni deo operativnog sistema xv6 tako da podrži raspoređivače čije je opis dat u ovom projektu. Izradom projekta se ni na koji način ne sme ugroziti ispravno funkcionisanje ostalih delova operativnog sistema. Svaki eventualni problem koji se pojavi po pokretanju projekta biće smatran kao greška pri izradi projekta. Deo koda koji je obezbeđen u okviru postavke projekta je pažljivo napisan, i ukoliko se koristi u skladu sa uputstvom za rad, ne može prouzrokovati nikakve probleme i greške pri izvršavanju.

Razvojno okruženje

xv6 operativni sistem se izvršava na emulatoru u okviru operativnog sistema domaćina (Linux x64). Virtuelna mašina sa instaliranim operativnim sistemom domaćinom, emulatorom i svim potrebnim alatima za prevođenje je data na sajtu predmeta. Ista ta virtuelna mašina će biti korišćena i na odbrani projekta. Predviđeno razvojno okruženje je CLion (studenti mogu nabaviti akademsku licencu na sajtu proizvođača sa studentskim email nalogom). Uputstvo za podešavanje projekta i testiranja je dato na sajtu predmeta. S obzirom da se razvija kernel operativnog sistema standardna biblioteka C jezika nije dostupna.

Raspoređivač

Uvod

Potrebno je realizovati dva algoritama za raspoređivanje procesa. Algoritam za raspoređivanje se bira pomoću sistemskog poziva. Broj i vrsta parametara sistemskog poziva se ostavljaju u nadležnosti samog rešenja. Pored postavljanja samog algoritma, potrebno je podesiti i parametra algoritma. Sistem može imati više procesora. Dati operativni sistem ima podrazumevani algoritam za raspoređivanje, ali on nije realizovan pomoću funkcija *get* i *set*. Podrazumevani kvant izvršavanja je do prvog sledećeg prekida od strane tajmera. Raspoređivač sadrži skup svih spremnih procesa u sistemu i vrši raspoređivanje za sve procesore.

Raspoređivač

Opis zadatka i funkcionalnosti

Potrebno je realizovati funkcije *get* i *put*, koja predstavlja interfejs raspoređivača. Sva mesta u kodu gde se vrši ubacivanje procesa u red spremnih treba zameniti pozivom *put* funkcije. Sva mesta u kodu gde se vrši dohvaćanje procesa iz reda spremnih treba zameniti pozivom *get* funkcije. Konkretan algoritam treba implementirati u proizvoljnim funkcijama. Te funkcije pozivaju funkcije *get* i *put* prema potrebi. Dodati mogućnost da se svakom procesu može dodeliti kvant izvršavanja izražen u boju prekida od tajmera (podrazumevano učestanost prekida od tajmera je 10 puta u 1s).

Potrebno je realizovati sledeće funkcionalnosti:

- Sistemski poziv za postavljanje algoritma raspoređivanja;
- Funkcije *get* i *put*;
- Algoritam raspoređivanja. Opisi potrebnih algoritama su dati u nastavku;
- Vremenski ograničeno izvršavanje procesa. Algoritam raspoređivanja određuje i postavlja taj vremenski kvant za svaki proces pojedinačno. Nije moguće da sama korisnička aplikacija menja taj kvant.

Procesi i procesori u operativnom sistemu xv6

Procesi i procesori su u operativnom sistemu opisani pomoću structura *proc* i *cpu*. Sva mesta za procese i svi procesori su definisani u statičkim nizovima *proc* i *cpus*. Broj procesa u sistemu je ograničen konstantom *NPROC*. Za potrebe rešenja dozvoljeno je dodavati polja u strukture koja predstavljaju procese i procesore. Operativni sistem je napravljen tako da obezbeđuje korektno izvršavanje na više procesora. Međusobno isključenje se radi pomoću zabrane prekida i uposlenog čekanja realizovanim instrukcijom iz grupe test-and-set.

Algoritmi za raspoređivanje

Potrebno je realizovati sledeća dva algoritma raspoređivanja: aproksimaciju Shortest-Job-First algoritma i uprošćenu verziju Completely Fair Scheduler algoritma. Izbeći koliko je moguće izgladnjivanje u bilo kom algoritmu.

Aproksimacija Shortest-Job-First algoritma

Algoritam može biti sa preuzimanjem ili bez preuzimanja. Procena narednog vremena izvršavanja se vrši eksponencijalnim usrednjavanjem. Vremenski kvant za izvršavanje je 0 (neograničen). Koeficijent za eksponencijalno usrednjavanje i varijantu algoritma je moguće postaviti prilikom zadavanja algoritma raspoređivanja.

Uprošćenu verziju Completely Fair Scheduler algoritma

Algoritam odabira sledeći proces za izvršavanje, koji je od početka svog tekućeg naleta izvršavanja (tj. od kako je došao u red spremnih npr. iz stanja suspenzije) imao najkraće ukupno vreme izvršavanja tokom tekućeg naleta izvršavanja (engl. CPU burst). Vremenski kvant za sledeće izvršavanje jednak je vremenu koje je proces proveo u skupu spremnih čekajući da dobije procesor podeljen sa brojem procesa.

Testovi

Javni testovi

Javni test-program služi da pomogne studentima da elementarno testiraju svoj projekat. Ovi testovi neće obavezno pokriti sve funkcionalnosti koje projekat treba da ima, ali će testirati većinu tih funkcionalnosti. Da bi se projekat uopšte odbranio, neophodno je da projekat sa javnim testom radi u potpunosti ispravno. Studentima se preporučuje da pored javnog testa naprave i svoje iscrpne testove koji će im pomoći da što bolje istestiraju svoj projekat.

Tajni testovi

Tajni testovi detaljnije testiraju sve zahtevane funkcionalnosti u različitim regularnim i neregularnim situacijama (greške u pozivu ili radu), i nisu unapred dostupni studentima.

Testovi performansi

Testovi performansi mere srednje vreme odziva procesa i ukupno vreme izvršavanja procesa. Ovi testovi nisu obavezni, i mogu, ali ne moraju, doneti dodatne bodove u predroku posle nastave za do 20 najboljih odbranijenih radova. Za potrebe povećanja performansi, student može obezbediti svoj algoritam za raspoređivanje, podesiti parametre već datih algoritama, itd. Kompleksnost algoritma se uzima u obzir prilikom ocenjivanja rada. Težiti da proces dobije isti procesor na kom se prethodno već izvršavao, kako bi se smanjilo kašnjenje zbog promašaja u procesorskom kešu koje bi postojalo kada bi se on izvršavao na drugom procesoru. Voditi računa o raspodeli opterećenja procesora (engl. *load balancing*).

Zaključak

Potrebno je realizovati opisane podsisteme prema datim zahtevima na jeziku C. Testiranje se vrši u laboratorijama katedre na računarima pod operativnim sistemom Windows 10 x64. Virtuelna mašina sa sajta predmeta biće dostupna za vreme odbrane.

Pravila za predaju projekta

Projekat se predaje isključivo kao jedna zip arhiva. U arhivu smestiti samo .c fajlove koji su rezultat izrade projekta. Opisani sadržaj ujedno treba da bude i jedini sadržaj arhive (arhiva ne sme sadržati ni izvršne fajlove, ni biblioteke, ni bilo kakve testove, niti bilo šta što iznad nije opisano). Projekat je moguće predati više puta, ali do trenutka koji će preko imejl liste biti objavljen za svaki ispitni rok i koji će uvek biti pre ispita, po pravili prvi radni dan pre ispita. Na serveru uvek ostaje samo poslednja predata verzija i ona će se koristiti na odbrani. Za izlazak na ispit neophodno je predati projekat (prijava ispita i položeni kolokvijumi su takođe preduslovi za izlazak na ispit). Nakon isteka roka za predaju, projektni zadaci se brišu sa servera, pa je u slučaju ponovnog izlaska na ispit potrebno ponovo postaviti ažurnu verziju projektnog zadatka.

Sajt za predaju projekta je https://rti.etf.bg.ac.rs/domaci/index.php?servis=os2_projekat

Zapisnik revizija

Ovaj zapisnik sadrži spisak izmena i dopuna ovog dokumenta po verzijama.

Verzija 1.0

Strana	Izmena