

Rešenja trećeg kolokvijuma iz Operativnih sistema 2, Septembar 2013.

1. (10 poena)

a)(5) Stavljanje novog zahteva: $O(1)$, jer se zahtev stavlja na početak liste (promena nekoliko pokazivača, nezavisno od veličine liste). Izbor zahteva: $O(n)$, linearna pretraga cele liste da se pronađe zahtev na minimalnom rastojanju od upravo opsluženog.

b)(5) Na primer, kreirati niz (statički alociran, veličine jednake broju cilindara na disku C) u kome je svaki element k glava liste zahteva koji se odnose na cilindar k . Novi zahtev koji se odnosi na cilindar k se stavlja na početak liste u ulazu k . Pretraga za najbližim zahtevom ide iterativno kroz niz, počev od ulaza koji odgovara cilindru upravo opsluženog zahteva k i proverava najpre susedne ulaze ($k-1$ i $k+1$), pa onda sledeće ($k-2$ i $k+2$), itd. sve dok ne naiđe na neprazan ulaz. Ovaj postupak ima složenost $O(C)$, ali je složenosti $O(1)$ u odnosu na broj zahteva n .

2. (10 poena)

```
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Nedovoljan broj parametara"
    exit 1
fi
dir=$1
string=$2
if [ -d "$dir" ];then
    IFS_old=$IFS
    IFS=$'\n'
    for i in $(find "$dir");do
        if [ -f "$i" ];then
            grep $string "$i" > /dev/null && echo "$i"
        fi
    done
    IFS=$IFS_old
else
    echo "Prvi parametar nije direktorijum"
    exit 2
fi
```

3. (10 poena)

```
#include <stdio.h>
#include <sys/shm.h>
#include <sys/stat.h>

int main (int argc, const char* argv[])
{
    int shmkey, N, shmsize;
    if ( argc > 2 ) {
        shmkey = atoi( argv[1] );
        N = atoi( argv[2] );
    }
    else return -1;

    shmsize = N*N*sizeof(int);

    int shmid1 = shmget(shmkey,shmsize, IPC_CREAT | S_IRUSR);
    int shmid2 = shmget(shmkey+1,shmsize, IPC_CREAT | S_IRUSR);
    int shmid3 = shmget(shmkey+2,shmsize, IPC_CREAT | S_IRUSR | S_IWUSR);

    for (i = 0; i < N; i++)
        for(j=0;j<N;j++)
            if ( fork() == 0 ) {

                int * a = (int*)shmat(shmid1,0,0);
                int * b = (int*)shmat(shmid2,0,0);
                int * c = (int*)shmat(shmid3,0,0);

                c[i*N+j]=0;
                for(k=0;k<N;k++)
                    c[i*N+j] += a[i*N+k]*b[k*N+j];

                shmdt((void*)a);
                shmdt((void*)b);
                shmdt((void*)c);
                exit(0);
            }

    wait(0);
    return 0;
}
```