

Strukture podataka i algoritmi 1

(zadatak - max 30 poena)

Septembar, 2020

Zbog nestanka struje svi IMI severi su restartovani. Za svaki IMI server se zna njegov ID (ceo broj) i broj procesora (ceo broj). U datoteci su zabeleženi sve promene koje su se dešavale na serverima 24 sata nakon restarta i te informacije su date hronološki. Za svaku promenu situacije na serveru je zabeleženo vreme promene (ceo broj, broj sekundi proteklih od restartovanja servera), koliko procesa je napravilo promenu u tom trenutku i za svaki proces njegov ID (ceo broj) i broj procesora koji on zahteva (ceo broj). S obzirom da se promene nisu znale unapred, prvo pojavljivanje procesa u spisku promena označava da je proces pokrenut. Tada se traži server koji ima dovoljno procesora na kojima proces može da se pokrene (jedan proces ne može da se pokrene na više servera) i pri tome ima najmanje procesa na čekanju. Može se desiti da su svi serveri zauzeti i proces se prijavljuje na čekanje na tom serveru. Svako naredno pojavljivanje istog procesa u listi promena označava da je deo izvršavanja ili čitavo izvršavanje procesa završeno i da su oslobođeni procesori ili deo procesora koji je taj proces zauzimao. Pretpostaviti da će se proces automatski pokrenuti kada se na serveru na koji je prijavljen oslobodi dovoljan broj procesora za pokretanje procesa, kao i da se proces neće odjaviti više puta nego što je moguće. Kada proces kompletno završi sa svojim izvršavanjem, on se briše iz spiska aktivnih procesa. Na jednom serveru se u jednom trenutku može izvršavati više procesa ukoliko to broj procesora dozvoljava, ali se procesi na čekanju ne mogu pokretati preko reda. Ukoliko se u spisku promena u nekom trenutku unese ID procesa 0 sa 0 zahtevani procesora, to je signal da na izlazu treba prikazati trenutno stanje na serverima, tj. za svaki server koji procesi se izvršavaju i/ili su na čekanju i na koliko procesora. Napisati program koji će simulirati rad IMI servera u posmatranom periodu.

(3 poena)

Za rešavanje problema napisati sledeće funkcije:

a) Definirati sve potrebne složene tipove podataka neophodne za rešavanje opisanog problema.

(3 poena, obavezno)

b) Napisati funkciju **UcitajServere** koja iz datoteke *Serveri.txt* učitava podatke o serverima i formira listu/niz servera.

(3 poena, obavezno)

c) Napisati funkciju **NadjiServera** koja za dati ID servera vraća pokazivač na određeni server.

(3 poena, obavezno)

d) Napisati funkciju **UcitajProcese** koja iz datoteke *Procesi.txt* učitava podatke o procesima i formira listu/niz grupa.

(4 poena)

e) Napisati funkciju **ZauzimanjeServera** koja u nizu/listi servera pronalazi najmanje opterećeni server sa dovoljnim brojem procesora.

(4+1 poena)

f) Napisati funkciju **BrisanjeProcesa** koja dati proces uklanja iz spiska procesa ukoliko njegovo izvršavanje potpuno završeno.

(4+1 poena)

g) Napisati funkciju **Ispis** koja ispisuje trenutnu situaciju na serverima.

(4 poena)

Pri učitavanju podataka podrazumevati da su svi podaci korektno zadati

Dozvoljeno je proširivanje struktura i definisanje novih, kao i definisanje dodatnih funkcija.

Zadatak se može rešavati korišćenjem povezanih lista, nizova ili kombinacijom.

Zadatak rešiti bez korišćenja globalnih promenljivih i bez unapred definisanih dužina korišćenih nizova.

Strukture podataka i algoritmi 1
Test – max 20 poena

Septembar, 2020

Ime i prezime	Broj indeksa	Broj poena
---------------	--------------	------------

1. (1.5) Šta je rezultat sledećih kodova?

```
int a = 38;
printf("%d", a & 11);
```

2

```
int a = 60;
printf("%d", a | 12)
```

60

```
int a = 45;
printf("%d", ~(~a) & (~16));
```

61

2. (2.0) Sortiraj niz brojeva 17 3 22 13 78 30 54 47 6 28 u rastućem poretку koristeći Merge sort i ispisati svaki korak prilikom sortiranja brojeva.

17	3	22	13	78	30	54	47	6	28
3	17	22	13	78	30	54	47	6	28
3	13	17	22	78	30	54	47	6	28
3	13	17	22	30	54	78	47	6	28
3	13	17	22	30	47	54	78	6	28
3	6	13	17	22	28	30	47	54	78

3. (2.5) Data je struktura:

```
struct node{
    float x;
    struct node* next;
};
```

```
struct node* head;
```

Napisati niz komandi kojima se formira lista brojeva unošenjem sa tastature i dodavanjem na početak novog broja, sve dok se ne unese negativan broj.

```
struct node *head=NULL;
struct node *novi;
float x;
scanf("%f",&x);
while(x>=0){
    novi=(struct node*)malloc(sizeof(struct node));
    novi->x = x;
    novi->next = head;
    head = novi;
    scanf("%f",&x);
}
```

4. (2.5) Za strukturu iz zadatka 3 napisati funkciju koja za dati pokazivač na početak liste i broj k briše element koji su bar k puta veći od vrednosti koja se nalazi u glavi liste.

```
void obrisi(struct node *p, float k){
    struct node *temp, *temp2;
    temp = p;
    while(temp->next){
        if(temp->next->x >= k * p->x){
            temp2 = temp->next;
            temp->next = temp2->next;
            free(temp2);
        }
        temp = temp->next;
    }
}
```

5. (1.0) Napisati uslovni izraz koji odgovara sledećem kodu

```
if(a%4==0)
    b = 7 * a;
else if(!a)
    b = pow(a, 2);
else
    b = a / 3;
```

```
b = (!(a%4)) ? a*7 : (!a ? pow(a,2) : a/3 ) ;
```

6. (2.5) U programskom jeziku C napisati funkciju Razlika koja vrši operaciju razlike za dva prosleđena stringa, gde je rezultat string koji sadrži karaktere koji se nalaze u prvom, a ne nalaze u drugom stringu. Na primer: Razlika("papirologija", "papir") -> "ologij"

```
char* razlika(char *a, char *b){
    char *s;
    int i=0,j=0,k;
    s = (char*)malloc(strlen(a));
    while(a[i]!='\0'){
        k=0;
        while(b[k]!='\0' && b[k]!=a[i]) k++;
        if(k == strlen(b))
            s[j++]=a[i];
        i++;
    }
    s[j]='\0';
    return s;
}
```

7. (0.5) Koliko najmanje memorijskog prostora (u bajtovima) zauzima promenljiva koja je tipa struct student ukoliko promenljiva tipa char zauzima 1 bajt, int 4 bajta i double 8 bajtova?

```
struct student
{
    char ime[20];
    int godina;
    double prosek;
    union
    {
        struct { char imeOca[20]; char imeMajke[20]; } saRoditeljima;
        struct { char imeDoma[120]; int samofinansiranje;} studentskiDom;
        struct { char adresa[50]; double kirija; } privatno;
    } mestoStanovanja;
};
```

Najmanje 156 bajtova

8. (2.5) Napisati program koji sadrži rekurzivnu funkciju koja za dva data cela broja a i b računa sumu celih brojeva koji se nalaze između njih. Brojevi a i b su predati prilikom pokretanja programa na sledeći način:
- ```
./saberi 4 8
```

```
#include <stdio.h>
int saberi(int a, int b){
 if(a==b)
 return b;
 return a + saberi(a+1,b);
}
main(int argc, char **argv)
{
 int x,y;
 x = atoi(argv[1]);
 y = atoi(argv[2]);
 printf("%d %d %d",x,y, saberi(x,y));
}
```

9. (1.0) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
main()
{
 int a=10;
 float b=4.5;
 if (a/b) printf("%d\n",(int)(a++?-a:b*b*b));
 b?printf("Test 1\n"):printf("%d\n",(2+a)/++a);
}
```

10. (0.5) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
#define POMNOZI(a,b) (b * a)
main() {
 int c=8;
 float d=4.0;
 printf("%.2f\n", POMNOZI (c - d, d - c));
}
```

-64.00

11. (1.5) Šta je rezultat sledećeg koda

```
#include <stdio.h>
main(){
 unsigned x, y;
 scanf("%d%d", &x, &y);

 while (x<50)
 {
 y >>= (x<=1)%3;
 printf("%d %d\n",x, y);
 }
}
```

Ako se kao vrednosti promenljivih x i y unesu 2 i 100

4 50  
8 12  
16 6  
32 1  
64 0

12. (2.0) Šta je rezultat sledećeg koda?

```
char *str[] = { "dvestotine", "hiljadu", "cetrístotine", "milion", "sesnaest"};
char **s1 = str;
int n = sizeof(str)/sizeof(str[0]);
while(n-->0)
 printf("%c\n", *((*++s1)++));
```

h  
c  
m  
s  
Segmentation fault