# Jegyzőkönyv

Operációs rendszerek BSc

2022. tavasz féléves feladat

# Készítette:

Juhász Tibor

Mérnökinformatikus BSc. levelező

X7KWVG

#### 10. feladat

#### IPC Feladat leírás:

Írjon egy olyan C programot, mely egy fájlból számpárokat kiolvasva meghatározza a legnagyobb közös osztóját. A feladat megoldása során használjon message queue(üzenetsoros) IPC mechanizmust, valamint a kimenet kerüljön egy másik fájlba.

A kimeneti fájl struktúrája kötött!

Kimeneti fájl (Az x,y jelzi a bemeneti adatokat, a z pedig a kimenet eredményét):

хуг

Megoldás:

Az X7KWVG\_felevesFeladat.c programmal számpárokat olvasunk be a szamok.txt-ből.

A számokat egy tömbben tárolom. Amennyiben a fájlt nem sikerül megnyitni, hibaüzenetet dob a program ("A fajl nem nyithato meg").

```
szamok = fopen("szamok.txt", "r");
osztokki = fopen("osztok.txt", "w");

if (NULL == szamok) {
  printf("A fajl nem nyithato meg!\n");
  exit(1);
}

for (i = 0; i < 8; i++){

fscanf(szamok, "%d", &szamTomb[i]);
}
fclose(szamok);</pre>
```

Amennyiben sikerült beolvasni a számokat, kettesével kezeli őket a program. Megnézzük, hogy melyik szám a kisebb. Amennyiben a < b, megcseréjük a két számot, hogy egyszerűbb legyen a legnagyobb közös osztó kiszámítása.

```
for(n = 0; | n < 4; n++){
a = szamTomb[i];
b = szamTomb[i+1];

if(a < b){
c = a;
a = b;
b = c;
}</pre>
```

Ezután kiszámoljuk a legnagyobb közös osztót és betesszük az osztok tömbbe:

```
c = b;

while(a%c !=0 || b%c !=0){

c = c-1;

}

osztok[n] = c;
```

Ezután kiírjuk az eredeti számpárt, illetve a legnagyobb közös osztójukat, ami a c változóban van eltárolva.

```
key_t key;
int msgid;
key = ftok("szamok",65);

msgid = msgget(key, 0666 | IPC_CREAT);
message.mesg_type = 1;

printf("Beolvasott szamok: %d\t%d\n",szamTomb[i], szamTomb[i+1]);
printf("Legnagyobb kozos osztojuk: %d\n",c);

msgsnd(msgid, &message, sizeof(message), 0);
```

Majd a bemeneti és kimeneti adatokat az osztok.txt fájlba írjuk ki.

```
if(osztokki != NULL)
fprintf(osztokki, "%d %d %d\n",a,b,osztok[n]);
i = i + 2;
}
fclose(osztokki);
```

Az osztok.txt fájl tartalma a program futása után:

```
63 52 1
700 250 50
2000 970 10
10002 347 1
```

#### Program fordítása és futtatása:

```
tibi@tibi-VirtualBox:~$ gcc -o X7KWVG_felevesBeadando X7KWVG_felevesBeadando.c
tibi@tibi-VirtualBox:~$ ./X7KWVG_felevesBeadando
Beolvasott szamok: 63 52
Legnagyobb kozos osztojuk: 1
Beolvasott szamok: 250 700
Legnagyobb kozos osztojuk: 50
Beolvasott szamok: 970 2000
Legnagyobb kozos osztojuk: 10
Beolvasott szamok: 347 10002
Legnagyobb kozos osztojuk: 1
tibi@tibi-VirtualBox:~$
```

## 1-es feladat

## Feladat leírás:

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

	P1	P2	Р3	P4
Érkezés	0	8	12	20
CPU idő	15	7	26	10
Indulás				
Befejezés				
Várakozás				

Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.

# Megoldás:

FCFS	Érkezés	CPU idő	Indulás	Befejezés	Várakozás
P1	0	15	0	15	0
P2	8	7	15	22	7
Р3	12	26	22	48	10
P4	20	10	48	58	28
Átlagos várakozási idő:			11,25		
Befejezési idő:			58		
Körülfordulási idő:			58		
Átlagos körülfordulási idő:			35,75		

# Diagramm:

