

JEGYZŐKÖNYV

Operációs rendszerek BSc

2021 tavasz féléves feladat

9. Gyakorlat

Készítette: **Érsek Norbert**

Neptunkód: **IIJU0Z**

A feladat leírása:

Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy csővezetékét, a gyerek processz beleír egy szöveget a csővezetékbe (A kiírt szöveg: XY neptunkod), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

A feladat elkészítésének lépései:

1. Nyilván első lépésként létrehozzuk a az *_unnamed.c fájlt.

```
pi@raspberrypi:~/Documents $ sudo nano IIJU0Z_unnamed.c
```

2. Megírjuk az egyszerű programot a pipe() használatával.

- a. A gyermek processz által kiírandó szöveget felül definiáljuk az üzenetméret konstanssal együtt.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#define MSGSIZE 16
char* msg1 = "IIJU0Z";
```

- b. A main() elején létrehozzuk az inputbuffer-t, ami később az üzenet kiírásában fog segíteni, valamint a „p” nevű egész típusú kételemű tömböt, később a pipe() -hoz használatos.

```
int main()
{
    char inputbuffer[MSGSIZE];
    int p[2];
```

- c. Ezzel eldöntjük, hogy jó-e a csővezeték. Ha nullánál kisebb értékkel tér vissza, akkor a program leáll.

```
if (pipe(p) < 0)
    exit(1)
```

- d. A gyermekprocessz kiírja az üzenetet.

```
write(p[1], msg1, MSGSIZE);
```

- e. A szülő megkapja a gyermektől az üzenetet, amit beolvas (read()). Ezt követően kiírja a standard outputra. Majd vége a programnak.

```
read(p[0], inputbuffer, MSGSIZE);
printf("%s\n", inputbuffer);
return 0;
}
```

- f. Ezt a kódot a GCC compiler-rel futtathatóvá tesszük. Jelen esetben „unnamed” lesz a projekt neve.

```
pi@raspberrypi:~/Documents $ gcc IIJU0Z_unnamed.c -o unnamed
```

- g. Később ezt a ./unnamed paranccsal tudjuk meghívni.

A futtatás eredménye: Eredményképp megkapjuk az átadott üzenetet.

```
pi@raspberrypi:~/Documents $ ./unnamed
IIJU0Z
```

A feladat leírása:

Készítsen C nyelvű programot, ahol egy szülő processz létrehoz egy nevesített csővezetékét (neve: neptunkod), a gyerek processz beleír egy szöveget a csővezetékbe (A hallgató neve), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

A feladat elkészítésének lépései:

Nagyon hasonlóak az előzőhöz, ezért csak a különbségeket sorolom fel.

1. Létrehozunk egy „x” egész típusú változót, aminek a fork() értékét adjuk.

```
int x = fork();
```

2. Ha „x” értéke kevesebb, mint nulla, akkor hiba van, leáll a program. Ha pont nulla, akkor viszont lefut ugyanúgy, mint az előző.

```
if (x < 0)
    exit(1);
if (x == 0) {
    write(p[1], msg1, MSGSIZE);

    read(p[0], inputbuffer, MSGSIZE);
    printf("%s\n", inputbuffer);
}
return 0;
}
```

A futtatás eredménye: Az előzőhöz hasonlóképpen kiírja a kért szöveget.

```
pi@raspberrypi:~/Documents $ ./named2
Ersek_Norbert
```