

# Operációs rendszerek BSc

Operációs rendszerek Bsc

2022. tavaszi féléves feladat

**Készítette:**

Zarándi Ákos Bsc

Gazdaságinformatikus

DX6C4R

## 1. Feladat

Adott az alábbi terhelés esetén a rendszer. Határozza meg az indulás, befejezés, várakozás/átlagos várakozás és körülfordulás/átlag körülfordulás, válasz/átlagos válaszidő és a CPU kihasználtság értékeket az RR: 6 ms ütemezési algoritmusok mellett! (cs:0,1ms, sch:0,1ms)

A RR az tulajdonképpen FCFS+ időszeletoes beavatkozás, egy soron vannak a processzek, az első kapja meg a CPU-t, de csak addig, amíg az ideje le nem jár.

	P1	P2	P3	P4	P5			
Érkezés	1	4	4	7	8			
CPU idő	1	11	4	7	4			
Indulás								
Befejezés								
Várakozás								
	Processz	Érkezés	CPU idő	Indulás	Befejezés	Várakozás	Maradék idő	Váró processz
	p1	1	1	1	6	0	0	p2
	p2	4	11	6	17	2	2	p3
	p3	4	4	17	21	13	0	p2,p4
	p2*	17	2	21	23	4	0	p4,p5
	p4	7	7	23	30	16	0	p5
	p5	8	4	30	34	26	0	
			Befejezés	Körülforgási idő	Várakozási idő	CPU idő		
		p1	6	5	0	1		
		p2	21	19	6	11		
		p3	21	17	13	4		
		p4	30	23	16	7		
		p5	34	26	26	4		

CPU kihasznált	99,27 ms
Körülforgulási	28 ms
Várakozási idő	11 ms

## 2. Feladat

Írjon egy C programot, ami egy másodfokú egyenlet megoldóképletét reprezentálja osztott memória szegmens segítségével. A műveletvégzéshez szükséges adatokat egy bemeneti fájlból olvassa be, majd az adatokat és az eredményt adja vissza egy kimeneti fájlba. A bemeneti illetve a kimeneti fájl struktúrája kötött!

Elsősorban a változókat deklarálom. Ezután létrehozok egy kulcsot, aztán rácsatlakozok a memóriaszegmensre. Továbbá létrehozom a forrás és a cél fájlt. A forrás fájlt beolvasom, majd kiíratom a szegmenssel. A másodfokú egyenletet megoldom(komplex számokkal), beleíratom az eredmény.txt mappába. Végző sorban lecsatlakozom a szegmensről

```

int main()
{
    // Változók deklarálása
    int a, b, c;
    float x, y, d;
    char as, bs, cs, space=' ';

    // Kulcs létrehozása
    key_t kulcs = ftok("shmfile",65);

    // Az shmget egy azonosítót ad vissza az shmid-vel
    int shmid = shmget(kulcs, 1024, 0666|IPC_CREAT);

    // shmat-al csatlakozunk a memóriaszegmensre
    char *str = (char*) shmat(shmid,(void*)0,0);

    FILE* ptr;

    if ((ptr=fopen("teszt.txt", "r")) == NULL){
        printf("Fajl megnyitása sikertelen! A hiba oka:\n");
        perror(0);
        exit(1);
    }

    FILE* ptr2;

    if ((ptr2=fopen("eredmeny.txt", "w")) == NULL){
        printf("Fajl megnyitása sikertelen! A hiba oka:\n");
        perror(0);
        exit(1);
    }

    // Első fájl tartalmának olvasása majd kiírása a memóriaszegmensbe, fájl pointer lezárása
    fgets(str, 256, ptr);
    fclose(ptr);

    printf("A szegmensbe irt adat: %s\n", str);

    // Szétszedjük a beolvasott sort,
    // és mivel ezek karakterek, átkonvertáljuk intbe őket
    as=str[0];
    a=atoi(&as);

    bs=str[2];
    b=atoi(&bs);

    cs=str[4];
    c=atoi(&cs);

    // Másodfokú egyenlet megoldása (komplex számokkal is működik)

    d = b * b - 4 * a * c;

    printf("d: %f\n\n", d);

    if(d < 0){
        printf("A gyokok komplex szamok.\n\n");

        printf("A(z) %dx^2 + %dx + %d = 0 egyenlet gyokei: \n", a, b, c);
        printf("%.4f+%.4fi", -b/(2*a), sqrt(-d) / (2*a));
        printf(", %.4f+%.4fi", -b/(2*a), -sqrt(-d) / (2*a));
        printf("\n");
    }

    return 0;
}

else if(d == 0){
    printf("A ket gyok megegyezik.\n");

    x = -b / (2*a);
    printf("A(z) %dx^2 + %dx + %d = 0 egyenlet gyoke: %.4f\n", a, b, c, x);

    return 0;
}

```

```

else{
    printf("A gyokok valos szamok.\n");

    x = ( -b + sqrt(d)) / (2*a);
    y = ( -b - sqrt(d)) / (2*a);
    printf("A(z) %dx^2 + %dx +%d = 0 egyenlet gyokei: %.4f , es %.4f\n", a, b, c, x, y);
}

fprintf(ptr2, "%d %d %d %.4f %.4f\n", a, b, c, x, y);

// Lecsatlakozunk a memóriaszegmensről.
shmdt(str);
fclose(ptr2);

return 0;
}

```

```

b3lga@b3lga-VirtualBox:~$ cc write.c -o write.out -lm
write.c: In function 'main':
write.c:74:16: warning: format '%f' expects argument of type 'double', but argum
ent 2 has type 'int' [-Wformat=]
  74 |     printf("%.4f%+.4fi", -b/(2*a), sqrt(-d) / (2*a));
      |                ^~         ~~~~~
      |                |         |
      |            double      int
      |            %.4d
write.c:75:18: warning: format '%f' expects argument of type 'double', but argum
ent 2 has type 'int' [-Wformat=]
  75 |     printf(", %.4f%+.4fi", -b/(2*a), -sqrt(-d) / (2*a));
      |                ^~         ~~~~~
      |                |         |
      |            double      int
      |            %.4d
b3lga@b3lga-VirtualBox:~$ ./write.out
A szegmensbe irt adat: 4 6 2
d: 4.000000

A gyokok valos szamok.
A(z) 4x^2 + 6x +2 = 0 egyenlet gyokei: -0.5000 , es -1.0000
b3lga@b3lga-VirtualBox:~$

```

