

JEGYZŐKÖNYV

Operációs rendszerek BSc

2021. tavasz féléves feladat

Készítette: **Horváth Ákos**
Zsigmond

Neptunkód: **R3SZY2**

A feladat leírása:

C program, 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.

Bemeneti fájl:

i(megoldani kívánt egyenletek száma)
a b c

Kimeneti fájl:

a b c x y

A feladat elkészítésének lépései:

Az egyenletek reprezentálásához készítettem egy struktúrát, ami tartalmazta az egyenlet változóit és gyökeit tartalmazza, illetve egy logikai változót, ami jelezte, hogy lesznek-e komplex gyökei az egyenletnek (ha a diszkrimináns negatív):

```
typedef struct equation
{
    float a;
    float b;
    float c;
    float root1;
    float root2;
    float realPart;
    float imaginaryPart;
    int isImaginary;
}eq_t;
```

Készítettem egy kulcsot a bemeneti fájl alapján:

```
key_t key = ftok("input", 65);
```

Fájl kezelése, adatok beolvasása:

```
fptr = fopen("input", "r");
if(fptr == NULL)
{
    perror("File cannot be opened!");
    return 1;
}

int numofequations = 0;
fscanf(fptr, "%d", &numofequations);

eq_t eqarray[numofequations];
printf("numofequations: %d\n", numofequations);

for(int i = 0; i < numofequations; i++)
{
    float a, b, c;
    fscanf(fptr, "%f %f %f", &a, &b, &c);
    eqarray[i].a = a;
    eqarray[i].b = b;
    eqarray[i].c = c;
    eqarray[i].root1 = 0;
    eqarray[i].root2 = 0;
```

```

    eqarray[i].isImaginary = 0;
    eqarray[i].realPart = 0;
    eqarray[i].imaginaryPart = 0;
}

```

```
fclose(fp);
```

Az osztott memória szegmens allokálása, a méret a beolvasott egyenlet struct-ok alapján:

```

int shmid = shmget(key, sizeof(eqarray), 0666|IPC_CREAT);
if(shmid == -1)
{
    perror("Shared memory segment allocation error!");
    return 1;
}

```

Mivel az egyenletmegoldó programnál is hozzá kell férni az osztott memória szegmenshez, és az erre használt függvénynek az egyik paramétere a memóriaszegmens mérete, message queue használatával elküldtem a másik processznek:

```

struct msg_buffer
{
    long mtype;
    int msg;
}message;

int msgid = msgget(key, 0644|IPC_CREAT);
message.mtype = 2;

message.msg = sizeof(eqarray);
msgsnd(msgid, &message, sizeof(message), 0);
printf("message sent: %d\n",message.msg);

```

Végül az egyenleteket tartalmazó tömb bemásolása az osztott memória szegmensbe, illetve lekapcsolódás a memóriaszegmensről:

```

eq_t* eq = (eq_t*) shmat(shmid, 0, 0);

memcpy(eq, eqarray, sizeof(eqarray));

shmdt(eq);

```

Egyenletmegoldó program:

Kulcs generálása:

```
key_t key = ftok("input", 65);
```

Memóriaszegmens méretét tartalmazó üzenet fogadása:

```

int msgid = msgget(key, 0644|IPC_CREAT);

msgrcv(msgid, &message, sizeof(message), 2, 0);
int buffer_size = message.msg;

```

Csatlakozás a memóriaszegmenshez:

```
int shmid = shmget(key, buffer_size, 0666|IPC_CREAT);  
if(shmid == -1)  
{  
    perror("Shared memory error!");  
    return 1;  
}
```

Adatok kiolvasása a memóriaszegmensből:

```
eq_t* eq;  
eq = (eq_t*) shmat(shmid, 0, 0);
```

Kimeneti fájl megnyitottam, majd egy for ciklussal átmentem a tömbön, megoldottam az egyenleteket ezzel a függvénnyel:

```
void solve_equation(eq_t* e)  
{  
    float discriminant = e->b * e->b - 4 * e->a * e->c;  
  
    if(discriminant > 0)  
    {  
        e->root1 = (-e->b + sqrt(discriminant)) / 2 * e->a;  
        e->root2 = (-e->b - sqrt(discriminant)) / 2 * e->a;  
    }  
    else if (discriminant == 0)  
    {  
        e->root1 = e->root2 = -e->b / (2*e->a);  
    }  
    else  
    {  
        e->isImaginary = 1;  
        e->realPart = -e->b / (2*e->a);  
        e->imaginaryPart = sqrt(-discriminant) / (2*e->a);  
    }  
}
```

Majd kiíratam a konzolra, és beíratam a kimeneti fájlba a megoldásokat:

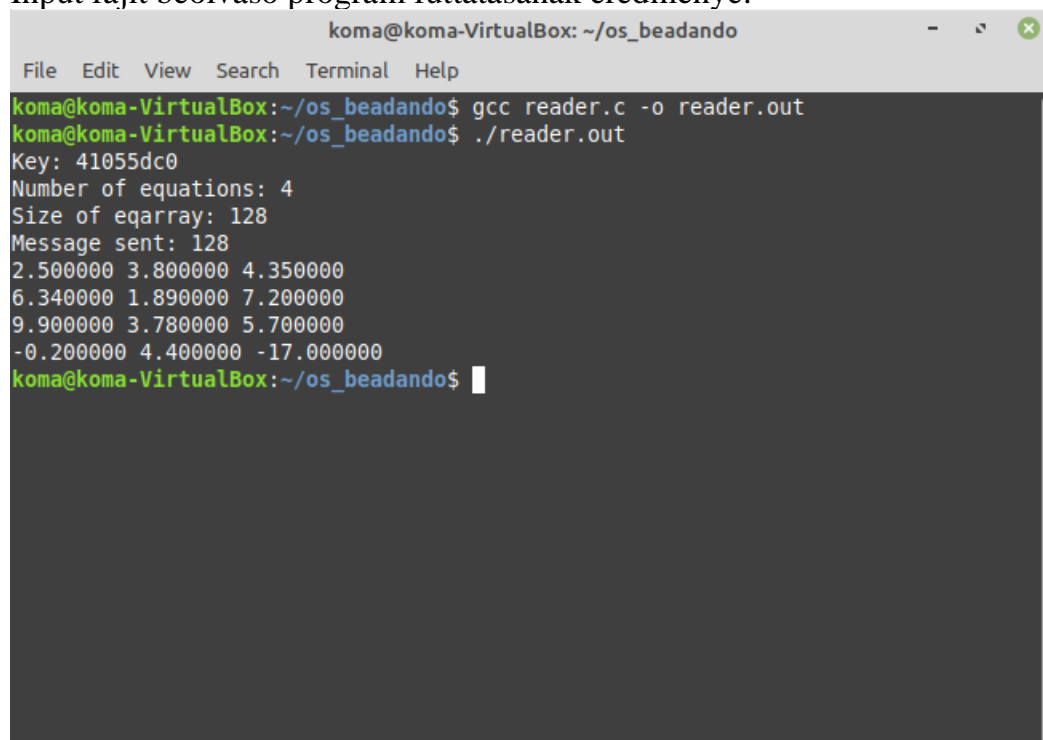
```
    solve_equation(&eq[i]);
    eq_t e = eq[i];
    if(e.isImaginary == 0)
    {
        printf("%f %f %f %f %f\n", e.a, e.b, e.c, e.root1,      e.root2);
        fprintf(fp, "%f %f %f %f %f\n", e.a, e.b, e.c,      e.root1, e.root2);
    }
    else
    {
        printf("%f %f %f %f+%f %f+%f\n", e.a, e.b, e.c,      e.realPart,
e.imaginaryPart, e.realPart,      e.imaginaryPart);
        fprintf(fp, "%f %f %f %f+%f %f+%f\n", e.a, e.b,      e.c, e.realPart,
e.imaginaryPart, e.realPart, e.imaginaryPart);
    }
}
```

Végül lecsatlakoztam a memóriaszegmensről, és bezártam:

```
shmdt(eq);
shmctl(shmid,IPC_RMID,NULL);
```

A futtatás eredménye:

Input fájlt beolvasó program futtatásának eredménye:



```
koma@koma-VirtualBox: ~/os_beadando
File Edit View Search Terminal Help
koma@koma-VirtualBox:~/os_beadando$ gcc reader.c -o reader.out
koma@koma-VirtualBox:~/os_beadando$ ./reader.out
Key: 41055dc0
Number of equations: 4
Size of eqarray: 128
Message sent: 128
2.500000 3.800000 4.350000
6.340000 1.890000 7.200000
9.900000 3.780000 5.700000
-0.200000 4.400000 -17.000000
koma@koma-VirtualBox:~/os_beadando$
```

Egyenletmegoldó, output fájlba író program futtatásának eredménye:

```
koma@koma-VirtualBox: ~/os_beadando
File Edit View Search Terminal Help
koma@koma-VirtualBox:~/os_beadando$ gcc writer.c -o writer.out -lm
koma@koma-VirtualBox:~/os_beadando$ ./writer.out
key: 41055dc0
Waiting for message...
Recieved message: 128
2.500000 3.800000 4.350000 0.000000+0.000000i 0.000000+0.000000i
6.340000 1.890000 7.200000 0.000000+0.000000i 0.000000+0.000000i
9.900000 3.780000 5.700000 0.000000+0.000000i 0.000000+0.000000i
-0.200000 4.400000 -17.000000 0.000000+0.000000i 0.000000+0.000000i
2.500000 3.800000 4.350000 -0.760000+1.078147i -0.760000+1.078147i
6.340000 1.890000 7.200000 -0.149054+1.055192i -0.149054+1.055192i
9.900000 3.780000 5.700000 -0.190909+0.734378i -0.190909+0.734378i
-0.200000 4.400000 -17.000000 0.200000 0.680000
koma@koma-VirtualBox:~/os_beadando$
```

Bemeneti és kimeneti fájl:

```
koma@koma-VirtualBox: ~/os_beadando
File Edit View Search Terminal Help
koma@koma-VirtualBox:~/os_beadando$ cat input
4
2.5 3.8 4.35
6.34 1.89 7.2
9.9 3.78 5.7
-0.2 4.4 -17
koma@koma-VirtualBox:~/os_beadando$ cat output
2.500000 3.800000 4.350000 -0.760000+1.078147i -0.760000+1.078147i
6.340000 1.890000 7.200000 -0.149054+1.055192i -0.149054+1.055192i
9.900000 3.780000 5.700000 -0.190909+0.734378i -0.190909+0.734378i
-0.200000 4.400000 -17.000000 0.200000 0.680000
koma@koma-VirtualBox:~/os_beadando$
```