

Jegyzőkönyv Operációs rendszerek

5. gyakorlat

1.feladat

Timkó András
HVS05V
ge-BGI

1. A `system()` rendszerhívással hajtson végre egy parancsot.

```
GNU nano 5.4 hzs05v1fel.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    char vegrehajto[20];
    strcpy( vegrehajto, "ls -l" );
    system(vegrehajto);

    return(0);
}

root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# nano hzs05v1fel.c
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# gcc hzs05v1fel.c -o hzs05v1fel
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v1fel
total 24
-rwxr-xr-x 1 root root 16616 márc  7 17:44 hzs05v1fel
-rw-r--r-- 1 root root  167 márc  7 17:44 hzs05v1fel.c
```

2. Írjon programot, amely billentyűzetről bekér Unix parancsokat és végrehajtja őket, majd kiírja a szabványos kimenetre

```
GNU nano 5.4 hzs05v2fel.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    char str[10];

    printf( "Enter a command\n" );
    fgets(str, 20, stdin);
    system(str);
    return(0);
}

root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# nano hzs05v2fel.c
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# gcc hzs05v2fel.c -o hzs05v2fel
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v2fel
Enter a command
ls
hzs05v1fel hzs05v1fel.c hzs05v2fel hzs05v2fel.c
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v2fel
Enter a command
date
2022. márc. 7., hétfő, 17:56:03 CAT
```

3. Készítsen egy XY_parent.c és a XY_child.c programokat. A XY_parent.c elindít egy gyermek processzt, ami különbözik a szülőtől. A szülő megvárja a gyermek lefutását. A gyermek szöveget ír a szabványos kimenetre (10-ször)

```
GNU nano 5.4 hzs05v parent.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    system("./hzs05v_child");
    return(0);
}
```

```
GNU nano 5.4 hzs05v child.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    int i;
    for(i=1; i<=10; i++){
        printf("hello\n");
    }
    return(0);
}
```

```
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v_parent
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
hello
```

4. A `fork()` rendszerhívással hozzon létre egy gyerek processzt-t és abban hívjon meg egy `exec` családbeli rendszerhívást (pl. `execvp`).

```
GNU nano 5.4 hzs05v4fel.c *
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {
    char *argv[3] = {"Command-Line", ".", NULL};
    int pid = fork();
    if ( pid == 0 ) {
        execvp( "find", argv );
    }else{
        printf("Ez egy szulo process \n a gyereke: %d\n ", pid);
        wait(2);
    }

    return(0);
}
```

```
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v4fel
Ez egy szulo process
a gyereke: 2933
.
./hzs05v1fel
./hzs05v_parent
./hzs05v_child
./hzs05v_parent.c
./hzs05v2fel.c
./hzs05v1fel.c
./hzs05v4fel.c
./hzs05v2fel
./hzs05v_child.c
./hzs05v4fel
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307#
```

5. A `fork()` rendszerhívással hozzon létre gyerekeket, várja meg és vizsgálja a befejeződési állapotokat (gyerekekben: `exit`, `abort`, nullával való osztás)!

```
GNU nano 5.4 hzs05v5fel.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {

    int pid = fork();
    if ( pid == 0 )
    {
        exit(999);

    }

    int status;
    waitpid(pid, &status, 0);

    if ( WIFEXITED(status)) {

        int exit_status = WEXITSTATUS(status);
        printf("Exit kod: %d\n", exit_status);
    }

    return(0);
}
```

```
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v5fel
Exit kod: 231
```

```
GNU nano 5.4 hzs05v5_1fel.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main () {

    FILE *pf = fopen("nemletezik.txt","r");
    if (pf == NULL) {
        fprintf(stderr, "hiba a file megnyitasaval\n");
        abort();
    }
    return(0);
}
```

```
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v5_1fel
hiba a file megnyitasaval
Aborted
```

```
GNU nano 5.4 hzs05v5_2fel.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>

int main () {

    assert(1/0);
    return(0);

}
```

```
hzs05v5_2fel.c: In function 'main':
hzs05v5_2fel.c:7:11: warning: division by zero [-Wdiv-by-zero]
   7 |     assert(1/0);
     |           ^
```

```
root@elegemvan:/home/elegemvan/Desktop/HZS05V0sGyak/HZS05V_0307# ./hzs05v5_2fel
Floating point exception
```

I. Határozza meg FCFS és SJF esetén

- a.) A befejezési időt?
 - b.) A várakozási/átlagos várakozási időt?
 - c.) Ábrázolja Gantt diagram segítségével az aktív/várakozó processzek futásának menetét.
- Megj.: a Gantt diagram ábrázolása szerkesztő program segítségével vagy Excel programmal.

A szorgalmis programmal számolva

FCFS	Érkezés	CPU idő
P1	0	3
P2	1	8
P3	3	2
P4	9	20
P5	12	5

```
+-----+-----+-----+-----+-----+-----+
| Process sz. | Erkezési ido | CPU ido | Befejezési ido | Atfordulasi ido | Varakoz ido | Valasz ido |
+-----+-----+-----+-----+-----+-----+
| 1 | 0 | 3 | 3 | 3 | 0 | 0 |
| 2 | 1 | 8 | 11 | 10 | 2 | 2 |
| 3 | 3 | 2 | 13 | 10 | 8 | 8 |
| 4 | 9 | 20 | 33 | 24 | 4 | 4 |
| 5 | 12 | 5 | 38 | 26 | 21 | 21 |
+-----+-----+-----+-----+-----+-----+

Total Befejezési ido :- 98
Atlagos Befejezési ido :- 19.6

Total Atfordulasi ido :- 73
Atlagos Atfordulasi ido :- 14.6

Total Varakozasi ido :- 35
Atlagos Varakozasi ido :- 7

Total Valasz ido :- 35
Atlagos Valasz ido :- 7

Gantt abra(IS = idle statusz) :-

+-----+-----+-----+-----+-----+-----+
| P1 | P2 | P3 | P4 | P5 |
+-----+-----+-----+-----+-----+-----+
0 3 11 13 33 38
```

```
-SJF-
+-----+-----+-----+-----+-----+-----+
| Process sz. | Erkezési ido | CPU ido | Befejezési ido | Atfordulasi ido | Varakoz ido | Valasz ido |
+-----+-----+-----+-----+-----+-----+
| 1 | 0 | 3 | 3 | 3 | 0 | 0 |
| 2 | 1 | 8 | 13 | 12 | 4 | 4 |
| 3 | 3 | 2 | 5 | 2 | 0 | 0 |
| 4 | 9 | 20 | 38 | 29 | 9 | 9 |
| 5 | 12 | 5 | 18 | 6 | 1 | 1 |
+-----+-----+-----+-----+-----+-----+

Total Befejezési ido :- 77
Atlagos Befejezési ido :- 15.4

Total Atfordulasi ido :- 52
Atlagos Atfordulasi ido :- 10.4

Total Varakozasi ido :- 14
Atlagos Varakozasi ido :- 2.8

Total Valasz ido :- 14
Atlagos Valasz ido :- 2.8

Gantt abra(IS = idle statusz) :-

+-----+-----+-----+-----+-----+-----+
| P1 | P3 | P2 | P5 | P4 |
+-----+-----+-----+-----+-----+-----+
0 3 5 13 18 38
```

II. Round Robin (RR) esetén

- Ütemezze az adott időszel (5ms) alapján az egyes processzek (befejezési és várakozási/átlagos várakozási idő) paramétereit (ms)!
- A rendszerben lévő processzek végrehajtásának sorrendjét?
- Ábrázolja Gantt diagram segítségével az *aktív/várakozó* processzek futásának menetét!"

RR: 5ms	Érkezés	CPU idő
P1	0	3
P2	1	8
P3	3	2
P4	9	20
P5	12	5

-RR-							
Process sz.	Erkezesi ido	CPU ido	Befejezesi ido	Atfordulasi ido	Varakoz ido	Valasz ido	
1	0	3	3	3	0	0	
2	1	8	13	12	4	2	
3	3	2	10	7	5	5	
4	9	20	38	29	9	4	
5	12	5	23	11	6	6	
Total Befejezesi ido :- 87							
Atlagos Befejezesi ido :- 17.4							
Total Atfordulasi ido :- 62							
Atlagos Atfordulasi ido :- 12.4							
Total Varakozasi ido :- 24							
Atlagos Varakozasi ido :- 4.8							
Total Valasz ido :- 17							
Atlagos Valasz ido :- 3.4							
MS:- 5							
Gantt abra(IS = idle statusz) :-							
P1	P2	P3	P2	P4	P5	P4	
0	3	8	10	13	18	23	38