

JEGYZŐKÖNYV

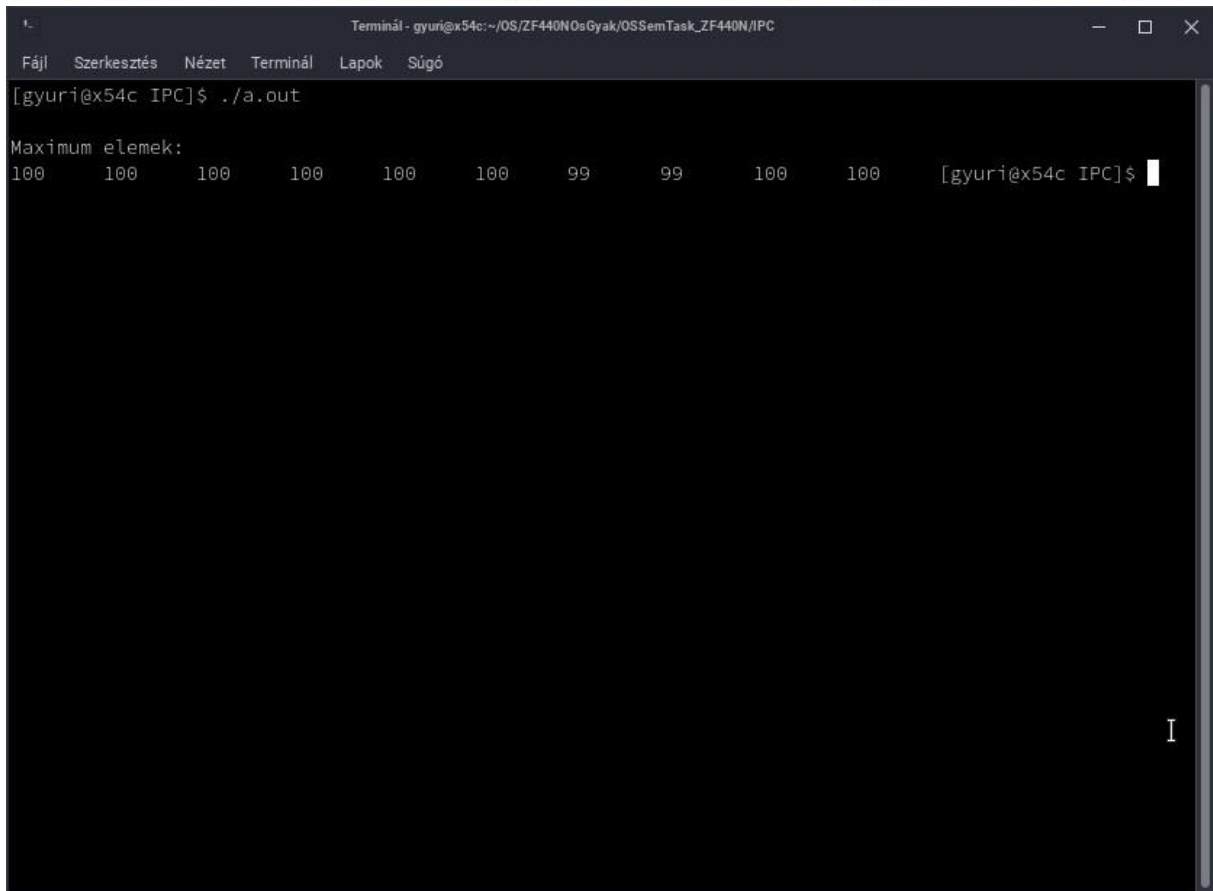
Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Kórád György**

Neptunkód: **ZF440N**

8. Készítsen egy programot, ami egy 1000 elemű egész szám típusú tömbben úgy keresi meg a maximumot, hogy 10 szálal futtat parhuzamosan, amik közül mindegyik 100 elemet vizsgál meg. Az eredeti szülő processz nem számol, viszont ő gyűjti be az eredményeket, amit pipe-on keresztül vár a processzekről.



```
Terminál - gyuri@x54c: ~/OS/ZF440N0sGyak/OSSemTask_ZF440N/IPC
Fájl Szerkesztés Nézet Terminál Lapok Súgó
[gyuri@x54c IPC]$ ./a.out
Maximum elemek:
100 100 100 100 100 100 99 99 100 100 [gyuri@x54c IPC]$
```

A program véletlenszerűen generál 1 és 100 között számokat, majd az 1000 elemű tömböt 10 részre osztva adja át a child processzeknek, azok pedig egy tömbbe küldik vissza az eredményüket.

```

int tomb[100]; //helyi tomb
int tomb1[100]; // "kapott" tomb

for(int i = 0; i<100; i++){
    read(fd[0][0], &tomb1[i], sizeof(int)); //itt történik a tömb beolvasása és eltárolása egy másik tömbben
    tomb[i] = tomb1[i];
}
int max1 = 0;
for(int j = 0; j < 100; j++){ //itt a maximum kiszámítása történik
    if(tomb[j] > max1){
        max1 = tomb[j];
    }
}

write(fd[1][1], &max1, sizeof(int)); // a maximum visszaküldése
close(fd[0][0]);
close(fd[1][1]); //a nyitva maradt pipeok bezárása
return 0;
}

```

A fenti képen egy child látható, a tomb1[]-ben van 100 az ezer elemű tömb elemei közül amit átmásol egy helyi tömbbe, majd ezek között megkeresi a maximum elemet, az fd[0][0] a csővezeték írható végére vezeti végül.

Majd a szülő processz az fd[1][1]-ről kiolvassa.

```

int nagyTomb[1000];

for(int i = 0; i < 1000; i++){ //random szám generálás
    nagyTomb[i] = (rand() % (100 - 0 +1)) + 0;
}

```

8. Adott egy rendszerbe az összes osztály-erőforrások száma: R (R1: 8; R2: 9; R3: 13)

A rendszerbe 4 processz van: P1, P2, P3, P4.

Teljesíthető-e P2 (1, 2, 2) kérése?

Biztonságos-e vagy nem biztonságos holtpontmentesség szempontjából a rendszer - a következő kiinduló állapot alapján?

a) Határozza meg a processzek által igényelt erőforrások mátrixát – P2 processz kérésének figyelembe vételével?

b) Határozza meg pillanatnyilag szabad erőforrások számát?

c) Igazolja az egyes processzek végrehajtásának lehetséges sorrendjét - számolással?

Kiinduló állapot

	Max. igény				Foglal		
	R1	R2	R3		R1	R2	R3
P1	4	2	5	P1	2	2	3
P2	7	7	7	P2	0	1	1
P3	1	4	3	P3	1	2	2
P4	3	7	4	P4	2	1	2

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	4	2	5		2	2	3		2	0	2	
P2	7	7	7		0	1	1		7	6	6	
P3	1	4	3		1	2	2		0	2	1	
P4	3	7	4		2	1	2		1	6	2	

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	-	-	-		0	0	0		0	0	0	
P2	7	7	7		0	1	1		7	6	6	
P3	1	4	3		1	2	2		0	2	1	
P4	3	7	4		2	1	2		1	6	2	

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	-	-	-		0	0	0		0	0	0	
P2	7	7	7		0	1	1		7	6	6	
P3	-	-	-		0	0	0		0	0	0	
P4	3	7	4		2	1	2		1	6	2	

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	-	-	-		0	0	0		0	0	0	
P2	7	7	7		0	1	1		7	6	6	
P3	-	-	-		0	0	0		0	0	0	
P4	-	-	-		0	0	0		0	0	0	

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	-	-	-		0	0	0		0	0	0	
P2	7	7	7		1	3	3		6	4	4	
P3	-	-	-		0	0	0		0	0	0	
P4	-	-	-		0	0	0		0	0	0	

MAX IGÉNY				FOGLAL			IGÉNY			SZABAD		
	R1	R2	R3		R1	R2	R3		R1	R2	R3	
P1	-	-	-		0	0	0		0	0	0	
P2	-	-	-		0	0	0		0	0	0	
P3	-	-	-		0	0	0		0	0	0	
P4	-	-	-		0	0	0		0	0	0	

Lehetséges sorrend: P1, P3, P4,P2