

JEGYZŐKÖNYV

Operációs rendszerek BSc

2022. tavasz féléves feladat

Készítette: **Nyíri Levente**

Neptunkód: **F023QC**

1. feladat:

IPC mechanizmus

A feladat leírása:

1.feladat: Írjon egy C programot, amely egy szülőprocessz révén készít egy gyermekprocesszt, a gyermekben futtasson egy másik programot az `execl()` hívással(Environemnten keresztül kapja

meg, hogy mit indítson el a program), mely kiírja a PID-jét és szülője PID-jét, majd a szülő is kiírja mi a PID-je és a gyermeke PID-je.

A feladat elkészítésének lépései:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <unistd.h>
5  #include <sys/wait.h>
6
7  int main()
8  {
9
10
11     pid_t cpid = fork();
12
13     if (cpid == -1)
14     {
15         perror("Elforkol");
16         exit(-1);
17     }
18
19     int szpid = getpid();
20
21     if (cpid == 0)
22     {
23         char gyperpid[6];
24         char szulpid[6];
25         sprintf(szulpid,"%d",getppid());
26         sprintf(gyerpid,"%d",getpid());
27         execl("./masikp","masikp",szulpid,gyerpid,(char *)NULL);
28         printf("Error execl nem ter vissza");
29     }
30
31     else
32     {
33         printf("A szulo vagyok \n");
34         printf("Szulo pidje: %d Gyerek pidje: %d \n", szpid,cpid);
35
36
37         exit(0);
38     }
39
40 }
41
42 return 0;
43 }
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(int argc, char *argv[]){
4      printf("masikp.c vagyok\n");
5      printf("Szulo pidje: %s\n",argv[1]);
6      printf("Gyerek pidje: %s\n",argv[2]);
7      return 0;
8  }
```

A futtatás eredménye:

```
levente@levente-VirtualBox:~/OS$ gcc os.c -o os
levente@levente-VirtualBox:~/OS$ gcc masikp.c -o masikp
levente@levente-VirtualBox:~/OS$ ./os
A szulo vagyok
Szulo pidje: 2430 Gyerek pidje: 2431
levente@levente-VirtualBox:~/OS$ masikp.c vagyok
Szulo pidje: 2430
Gyerek pidje: 2431
```

2. feladat:

OS algoritmusok

A feladat leírása:

6. feladat: Adott egy igény szerinti lapozást használó számítógéprendszer, melyben futás közben egy processz számára a következő laphivatkozással lehet hivatkozni: 6, 5, 4, 3, 5, 6, 2, 8, 5, 6, 5, 4, 7, 8, 4, 5, 6, 5, 5, 8. Memóriakeret (igényelt lapok): 3, ill. 4 memóriakeret. Készítse el a laphivatkozások betöltését külön-külön táblázatba 3, ill. 4 memóriakeret esetén. Mennyi laphiba keletkezik az alábbi algoritmusok esetén: FIFO, OPT? Hasonlítsa össze és magyarázza az eredményeket!

A feladat elkészítésének lépései:

OPT

3 méretű memóriánál																			
	6	5	4	3	5	6	2	8	5	6	5	4	7	8	4	5	6	5	8
MEM 1	6	6	6	6	6	6	6	6	6	6	6	6	4	4	4	4	6	6	6
MEM 2		5	5	5	5	5	5	5	5	5	5	5	7	7	7	7	5	5	5
MEM 3			4	3	3	3	2	8	8	8	8	8	8	8	8	8	8	8	8
Laphiba	x	x	x	x			x	x					x	x			x	x	
Egyezés																			
Laphiba																			
Összes laphiba: 10																			
4 méretű memóriánál																			
	6	5	4	3	5	6	2	8	5	6	5	4	7	8	4	5	6	5	8
MEM 1	6	6	6	6	6	6	6	6	6	6	6	6	7	7	7	7	6	6	6
MEM 2		5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
MEM 3			4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
MEM 4				3	3	3	2	8	8	8	8	8	8	8	8	8	8	8	8
Laphiba	x	x	x	x			x	x					x				x		
Egyezés																			
Laphiba																			
Összes laphiba: 8																			

A 3 méretű laphibánál több laphiba keletkezik.

FIFO

[illegible]

A futtatás eredménye: 4 memóriakeret esetén kevesebb laphiba történt a FIFO és OPT algoritmusok használatával.