
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 1 (SI2OS1, IR2OS1)

Nastavnik: prof. dr Dragan Milićev

Odsek: Softversko inženjerstvo, Računarska tehnika i informatika

Kolokvijum: Drugi, septembar 2016.

Datum: 4.9.2015.

Drugi kolokvijum iz Operativnih sistema 1

Kandidat: _____

Broj indeksa: _____ *E-mail:* _____

Kolokvijum traje 1,5 sat. Dozvoljeno je korišćenje literature.

Zadatak 1 _____/10

Zadatak 3 _____/10

Zadatak 2 _____/10

Ukupno: _____/30 = _____% = _____/15

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je uokviri (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Ocenjivanje unutar potpitanja je po sistemu "sve ili ništa", odnosno nema parcijalnih poena. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**.

1. (10 poena)

Školsko jezgro treba proširiti podrškom za slanje i prijem poruka između niti, implementacijom sledeće dve operacije klase `Thread`:

- `void Thread::send(char* message):` pozivajuća nit šalje poruku datoj niti (`this`); ukoliko je ovoj niti već stigla neka poruka koju ona nije preuzela, pozivajuća nit se suspenduje dok se prethodna poruka ne preuzme i tek onda ostavlja poruku i nastavlja izvršavanje;
- `static char* Thread::receive():` pozivajuća nit preuzima poruku koja joj je poslata; ukoliko poruke nema, pozivajuća nit se suspenduje dok poruka ne stigne.

Rešenje:

2. (10 poena)

Potrebno je implementirati proceduru `resolveSymbols` koja se koristi u drugom prolazu jednog linkera. Ova procedura obrađuje jedan ulazni `.obj` fajl i treba da razreši adresna polja mašinskih instrukcija koja koriste simbole koje taj fajl uvozi. Ulazni i izlazni fajl su memorijski preslikani, tehnikom virtuelne memorije, tako da se njihov sadržaj može jednostavno posmatrati kao sadržaj memorije procesa. Na početak memorijski preslikanog sadržaja ulaznog `.obj` fajla ukazuje prvi, a na početak tog prepisanog sadržaja u izlaznom (`.exe`) fajlu ukazuje drugi argument ove procedure; pre poziva ove procedure, linker je već prepisao sadržaj binarnog prevedenog koda (bez zaglavlja) svih ulaznih `.obj` fajlova u sadržaj izlaznog fajla.

Na samom početku ulaznog fajla nalazi se zaglavlje. Svi pomeraji (*offset*, odnosno relativne adrese) u njemu izraženi su u jedinicama `sizeof(char)==1`, a veličine su `unsigned long` (skraćeno `ulong`). Sadržaj početka zaglavlja je, redom, sledeći:

- jedan `ulong` koji sadrži pomeraj početka binarnog prevedenog koda u `.obj` fajlu u odnosu na početak celog sadržaja tog fajla (zapravo sadrži veličinu celog zaglavlja iza koga sledi binarni prevedeni kod);
- jedan `ulong` koji sadrži ukupan broj simbola koji se uvoze (n);
- n redom poređanih parova: ime simbola koji se uvozi (niz znakova proizvoljne dužine, završen znakom `'\0'`), iza koga sledi jedan `ulong` koji predstavlja pomeraj prvog nerazrešenog adresnog polja u mašinskoj instrukciji koje treba da sadrži vrednost razrešene adrese tog simbola; takva polja su dalje ulančana u jednostruku listu, tako da svako polje sadrži pomeraj narednog takvog polja za isti simbol, s tim da vrednost pomeraja 0 označava kraj liste (poslednje takvo polje za taj simbol); ovi pomeraji su relativni u odnosu na početak prevedenog binarnog koda unutar sadržaja `.obj` fajla.

Linker poseduje tabelu simbola čija operacija:

```
ulong SymbolTable::resolveSymbol(char* symbol);
```

vraća pomeraj (relativnu adresu) u odnosu na početak izlaznog fajla (`.exe`) u koji se dati simbol prevodi, ukoliko on postoji u tabeli, a 0 ako ga nema. Grešku nedefinisanog simbola treba obraditi pozivom funkcije:

```
int errorSymbolUndefined(char* symbol);
```

Ova funkcija ispisuje korisniku poruku o nedefinisanom datom simbolu i vraća -1, što u tom slučaju treba da vrati i funkcija `resolveSymbols`. U slučaju uspeha, funkcija `resolveSymbols` treba da vrati 0.

```
int resolveSymbols (char* inputObj, char* output);
```

Rešenje:

3. (10 poena)

Virtuelni adresni prostor nekog sistema je 16EB (eksabajt, $1E=2^{60}$) i organizovan je stranično, adresibilna jedinica je bajt, a stranica je veličine 64KB. Fizički adresni prostor je veličine 4TB (terabajt). PMT (*page map table*) je organizovana u dva nivoa, s tim da su i broj ulaza, kao i širina ulaza u PMT prvog i drugog nivoa isti (PMT oba nivoa su iste veličine). PMT oba nivoa smeštaju se u memoriju uvek poravnate na fizički okvir, odnosno uvek počinju na početku okvira. Zbog toga se u ulazu prvog nivoa čuva samo broj okvira u kom počinje PMT drugog nivoa, dok se preostali biti do celog broja bajtova u ulazu ne koriste; vrednost 0 u svim bitima označava hardveru da preslikavanje nije dozvoljeno ili moguće. U jednom ulazu PMT drugog nivoa čuva se broj okvira u koji se stranica preslikava i još 2 najniža bita koja koduju prava pristupa (00 – nedozvoljen pristup, stranica nije u memoriji, 01 – dozvoljeno samo izvršavanje instrukcije, 10 – dozvoljeno samo čitanje podataka, 11 – dozvoljeno i čitanje i upis podataka), dok se ostali biti ne koriste. Jedan ulaz u PMT prvog i drugog nivoa zauzima minimalan, ali ceo broj bajtova. Kada sistem kreira nov proces, ne učitava inicijalno ni jednu njegovu stranicu, niti alokira ijednu PMT drugog nivoa, već samo alokira PMT prvog nivoa, čije sve ulaze inicijalizuje nulama. Stranice se potom dohvataju na zahtev, tokom izvršavanja procesa, kada se po potrebi alociraju i PMT drugog nivoa.

a)(3) Prikazati logičku strukturu virtuelne i fizičke adrese i označiti veličinu svakog polja.

Odgovor:

b)(7) Implementirati sledeću funkciju:

```
void releasePMTEntry (unsigned* pmt1, unsigned long page);
```

Ovu funkciju poziva kod kernela kada iz memorije izbacuje datu stranicu `page` procesa čija je PMT prvog nivoa zadata prvim argumentom. Ova funkcija treba da ažurira odgovarajuće ulaze u PMT (po potrebi, oba nivoa), s tim što treba i da dealocira PMT drugog nivoa, ako ona više nije potrebna. Na raspolaganju je interna funkcija kernela `dealloc_pmt(unsigned* pmt)` koja dealocira jednu PMT u memoriji. Ignorirati sve eventualne greške.

Rešenje: