

JEGYZŐKÖNYV

Operációs rendszerek Bsc
2022. tavasz, féléves feladat

Készítette: Keresztes Iulia
Neptunkód: ULA7Z2

Miskolc, 2022

1. feladat: IPC mechanizmusok

A feladat leírása:

12. Írjon C nyelvű programokat, ami
hozzon létre egy osztott memória szegmenst
ebbe olvassa be egy nagy file tartalmát
egy másik program pedig olvass ki az osztott mem szegmenbol
es írja bele egy másik fileba
vegul szuntesse meg az shm szegmenst

A feladat elkészítésének lépései:

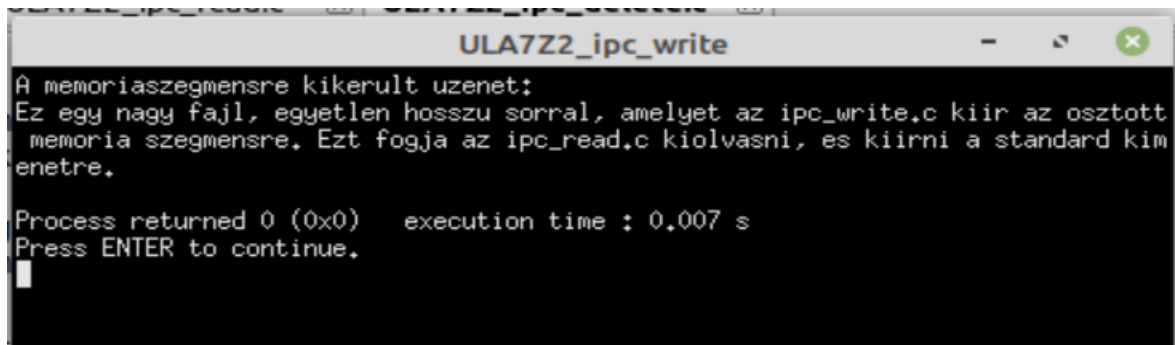
A feladatokat három program végezi el. Az első, az ULA7Z2_ipc_write.c létrehoz egy osztott memóriaszegmenst, majd kiolvassa egy nagy fájl tartalmát, és beírja a memóriaszegmensre, majd bezárja a fájlt.

A második program, az ULA7Z2_ipc_read.c azonosítja a korábban létrehozott szegmenst (ha az valóban létre volt hozva), és kiolvassa belőle az üzenetet. Létrehoz egy fájlt, és beleírja az üzenetet.

A harmadik program, az ULA7Z2_ipc_delete.c megszünteti a memóriaszegmenst, ha az létezik.

A futtatás eredménye:

Az ULA7Z2_ipc_write futtatásának eredménye:



```
ULA7Z2_ipc_write
A memóriaszegmensre kikerult uzenet:
Ez egy nagy fajl, egyetlen hosszú sorral, amelyet az ipc_write.c kiír az osztott
memória szegmensre. Ezt fogja az ipc_read.c kiolvasni, és kiírni a standard kin
enetre.

Process returned 0 (0x0)   execution time : 0.007 s
Press ENTER to continue.
█
```

A standard kimenetre is kiírja a memóriaszegmensre írt üzenetet.

Az ULA7Z2_ipc_read futtatásának eredménye:

```
ULA7Z2_ipc_read
A memoria szegmensen levo uzenet:
Ez egy nagy fajl, egyetlen hosszú sorral, amelyet az ipc_write.c kiir az osztott
memoria szegmensre. Ezt fogja az ipc_read.c kiolvasni, es kiirni a standard kim
enetre.
Process returned 0 (0x0)   execution time : 0.008 s
Press ENTER to continue.
```

A kiolvasott üzenetet nem csak egy új fájlba írja ki, hanem a standard kimenetre is.

Az ULA7Z2_ipc_read futtatása abban az esetben, ha nem volt létrehozva memóriaszegmens:

```
ULA7Z2_ipc_read
Nem letezik a memoria szegmens!
Process returned 0 (0x0)   execution time : 0.007 s
Press ENTER to continue.
```

Az ULA7Z2_ipc_delete futtatásának eredménye:

```
ULA7Z2_ipc_delete
Torolve
Process returned 0 (0x0)   execution time : 0.007 s
Press ENTER to continue.
```

Tudatja a felhasználóval, hogy a szegmens meg lett szüntetve.

Az ULA7Z2_ipc_delete futtatásának eredménye abban az esetben, ha nem létezik a szegmens:

```
ULA7Z2_ipc_delete
A memóriaszegmens nem letezik.
Process returned 0 (0x0)   execution time : 0.006 s
Press ENTER to continue.
```

2. feladat: OS ütemezési algoritmusok

A feladat leírás:

5. Adott négy processz (A, B, C, D) a rendszerbe, induláskor a p_cpu értéke $A=0$, $B=0$, $C=0$, $D=0$. A rendszerben a $P_USER = 60$. Az óráütés 1 indul, a befejezés 301-ig.

Induláskor a p_usrpri $A=60$, $B=60$, $C=65$ és $D=60$.

Induláskor a p_nice értéke $A=0$, $B=0$, $C=5$ és $D=0$.

a.) Határozza meg az ütemezést *RR nélkül 301 óráütésig* - táblázatba!

b.) Minden óráütem esetén határozza meg a processzek sorrendjét óráütés *előtt/után*.

c.) Igazolja a számítással a tanultak alapján.

A feladat eredménye:

Óráütés	A Processz		B Processz		C Processz		D Processz		Újraütemezés	
	p_usrpri	p_cpu	p_usrpri	p_cpu	p_usrpri	p_cpu	p_usrpri	p_cpu	óraütés előtt	óraütés után
Start	60	0	60	0	65	0	60	0		A
1		1		0		0		0	A	A
2		2		0		0		0	A	A
...
100	97	100	60	0	70	0	60	0	A	B
101		75		1		0		0	B	B
...
200	88	75	97	100	70	0	60	0	B	D
201		56		75				1	D	D
...
300	81	56	88	75	70	0	97	100	D	C
301		42		56		1		75	C	C

Lépések:

Prioritások alapján az első futó processz az A processz lesz. Mivel Round Robin nélküli az ütemezés, így fut a 100. óráütésig. Itt számolás:

$$A - p_cpu = 100 * 0.75 = 75$$

$$p_usrpri = 60 + 75/2 = 97 \text{ (nincs } p_nice)$$

$$B - p_cpu = 0$$

$$p_usrpri = 60 \text{ (nincs } p_nice)$$

$$C - p_cpu = 0$$

$$p_usrpri = 60 + 2 * 5 = 70 \text{ (} p_nice = 5)$$

$$D - p_{\text{cpu}} = 0$$

$$p_{\text{usrpri}} = 60 \text{ (nincs } p_{\text{nice}})$$

A legmagasabb prioritású processzek a B és D, így a 100. óraütéstől B fut, egészen a 200. óraütésig, ahol újra számolni kell:

$$A - p_{\text{cpu}} = 75 * 0.75 = 56$$

$$p_{\text{usrpri}} = 60 + 56/2 = 88$$

$$B - p_{\text{cpu}} = 100 * 0.75 = 75$$

$$p_{\text{usrpri}} = 97$$

$$C - p_{\text{cpu}} = 0$$

$$p_{\text{usrpri}} = 60 + 2 * 5 = 70$$

$$D - p_{\text{cpu}} = 0$$

$$p_{\text{usrpri}} = 60$$

Ekkor a D-nek a legmagasabb a prioritása. Ez fut a 200. és a 300. óraütések között. Újabb számolás:

$$A - p_{\text{cpu}} = 56 * 0.75 = 42$$

$$p_{\text{usrpri}} = 60 + 42/2 = 81$$

$$B - p_{\text{cpu}} = 75 * 0.75 = 56$$

$$p_{\text{usrpri}} = 88$$

$$C - p_{\text{cpu}} = 0$$

$$p_{\text{usrpri}} = 60 + 2 * 5 = 70$$

$$D - p_{\text{cpu}} = 100 * 0.75 = 75$$

$$p_{\text{usrpri}} = 60 + 75/2 = 97$$

Így a 300. óraütéstől a C processz fut tovább.