

## Operációs rendszerek -9. Gyakorlat

### IPC mechanizmus – PIPE (named, unnamed)

Készítette:

Dojcsák Bence

EES1D9

## Feladat 1.

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.

Mentés: neptunkod\_unnamed.c

### Forráskód:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <sys/types.h>
6  #include <sys/wait.h>
7
8  int main(int argc, char *argv[]) {
9      int fd1[2];
10     int fd2[2];
11
12     pid_t p;
13
14     if(pipe(fd1) == -1)
15     {
16         fprintf(stderr, "Pipe 1 Failed!");
17         return 1;
18     }
19
20     if(pipe(fd2) == -1)
21     {
22         fprintf(stderr, "Pipe 2 Failed!");
23         return 1;
24     }
25
26     p = fork();
27
28     if(p < 0)
29     {
30         fprintf(stderr, "Fork Failed!");
31     }
32
33     //parent process
34     else if (p > 0)
35     {
36         char get_str[100];
37
38         wait(NULL);
39
40         close(fd2[1]);
41
42         read(fd2[0], get_str, 100);
43         printf("String I've got from Child process: \"%s\".\n", get_str);
44         close(fd2[0]);
45     }
46
47     //child process
48     else
49     {
50         char fixed_str[] = "Dojcsak Bence, EES1D9";
51
52         close(fd1[0]);
53         close(fd1[1]);
54         close(fd2[0]);
55
56         write(fd2[1], fixed_str, strlen(fixed_str)+1);
57         close(fd2[1]);
58
59         exit(0);
60     }
61 }
```

### Futtatás:

```
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ nano EES1D9_unnamed.c
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ gcc EES1D9_unnamed.c -o ees1d9_unnamed
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ ./ees1d9_unnamed
String I've got from Child process: "Dojcsak Bence EES1D9".
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$
```

## Feladat 2.

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:pl. Keserű Ottó), a szülő processz ezt kiolvassa, és kiírja a standard kimenetre.

Mentés: neptunkod\_named.c

### Forráskód:

```
1 #include <stdio.h>
2 #include <fcntl.h>
3 #include <unistd.h>
4 #include <string.h>
5 #include <sys/types.h>
6 #include <sys/wait.h>
7
8 int main(int argc, char *argv[]) {
9     char* myfifo = "/temp/ees1d9";
10
11     mkfifo(myfifo, 0666); // mkfifo(<pathname>, <permission>) - Creating a named file (FIFO)
12
13     int fd;
14
15     char str1[] = "A hallgato neve: Dojcsak Bence";
16     char str2[80];
17
18     pid_t p;
19
20     p = fork();
21
22     if (p < 0)
23     {
24         fprintf(stderr, "Fork Failed!");
25     }
26
27     //parent process
28     else if (p > 0)
29     {
30         while(1)
31         {
32             fd = open(myfifo, O_RDONLY); //open in read only mode and read
33             read(fd, str1, 80);
34
35             printf("String from Child process: \"%s\".\n", str1);
36             close(fd);
37
38             fd = open(myfifo, O_WRONLY); //open in write mode, get string, and write, then close
39             fgets(str2, 80 stdin);
40
41             write(fd, str2, strlen(str2)+1);
42             close(fd);
43         }
44
45         return 0;
46     }
47
48     //child process
49     else
50     {
51         while(1)
52         {
53             fd = open(myfifo, O_WRONLY); //open in write mode, get string, and write, then close
54             write(fd, str1, strlen(str1)+1);
55
56             fgets(str1, 80 stdin);
57             close(fd);
58
59             fd = open(myfifo, O_RDONLY); //open in read only mode and read
60             read(fd, str2, 80);
61
62             printf("String from Child process: \"%s\".\n", str2);
63             close(fd);
64         }
65         return 0;
66     }
67 }
68
```

### Futtatás:

```
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ nano ees1d9_named.c
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ gcc ees1d9_named.c -o ees1d9_named
debian@debian10:~/Desktop/OS_Gyak/EES1D9_2021_04_13_Gyakorlat_9$ ./ees1d9_named
Child process: "A hallgato neve: Dojcsak Bence."
```

### Feladat 3.

*Írjon C nyelvű programot, amelyik kill() seg.-vel SIGALRM-et küld egy argumentumként megadott PID-u processznek, egy másik futó program a SIGALRM-hez rendeljen egy fv.-t amely kiírja pl.neptunkodot, továbbá pause() fv.-el blokkolódjon, majd kibillenés után jelezze, hogy kibillent és terminálódjon.*

Mentés. neptunkod\_gyak9\_3.

**Forráskód:**

**Futtatás:**

**Feladat 4.**

*Írjon C nyelvű programot, amelyik a SIGTERM-hez hozzárendel egy fv-t., amelyik kiírja az int paraméter értéket, majd végtelen ciklusban fusson, 3 sec-ig állandóan blokkolódva elindítás után egy másik shell-ben kill paranccsal (SIGTERM) próbálja terminálni, majd SIGKILL-el.*

Mentés. neptunkod\_gyak9\_4.c

**Forráskód:**

**Futtatás:**