

Προγραμματισμός Πυρήνα  
Συστημάτων  
Εργαστηριακές Σημειώσεις - 3η άσκηση

ΕΡΓΑΣΤΗΡΙΟ Λ.Σ.

Εαρινό 2017

# 1 Το πρόγραμμα main

Listing 1: main.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<unistd.h>
4  #include<sys/wait.h>
5  #include <string.h>
6  #include <fcntl.h>
7
8  #define READ 0
9  #define WRITE 1
10
11 int main()
12 {
13     //from child to parent and vice versa
14     int fromChild[2], fromParent[2];
15     int status, ret;
16     if (pipe(fromChild) == -1) {
17         perror("pipe");
18         exit(1);
19     }
20     if (pipe(fromParent) == -1) {
21         perror("pipe");
22         exit(1);
23     }
24     if ((ret = fork()) == -1) {
25         perror("fork");
26         exit(1);
27     }
28     if (ret == 0) { //child proc
29         char c, cl;
30         printf("\nthis is child with: PID=%d and PPID=%d\n", getpid(),
31             getppid());
32         close(fromChild[READ]);
33         close(fromParent[WRITE]);
34         int fd = open("temp", O_RDONLY);
35         if (fd == -1) {
36             perror("open");
37             exit(1);
38         }
39         while (read(fd, &c, sizeof(char)) > 0) {
40             //read one byte at a time from the file and send it to parent process
41             write(fromChild[WRITE], &c, sizeof(char));
42         }
43         // write has finished close writing end
44         close(fromChild[WRITE]);
45         sleep(1);
46         printf("\nChild process received from parent:\n");
47         while (read(fromParent[READ], &cl, sizeof(char)) > 0) {
48             // read one byte at a time from parent process and
49             write(1, &cl, sizeof(char)); // print it
50         }
51     }
```

Listing 2: main.c

```

51     printf("\n");
52     // reading has finished close reading end
53     close(fromParent[READ]);
54     return 0;
55 } else { //parent proc
56     char msg2[21]; //2^64-1 +'\0' -> 21 characters max length
57     char cr;
58     long i = 0;
59     printf("\n_this_is_parent_with:_PID=%d_and_PPID=%d\n", getpid(),
60           getppid());
61     close(fromParent[READ]);
62     close(fromChild[WRITE]);
63     sleep(1);
64     printf("_parent_process_received_from_child:_\n");
65     while (read(fromChild[READ], &cr, sizeof(char)) > 0) {
66         //read one byte at a time from child process and
67         write(1, &cr, sizeof(char)); // print it
68         i++; //increase char counter
69     }
70     close(fromChild[READ]); // reading has finished close reading end
71     printf("\n");
72     snprintf(msg2, 20, "%ld", i); // transform long value to string
73     write(fromParent[WRITE], msg2, strlen(msg2)); // write length value to pipe
74     close(fromParent[WRITE]); // write has finished close writing end
75     wait(&status); // wait for child process to end
76     printf("child_process_has_finished_with_code:_%d", status >> 8);
77 }
78 return 0;
79 }

```

## 2 Το πρόγραμμα server

Listing 3: server.c

```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <unistd.h>
5 #include <string.h>
6
7 int main(int argc, char *argv[])
8 {
9     int client_to_server;
10    char *myfifo = "/tmp/client_to_server_fifo";
11    int server_to_client;
12    char *myfifo2 = "/tmp/server_to_client_fifo";
13    char buf[BUFSIZ];
14    char str[BUFSIZ];
15
16    /* create the FIFO (named pipe) */
17    mkfifo(myfifo, 0666);
18    mkfifo(myfifo2, 0666);
19
20    /* open, read, and display the message from the FIFO */
21    client_to_server = open(myfifo, O_RDONLY);
22    server_to_client = open(myfifo2, O_WRONLY);
23
24    printf("Server_ON.\n");
25
26    while (1) {
27        read(client_to_server, buf, BUFSIZ);
28        if (strcmp("exit", buf) == 0) {
29            printf("Server_OFF.\n");
30            break;
31        } else if (strcmp("", buf) != 0) {
32            printf("Received:_%s\n", buf);
33            printf("Reply_...\n");
34            fgets(str, 1023, stdin);
35            write(server_to_client, str, sizeof(str));
36        }
37        /* clean buf from any data */
38        memset(buf, 0, sizeof(buf));
39    }
40
41    close(client_to_server);
42    close(server_to_client);
43    unlink(myfifo);
44    unlink(myfifo2);
45    return 0;
46 }
```

### 3 Το πρόγραμμα client

Listing 4: client.c

```
1  #include <sys/stat.h>
2  #include <sys/types.h>
3  #include <unistd.h>
4  #include <stdio.h>
5  #include <fcntl.h>
6  #include <string.h>
7
8  int main(int argc, char *argv[])
9  {
10     int client_to_server;
11     char *myfifo = "/tmp/client_to_server_fifo";
12
13     int server_to_client;
14     char *myfifo2 = "/tmp/server_to_client_fifo";
15
16     char str[BUFSIZ];
17     char str2[BUFSIZ];
18
19     client_to_server = open(myfifo, O_WRONLY);
20     server_to_client = open(myfifo2, O_RDONLY);
21
22     /* write str to the FIFO */
23
24     while (1) {
25
26         printf("Input_message_to_server:");
27         fgets(str, 1023, stdin);
28         write(client_to_server, str, sizeof(str));
29         read(server_to_client, str2, sizeof(str2));
30
31         printf("...received_from_the_server: %s\n", str2);
32         if (strcmp("exit", str2) == 0) {
33             printf("Client_OFF.\n");
34             break;
35         }
36
37         /* clean buf from any data */
38         memset(str, 0, sizeof(str));
39         memset(str2, 0, sizeof(str2));
40
41     }
42     close(client_to_server);
43     close(server_to_client);
44
45     /* remove the FIFO */
46
47     return 0;
48 }
```