
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2 (SI3OS2, IR3OS2)

Nastavnik: prof. dr Dragan Milićev

Odsek: Softversko inženjerstvo, Računarska tehnika i informatika

Kolokvijum: Treći, januar 2013.

Datum: 22.1.2013.

Treći kolokvijum iz Operativnih sistema 2

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 = _____%

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) Operativni sistem Windows

U nastavku je data ilustracija upotrebe objekta *critical section* iz dokumentacije za Win32 API. Na jeziku C++ implementirati klasu `Mutex` koja obezbeđuje objektno orijentisani „omotač“ oko ovih sistemskih poziva i realizuje apstrakciju binarnog semafora za međusobno isključenje niti.

```
// Global variable
CRITICAL_SECTION CriticalSection;

int main( void )
{
    ...

    // Initialize the critical section one time only.
    if (!InitializeCriticalSectionAndSpinCount(&CriticalSection,
        0x00000400) )
        return;
    ...

    // Release resources used by the critical section object.
    DeleteCriticalSection(&CriticalSection);
}

DWORD WINAPI ThreadProc( LPVOID lpParameter )
{
    ...

    // Request ownership of the critical section.
    EnterCriticalSection(&CriticalSection);

    // Access the shared resource.

    // Release ownership of the critical section.
    LeaveCriticalSection(&CriticalSection);

    ...
return 1;
}
```

Rešenje:

2. (10 poena) Operativni sistem Linux

Napisati *shell script* koji dohvata fajlove sa interneta određene ekstenzije u tekući direktorijum i ispisuje njihove nazive na standardnom izlazu. Prvi argument skripte je internet adresa veb stranice u kojoj se nalaze linkovi ka fajlovima koje treba dohvatiti. Drugi argument skripte je ekstenzija fajlova koji se dohvataju. Stranica je tekstualni fajl u HTML formatu. Svaki link ka fajlu predstavlja internet adresu zapisanu na sledeći način:

```
<a href="internet_adresa_fajla">Naslov</a>.
```

Primer:

```
<a href="http://www.etf.rs/diplome.pdf">Diplome</a>
```

U slučaju neodgovarajućeg broja argumenata ili nemogućnosti da se dohvati zadata stranica prijaviti grešku i prekinuti izvršavanje skripte. Nakon završetka izvršavanja skripte potrebno je ostaviti sistem u neizmenjenom stanju.

Program *wget* služi za dohvatanje fajla sa zadate adrese u tekući direktorijum. Njemu se kao parametar prosleđuje internet adresa fajla (smatrati da adresa ne sadrži razmake). Ovoj naredbi može se zadati opcija *-O* nakon koje sledi ime koje se dodeljuje određenom fajlu. Ukoliko se ova opcija ne koristi, ime fajla će biti nepromenjeno, tj. isto kao izvorišno.

Rešenje:

3. (10 poena) Operativni sistem Linux

Posmatra se sistem od tri procesa koji predstavljaju pušače i jedne klase koja predstavlja agenta. Svaki pušač ciklično zavija cigaretu i puši je. Za zavijanje cigarete potrebna su tri sastojka: duvan, papir i šibica. Jedan pušač ima samo duvan, drugi papir, a treći šibice. Agent ima neograničene zalihe sva tri sastojka. Agent postavlja na sto dva sastojka izabrana slučajno. Pušač koji poseduje treći potreban sastojak može tada da uzme ova dva, zavije cigaretu i puši. Kada je taj pušač popužio svoju cigaretu, on javlja agentu da može da postavi nova dva sastojka, a ciklus se potom ponavlja. Primer implementacije navedena tri procesa dat je u nastavku:

```
int main() {
    key_t key = ...;
    Agent a(key);

    for(int i=0;i<3;i++){
        if(fork()==0){
            while(1){
                switch (i) {
                    case 0: a.takePaperAndMatch(); break;
                    case 1: a.takeTobaccoAndMatch(); break;
                    case 2: a.takeTobaccoAndPaper(); break;
                    default: exit(1); break;
                }

                //consume
                sleep(1);

                a.finishedSmoking();

                //wait
                sleep(1);
            }
        }
    }
    wait(0);
    return 0;
}
```

Koristeći za potrebe međuprocenske komunikacije i sinhronizacije isključivo semafore operativnog sistema Linux realizovati na jeziku C++ klasu koji predstavlja agenta. `Agent(key_t key)` prima kao argument vrednost ključa koji jedinstveno određuje skup semafora koji se koristi. Na raspolaganju je funkcija `randNum()` koja nasumično vraća celobrojnu vrednost u opsegu $[0,2]$ različitu od prethodnog poziva. Nije potrebno proveravati uspešnost izvršavanja operacija nad semaforima.

Rešenje: