



UNIVERSITETI POLITEKNIK I TIRANËS  
FAKULTETI I TEKNOLOGJISË DHE INFORMACIONIT  
DEPARTAMENTI I INXHINIERISË INFORMATIKE

## **Punë Laboratori nr. 4**

**Lënda:** Sisteme Operative

**Grupi:** III-B

**Tema:** Përdorimi i **pipe()** në komunikimin e dy proceseve

**Punoi:**  
Piro Gjikhima

**Pranoi:**  
MSc.Megi Tartari

## Ushtrimi 1

Nderto nje program ne gjuhen C qe perdor thirrjen e sistemit `pipe()` per te krijuar një pipe ku procesi femije afishon ne ekran mesataren e numrave random qe gjeneron procesi prind.

```
Lab 4 > C usht1.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/wait.h>
5  #include <time.h>
6
7  int generateRandomNumber(int min, int max);
8
9  int main() {
10
11     srand(time(0));
12
13     int p[2];
14     pipe(p);
15     pid_t p1 = fork();
16
17     if (p1 == 0) {
18         close(p[1]);
19
20         int n;
21         float sum = 0;
22         int length = 0;
23
24         for (int i = 0; i < 100; i++) {
25             if (read(p[0], &n, sizeof(int)) == -1) {
26                 perror("Write failed");
27                 return 1;
28             }
29             sum += n;
30             length++;
31         }
32
33         close(p[0]);
34
35         float mesatarja = sum / length;
36
37         printf("Shuma: %.0f\n", sum);
38         printf("Mesatarja: %.2f\n", mesatarja);
39     }
40     else {
41
42         close(p[0]);
43
44         for (int i = 0; i < 10; i++) {
45             int random = generateRandomNumber(1, 1000);
46
47             if (write(p[1], &random, sizeof(int)) == -1) {
48                 perror("Write failed");
49                 return 1;
50             }
51         }
52
53         close(p[1]);
54         wait(NULL);
55     }
56
57     return 0;
58 }
59
60
61 int generateRandomNumber(int min, int max) {
62     return min + rand() % (max - min + 1);
63 }
64
```

Ky program kryen komunikim ndërproces duke përdorur pipe. Në procesin e prindit, gjenerohen 10 numra rastësorë midis 1 dhe 1000 dhe shkruhen në pipe. Procesi i fëmijës lexon këta 100 numra nga pipe, llogarit shumën dhe mesataren e tyre, dhe pastaj i shfaq këto rezultate.

Hapat specifike përfshijnë:

1. Krijimin e një pipe për komunikim
2. Krijimin e një procesi të ri përmes fork()
3. Në procesin e fëmijës:
  - Mbyllet skaji i shkrimit i pipe-s
  - Lexohen 100 numra nga pipe
  - Llogaritet shuma dhe gjatësia e numrave
  - Llogaritet mesatarja
  - Shfaqen shuma dhe mesatarja
4. Në procesin e prindit:
  - Mbyllet skaji i leximit i pipe-s
  - Gjenerohen 10 numra rastësorë
  - Shkruhen numrat në pipe
  - Pritet përfundimi i procesit të fëmijës

**Output:**

```
[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht1.c -o usht1 && "/home/kali/Documents/Lab 4/"usht1
Shuma: 85578
Mesatarja: 855.78

[Done] exited with code=0 in 0.133 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht1.c -o usht1 && "/home/kali/Documents/Lab 4/"usht1
Shuma: 92780
Mesatarja: 927.80

[Done] exited with code=0 in 0.115 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht1.c -o usht1 && "/home/kali/Documents/Lab 4/"usht1
Shuma: 92780
Mesatarja: 927.80

[Done] exited with code=0 in 0.154 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht1.c -o usht1 && "/home/kali/Documents/Lab 4/"usht1
Shuma: 77287
Mesatarja: 772.87

[Done] exited with code=0 in 0.129 seconds
```

## Ushtrimi 2

Nderto nje program ne gjuhen C qe perdor thirrjen e sistemit pipe( ) per te krijuar një pipe ku procesi femije llogarit mesataren e numrave random qe gjeneron procesi prind dhe ia degjon procesit prind per ta afishuar ne ekran.

```
ab4 > C ushi2.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <sys/wait.h>
5  #include <time.h>
6
7  int generateRandomNumber(int min, int max) {
8      return min + rand() % (max - min + 1);
9  }
10
11 int main() {
12
13     srand(time(0));
14
15     int p[2], p2[2];
16     pipe(p); pipe(p2);
17     pid_t p1 = fork();
18
19     if (p1 == 0) {
20
21         close(p[1]);
22         close(p2[0]);
23
24         int n;
25         float sum = 0;
26         int length;
27
28         for (length = 0; length < 10000; length++) {
29             if (read(p[0], &n, sizeof(int)) == -1) {
30                 perror("Read failed");
31                 return 1;
32             }
33             sum += n;
34         }
35         close(p[0]);
36
37         float mesatarja = sum / length;
38
39         write(p2[1], &mesatarja, sizeof(int));
40         close(p2[1]);
41         exit(0);
42     }
43     else {
44
45         close(p[0]);
46         close(p2[1]);
47
48         for (int i = 0; i < 10000; i++) {
49             int random = generateRandomNumber(1, 10000);
50
51             if (write(p[1], &random, sizeof(int)) == -1) {
52                 perror("Write failed");
53                 return 1;
54             }
55         }
56     }
57     close(p[1]);
58
59     wait(NULL);
60
61     float mesatarja;
62     read(p2[0], &mesatarja, sizeof(int));
63     close(p2[0]);
64
65     printf("Mesatarja: %.2f", mesatarja);
66 }
67 return 0;
68 }
```

Ky program kryen këto hapa kryesorë:

Programi gjeneron 10,000 numra rastësorë midis 1 dhe 10,000. Ai përdor dy pipe komunikimi midis proceseve prind dhe fëmijë. Në procesin e prindit, gjenerohen numra rastësorë dhe shkruhen në pipe e parë. Procesi i fëmijës lexon këta numra nga pipe i parë, llogarit mesataren e tyre dhe pastaj shkruan këtë mesatare në pipe e dytë. Procesi i prindit lexon mesataren nga pipe i dytë dhe e shfaq atë.

Hapat specifikë të programit janë:

1. Krijimi i dy pipe për komunikim ndërproces

2. Krijimi i një procesi të ri (fork)

3. Në procesin e fëmijës:

- Mbyllen skajet e panevojshme të pipe
- Lexohen 10,000 numra nga pipe i parë
- Llogaritet shuma e këtyre numrave
- Shkruhet mesatarja në pipe e dytë

4. Në procesin e prindit:

- Mbyllen skajet e panevojshme të pipe
- Gjenerohen 10,000 numra rastësorë
- Shkruhen këta numra në pipe e parë
- Pritet përfundimi i procesit të fëmijës
- Lexohet mesatarja nga pipe i dytë
- Shfaqet mesatarja përfundimtare

Kodi demonstroi komunikimin ndërproces duke përdorur pipe, gjenerimin e numrave rastësorë dhe sinkronizimin bazë të proceseve përmes fork() dhe wait().

## Output:

```
[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht2.c -o usht2 && "/home/kali/Documents/Lab 4/"usht2
Mesatarja: 4994.02
[Done] exited with code=0 in 0.134 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht2.c -o usht2 && "/home/kali/Documents/Lab 4/"usht2
Mesatarja: 4994.02
[Done] exited with code=0 in 0.111 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht2.c -o usht2 && "/home/kali/Documents/Lab 4/"usht2
Mesatarja: 4995.17
[Done] exited with code=0 in 0.111 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht2.c -o usht2 && "/home/kali/Documents/Lab 4/"usht2
Mesatarja: 4977.18
[Done] exited with code=0 in 0.115 seconds

[Running] cd "/home/kali/Documents/Lab 4/" && gcc usht2.c -o usht2 && "/home/kali/Documents/Lab 4/"usht2
Mesatarja: 5025.82
[Done] exited with code=0 in 0.213 seconds
```