
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Operativni sistemi 2
Nastavnik: prof. dr Dragan Milićev
Odsek: Softversko inženjerstvo, Računarska tehnika i informatika
Kolokvijum: Treći, januar 2026.
Datum: 20. 2. 2026.

Treći kolokvijum iz Operativnih sistema 2

Kandidat: _____

Broj indeksa: _____ *E-mail:* _____

Kolokvijum traje 90 minuta. 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) Arhitektura operativnih sistema

Data je delimična implementacija klase koja predstavlja uslovnu promenljivu koja se isključivo koristi unutar monitora. Međusobno isključenje u monitoru se radi pomoću jednog binarnog semafora. Napisati implementaciju metode `wait` koja nedostaje. Monitori i uslovne promenljive su realizovane pomoću Win32 API-ja. Deo dokumentacije Win32 API-ja je dat u nastavku.

```
class Condition {
public:
    Condition() : waiting_(0) {
        cond_ = CreateSemaphore(nullptr, 0, 0x7fffffff, nullptr);
    }

    ~Condition() {
        if (cond_) CloseHandle(cond_);
    }

    void wait(HANDLE mutex);

    void signal() {
        if (waiting_ > 0) {
            ReleaseSemaphore(cond_, 1, nullptr);
        }
    }

private:
    HANDLE cond_;
    int waiting_;
};
```

Syntax

```
DWORD WaitForSingleObject(
    [in] HANDLE hHandle,
    [in] DWORD dwMilliseconds
);
```

Waits until the specified object is in the signaled state or the time-out interval elapses.

Parameters

[in] hHandle

A handle to the object. For a list of the object types whose handles can be specified, see the following Remarks section.

If this handle is closed while the wait is still pending, the function's behavior is undefined.

[in] dwMilliseconds

The time-out interval, in milliseconds. If a nonzero value is specified, the function waits until the object is signaled or the interval elapses. If dwMilliseconds is zero, the function does not enter a wait state if the object is not signaled; it always returns immediately.

If dwMilliseconds is INFINITE, the function will return only when the object is signaled.

Syntax

```
DWORD SignalObjectAndWait(  
    [in] HANDLE hObjectToSignal,  
    [in] HANDLE hObjectToWaitOn,  
    [in] DWORD  dwMilliseconds,  
    [in] BOOL   bAlertable  
);
```

Signals one object and waits on another object as a single operation.

Parameters

[in] hObjectToSignal

A handle to the object to be signaled. This object can be a semaphore, a mutex, or an event.

[in] hObjectToWaitOn

A handle to the object to wait on.

[in] dwMilliseconds

Same as in WaitForSingleObject

[in] bAlertable

If this parameter is *TRUE*, the function returns when the system queues an I/O completion routine or APC function, and the thread calls the function. If *FALSE*, the function does not return, and the thread does not call the completion routine or APC function.

Rešenje:

2. (10 poena) Operativni sistem Linux

Data je delimično napisana skripta. Skripta treba da briše fajlove i direktorijume određenog korisnika. Prvi argument skripte je ime korisnika čije fajlove treba obrisati. Ostali argumenti su direktorijumi u kojima se vrši brisanje. Brisanje se vrši samo direktnih potomaka zadatih direktorijuma. Ispravan poziv skripte ima bar dva argumenta.

Na mestima označenim rednim brojevima napisati deo koji nedostaje da bi skripta radila zadati posao.

```
#!/bin/bash

if [[ $# -lt 2 ]]; then
    echo "Usage: $0 <owner> <dir1> [dir2 ...]"
    (1*)
fi

owner="$1"
(2*)

for dir in "$@"; do
    if [ (3*) ]; then
        echo "Skipping (not a directory): $dir"
        continue
    fi

    OLD_IFS="$IFS"
    IFS=$'\n'
    for item in $(ls -l "$dir"); do
        current_owner=$(echo "$item" | (4*) )
        path=$dir/$(echo "$item" | sed 's:.* \([^ ]*\)$:\1:')

        if [ "$current_owner" == "$owner" ]; then
            echo "Deleting: $path"
            rm (5*) "$path"
        fi
    done
    IFS="$OLD_IFS"
done

echo "Done."
```

Rešenje:

(1*): _____

(2*): _____

(3*): _____

(4*): _____

(5*): _____

3. (10 poena) Operativni sistem Linux

Napisati rešenje problema čitalaca i pisaca korišćenjem prosleđivanje poruka u operativnom sistemu Linux. Čitaoci i pisci su zasebni procesi. Smatrati da je kod za te procese već napisan i nije ga potrebno pisati. Pored njih postoji i proces kontroler od koga čitaoci i pisci traže dozvolu da obave svoju operaciju i kome javljaju da su svoju operaciju završili. Dozvoljeno je da rešenje sadrži izglednjivanje pisaca.

Rešenje: