

Jegyzőkönyv

Operációs rendszerek BSC

2021 tavaszi féléves feladat

Készítette: Bártfai Bálint

Neptunkód: NWTCA

A feladat leírása:

A programnak létre kell hozni két darab csővezetékét (két file deskriptor) elforkol. A szülő elküldi a saját pidjét egy csőn keresztül gyermeknek ezt kiírja a képernyőre majd ezután visszaküldi a másik csővön saját pidjét.

A feladat elkészítésének lépései:

1.lépés:

Létrehoztam a csöveket és egy segéd változót ami majd a pideket fogja tárolni.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h> //pid_t

int main(){
    int parent_fds1[2], child_fds2[2]; //parent_fds a szulo csove
                                     //child_fds a gyerek csove
    int writemessage = 0;
```

2.lépés:

Le ellenőrzi hogy létre tudnak-e jönni a csövek..

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h> //pid_t

int main(){
    int parent_fds1[2], child_fds2[2]; //parent_fds a szulo csove
                                     //child_fds a gyerek csove
    int writemessage = 0;

    if(pipe(parent_fds1) == -1){ //pipe ellenorzes
        printf("Nem lehet létrehozni az elso csövet \n");
        return 1;
    }

    if(pipe(child_fds2) == -1){
        printf("Nem lehet létrehozni a masodik csövet \n");
        return 1;
    }
```

3.lépés:

Elforkol

```
pid_t pid = fork(); //elforkol
```

A pidek csővön való átadásához két darab if-es szerkezetet hoztam létre.

4.lépés:

Az első if-es szerkezet : A szülő processz az parent_fds1 csövön keresztül elküldi a pidjét a gyermeknek. A gyermek fogadja és kiírja képernyőre.

```
if(pid != 0){ // szulo processz
    close(parent_fds1[0]); //kimenet

    writemessage = getpid();

    write(parent_fds1[1], &writemessage, sizeof(writemessage));

    close(parent_fds1[1]); //bemenet
}else { // gyerek processz

    close(child_fds2[0]); //kimenet

    read(parent_fds1[0], &writemessage, sizeof(writemessage));

    printf("A gyermek megkapta pid-et a szulotol %d\n", writemessage);

    close(child_fds2[0]); // kimenet
}
```

5.lépés:

A második if-es szerkezet: A gyermek processz az child_fds2 csövön keresztül elküldi a pidjét a szülőnek. A szülő fogadja és kiírja a képernyőre.

```
if(pid ==0){ // gyerek processz
    close(child_fds2[0]); // kimenet

    writemessage = getpid();

    write(child_fds2[1], &writemessage, sizeof(writemessage));

    close(child_fds2[1]); //bemenet
}
else // szulo processz
{
    close(parent_fds1[0]); //kimenet

    read(child_fds2[0], &writemessage, sizeof(writemessage));

    printf("Megkapta a szulo a gyerek pid-jet: %d\n", writemessage);

    close(parent_fds1[0]); //kimenet
}

return 0;
}
```

A futtatás eredménye:

```
genesys@DESKTOP-3KB733B:~$ nano beadando.c
genesys@DESKTOP-3KB733B:~$ gcc beadando.c -o test
genesys@DESKTOP-3KB733B:~$ ./test
A gyermek megkapta pid-et a szulotol 94
Megkapta a szulo a gyerek pid-jet: 118
genesys@DESKTOP-3KB733B:~$
```