
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

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

Nastavnik: prof. dr Dragan Milićev

Asistent: Živojin Šuštran

Školska godina: 2024/2025. (Zadatak važi počev od januarskog roka 2025.)

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 dela sistema za organizaciju diskova. Sistem za organizaciju diskova treba da obezbedi implementaciju RAID struktura. U sistemu može da postoji više diskova koji se tretiraju kao jedan veliki logički disk sa ciljem poboljšanjem performansi i povećanjem otpornosti na otkaze. Treba obezbediti mogućnost zamene pokvarenog diska u toku rada sistema i oporavak od otkaza.

Zadatak se sastoji od dva dela. Prvi deo je obavezan i za uspešnu odbranu projektnog zadatka student mora da ga uradi. Svi algoritmi i optimizacije se ostavljaju u nadležnosti samog rešenja i biće razmatrani samo u okviru testiranja performansi sistema.

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 xv6¹ (prilagođen programski kod se nalazi na sajtu predmeta). Zadatak studenta je da izmeni deo operativnog sistema xv6 tako da podrži RAID strukture čiji su opisi dati 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. Dati kod dozvoljeno je menjati u onoj meri koja je potrebna za realizaciju traženih zahteva, ali nikako uklaňanjem već postojećih funkcionalnosti i menjanjem postojećih sistemskih poziva.

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.

¹ Detaljan opis xv6 operativnog sistema se može naći u knjizi na adresi (pristupljeno: 9.11.2023. godine)
<https://pdos.csail.mit.edu/6.S081/2023/xv6/book-riscv-rev3.pdf>

Prvi deo (20 poena)

Uvod

Operativni sistem može da se pokrene sa proizvoljnim brojem diskova i proizvoljnom veličinom. Prilikom pravljenja operativnog sistema u make fajlu se podešava broj diskova i njihova veličina. Maksimalni broj diskova u sistemu je 8 (ograničenje emulatora). Jedan disk xv6 koristi za smeštanje korisničkih programa (protokol za pristup disku nije potrebno menjati u rešenju), dok se ostali diskovi koriste za RAID strukturu. Svi diskovi koji se nalaze u RAID strukturi su iste veličine.

Disk za zamenu stranica

Opis datog interfejsa za pristup disku

Za potrebe zamene stranica studentima su date na raspolaganje funkcije pomoću kojih se pristupa diskovima za zamenu stranica. Operacije koje su obezbeđene su:

- Upis jednog (i samo jednog) bloka na disk; zadaje se redni broj diska (broj od 1 do 7), redni broj bloka (blokovi su numerisani počev od 0); podaci za upis se čitaju sa zadatog mesta u memoriji (memorija se ne dealocira u funkciji);
- Čitanje jednog (i samo jednog) bloka na disk; zadaje se redni broj diska (broj od 1 do 7), zadaje se redni broj bloka (blokovi su numerisani počev od 0); podaci se upisuju na zadato mesto u memoriji (memorija se ne alocira u funkciji);

Transfer se vrši korišćenjem prekida. U toku trajanja operacije prenosa prekidi mogu biti dozvoljeni. Operacija je blokirajuća dok se ne završi transfer.

Funkcije za pristup disku na jeziku C

Deklaracije datih funkcija date su u zaglavlju `defs.h`, a implementacija je u fajlu `virtio_disk.c`:

```
void write_block(int diskn, int blkn, uchar* data);  
void read_block(int diskn, int blkn, uchar* data);
```

RAID strukture

Sistem treba da sve diskove koji su na raspolaganju koristi za implementaciju RAID strukture. Potrebno je podržati sledeće varijante: RAID0, RAID1 i RAID0+1. RAID struktura treba da ima sledeće sistemske pozive:

- Inicijalizacija RAID strukture; zadaje se varijanta RAID strukture; ova operacija pored inicijalizacija struktura podataka u memorije, zapisuje i trajne informacije na diskovima tako da se između dva pokretanja operativnog sistema se svi podaci iz RAID strukture ne izgube;

- Upis jednog (i samo jednog) bloka na disk, redni broj bloka (blokovi su numerisani počev od 0); podaci za upis se čitaju sa zadatog mesta u memoriji (memorija se ne dealocira u funkciji);
- Čitanje jednog (i samo jednog) bloka na disk; zadaje se redni broj bloka (blokovi su numerisani počev od 0); podaci se upisuju na zadato mesto u memoriji (memorija se ne alocira u funkciji);
- Označavanje jednog diska kao pokvarenog; zadaje se redni broj diska (broj od 1 do 7); dok je disk pokvaren, sistem treba da omogući funkcionisanje RAID strukture ako je to moguće za dati tip RAID strukture;
- Označavanje jednog diska kao popravljenog; zadaje se redni broj diska (broj od 1 do 7); operacija ima efekta samo ako je popravljen disk koji je prethodno bio pokvaren; popravljen disk nema nikakve smislene podatke, pa sistem treba da povрати podatke koji su uništeni;
- Dohvatanje informacija o RAID strukturi; zadaju se lokacije na koje treba smesiti informacije o maksimalnom broju logičkih blokova koji su mogu koristi za smeštanje podataka (ne uključujući redundantne blokove), veličini bloka u bajtovima i broju diskova u RAID strukturi;
- Brisanje RAID strukture; nakon ove operacije RAID struktura se više ne može koristi i podaci na diskovima ne treba više da budu validni;

Svi sistemski pozivi vraćaju vrednost 0 u slučaju uspeha, a negativnu vrednost u slučaju greške.

Sistemski pozivi za pristup RAID strukturi na jeziku C

Interfejs sistemskih je:

```
enum RAID_TYPE {RAID0, RAID1, RAID0_1, RAID4, RAID5};

int init_raid(enum RAID_TYPE raid);

int read_raid(int blkn, uchar* data);

int write_raid(int blkn, uchar* data);

int disk_fail_raid(int diskn);

int disk_repaired_raid(int diskn);

int info_raid(uint *blkn, uint *blks, uint *diskn);

int destroy_raid();
```

Drugi deo (10 poena)

Uvod

Cilj drugog dela projekta jeste da se obezbedi dodatne varijante RAID strukture i funkcionisanje RAID strukture kojoj pristupaju konkurentni procesi. Rešenje treba da uključi RAID4 i RAID5 varijante. Svi sistemski pozivi vezani za RAID strukturu treba da budu bezbedni za pristup iz više konkurentnih 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 testiraju svoj projekat. Programi koji dolaze uz xv6 operativni sistem se takođe smatraju kao javni.

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 vreme izvršavanja procesa i efikasnost u korišćenju resursa. 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, studentima je dozvoljeno izmena datog koda, kao i optimizacija/modifikacija traženih algoritama. Npr.: omogućavanje paralelnih zahteva diskovima od strane jednog procesa, restauriranje podataka na popravljenom disku asinhrono (kada se diskovi ne koriste značajno), implementacija kodova za oporavak koji su bolji od bita parnosti (RAID6), itd.

Zaključak

Potrebno je realizovati opisane podsisteme prema datim zahtevima na jeziku C. Testiranje se vrši u laboratorijama katedre na računarima pod operativnim sistemom Windows 11 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 fajlove sa kodom (.c, .cpp, .h i slične) 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, pogotovu ne sme sadržati repozitorijum koda. 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 pravilu 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