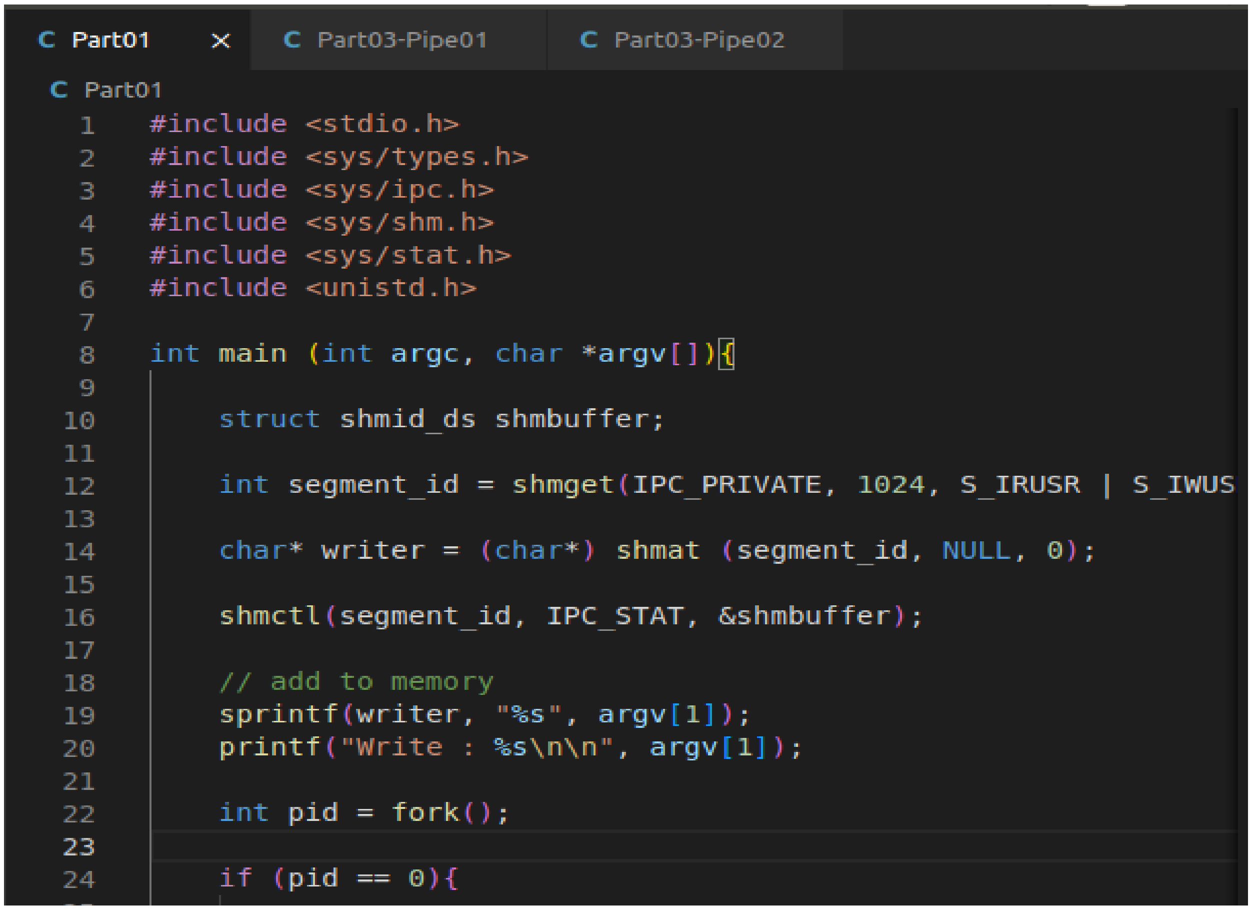
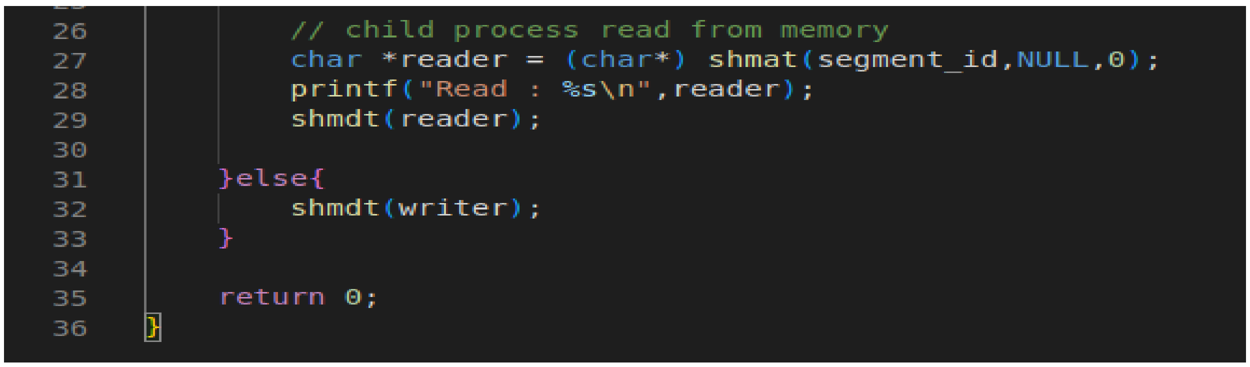
محمد مهدی نظری ۹۹۳۱۰۶۱ – امید خباز قانع ۹۹۳۱۰۱۸

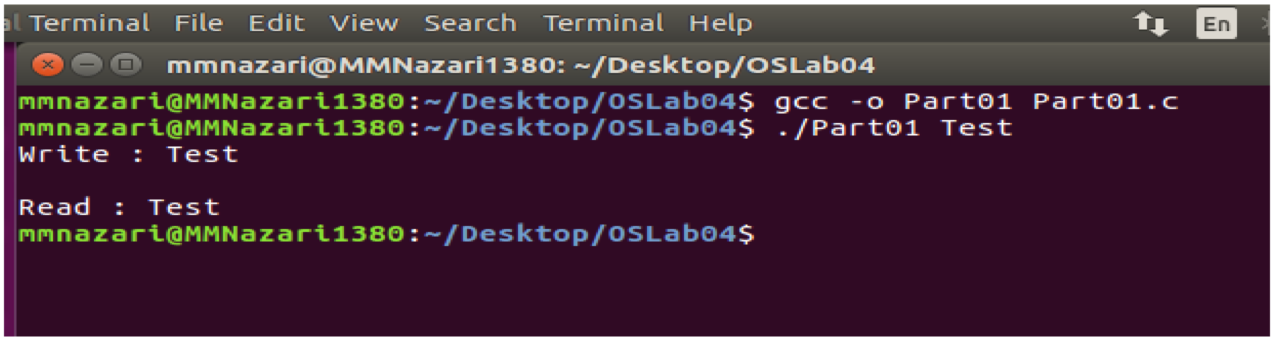
<u>بخش اول :</u>



```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <unistd.h>

int main (int argc, char *argv[]){

    struct shmid_ds shmbuffer;

    int segment_id = shmget(IPC_PRIVATE, 1024, S_IRUSR | S_IWUS

    char* writer = (char*) shmat (segment_id, NULL, 0);

    shmctl(segment_id, IPC_STAT, &shmbuffer);

    // add to memory
    sprintf(writer, "%s", argv[1]);
    printf("Write : %s\n\n", argv[1]);

    int pid = fork();

    if (pid == 0){
```



```c
        // child process read from memory
        char *reader = (char*) shmat(segment_id,NULL,0);
        printf("Read : %s\n",reader);
        shmdt(reader);

    }else{
        shmdt(writer);
    }

    return 0;
}
```


Terminal File Edit View Search Terminal Help
En
mmnazari@MMNazari1380: ~/Desktop/OSLab04

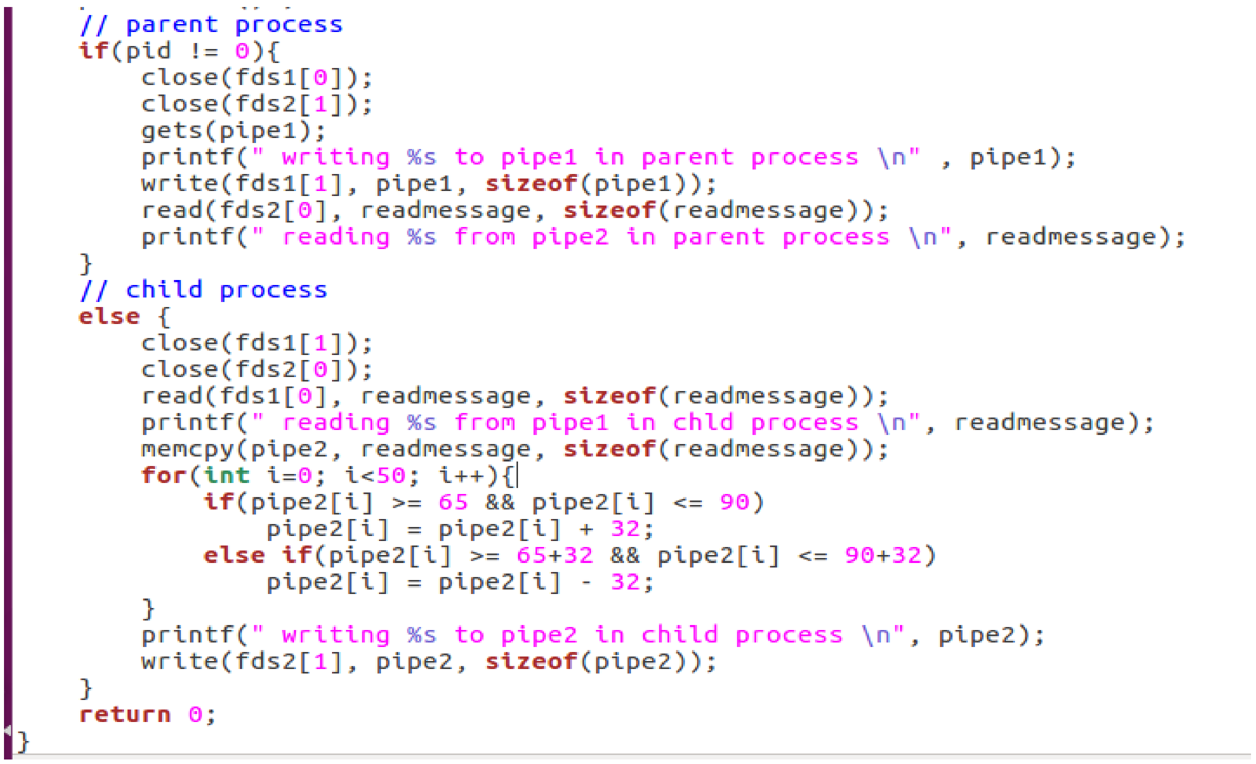
```
mmnazari@MMNazari1380:~/Desktop/OSLab04$ gcc -o Part01 Part01.c
mmnazari@MMNazari1380:~/Desktop/OSLab04$ ./Part01 Test
Write : Test

Read : Test
mmnazari@MMNazari1380:~/Desktop/OSLab04$
```

بخش سوم :

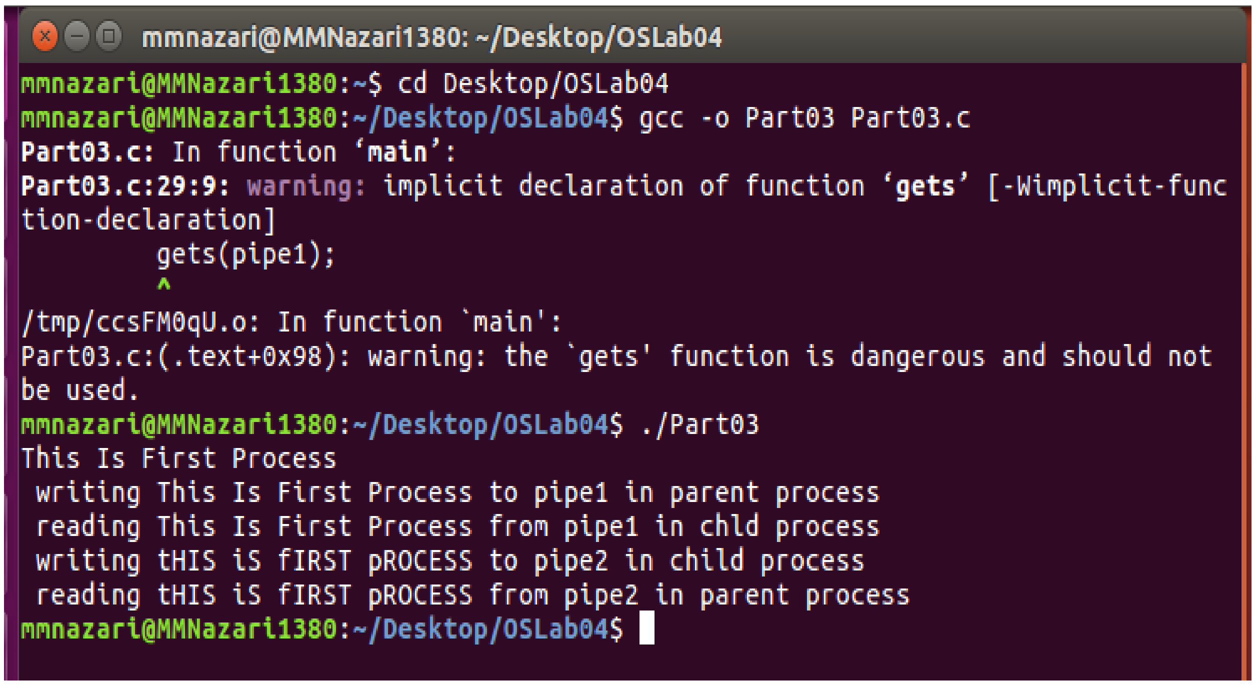
این بخش با دو روش پیادده سازی شده که روش اول به نتیجه صحیح مارا میرساند اما روش دوم نیاز به کار بیشتر دارد.

روش اول : فایل Part03.c

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include <pthread.h>

int main() {
    int fds1[2], fds2[2];
    char pipe1[50];
    char pipe2[50];
    char readmessage[50];
    int pid ;
    if(pipe(fds1)==-1){
        exit(EXIT_FAILURE);
    }
    if(pipe(fds2)==-1){
        exit(EXIT_FAILURE);
    }
    pid = fork() ;
```



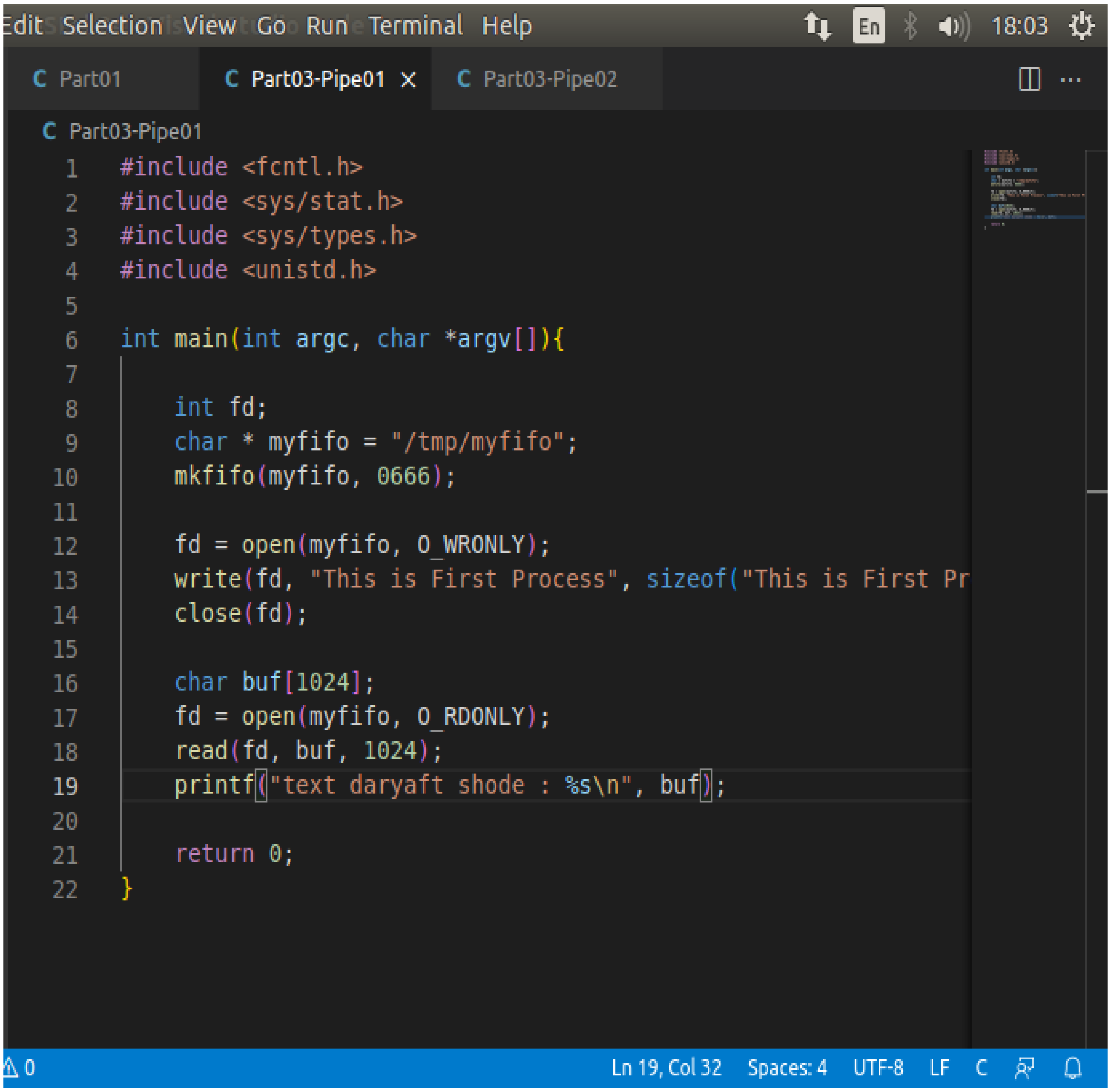
```c
    // parent process
    if(pid != 0){
        close(fds1[0]);
        close(fds2[1]);
        gets(pipe1);
        printf(" writing %s to pipe1 in parent process \n" , pipe1);
        write(fds1[1], pipe1, sizeof(pipe1));
        read(fds2[0], readmessage, sizeof(readmessage));
        printf(" reading %s from pipe2 in parent process \n", readmessage);
    }
    // child process
    else {
        close(fds1[1]);
        close(fds2[0]);
        read(fds1[0], readmessage, sizeof(readmessage));
        printf(" reading %s from pipe1 in chld process \n", readmessage);
        memcpy(pipe2, readmessage, sizeof(readmessage));
        for(int i=0; i<50; i++){
            if(pipe2[i] >= 65 && pipe2[i] <= 90)
                pipe2[i] = pipe2[i] + 32;
            else if(pipe2[i] >= 65+32 && pipe2[i] <= 90+32)
                pipe2[i] = pipe2[i] - 32;
        }
        printf(" writing %s to pipe2 in child process \n", pipe2);
        write(fds2[1], pipe2, sizeof(pipe2));
    }
    return 0;
}
```



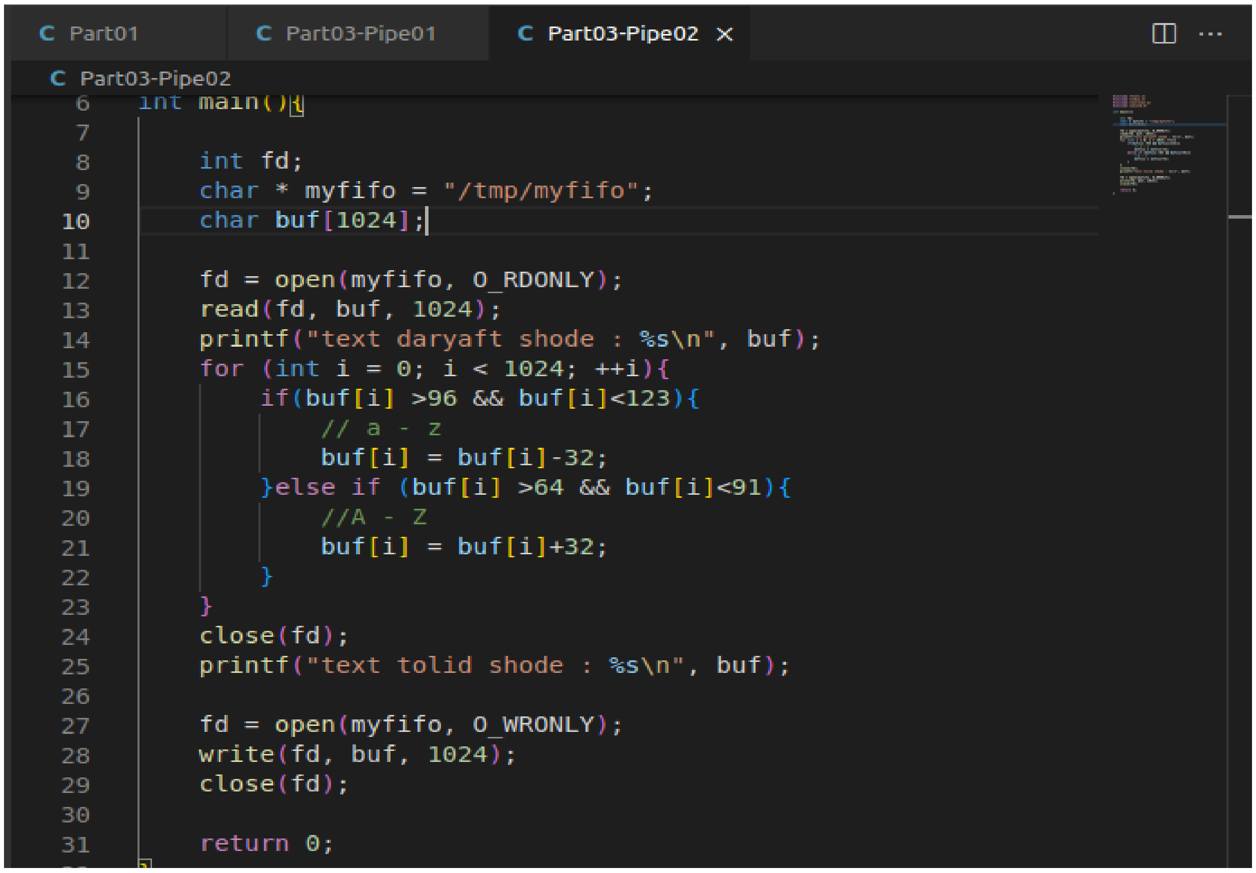
mmnazari@MMNazari1380: ~/Desktop/OSLab04

```
mmnazari@MMNazari1380:~$ cd Desktop/OSLab04
mmnazari@MMNazari1380:~/Desktop/OSLab04$ gcc -o Part03 Part03.c
Part03.c: In function ‘main’:
Part03.c:29:9: warning: implicit declaration of function ‘gets’ [-Wimplicit-func
tion-declaration]
         gets(pipe1);
         ^
/tmp/ccsFM0qU.o: In function `main':
Part03.c:(.text+0x98): warning: the `gets' function is dangerous and should not
be used.
mmnazari@MMNazari1380:~/Desktop/OSLab04$ ./Part03
This Is First Process
 writing This Is First Process to pipe1 in parent process
 reading This Is First Process from pipe1 in chld process
 writing tHIS iS fIRST pROCESS to pipe2 in child process
 reading tHIS iS fIRST pROCESS from pipe2 in parent process
mmnazari@MMNazari1380:~/Desktop/OSLab04$
```

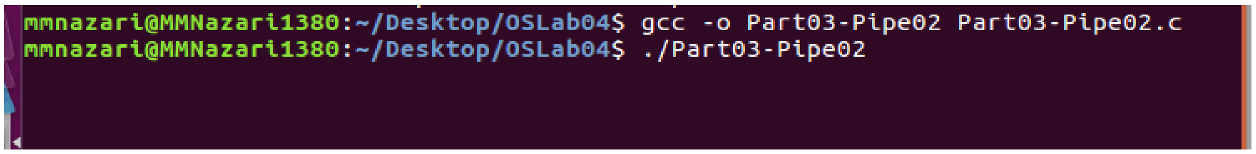
روش دوم : فایل های Part03-Pipe01 و Part03-Pipe02


Edit Selection View Go Run Terminal Help
En 18:03
Part01
Part03-Pipe01
Part03-Pipe02
Part03-Pipe01
⚠ 0
Ln 19, Col 32 Spaces: 4 UTF-8 LF C

```c
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char *argv[]){

    int fd;
    char * myfifo = "/tmp/myfifo";
    mkfifo(myfifo, 0666);

    fd = open(myfifo, O_WRONLY);
    write(fd, "This is First Process", sizeof("This is First Pr
    close(fd);

    char buf[1024];
    fd = open(myfifo, O_RDONLY);
    read(fd, buf, 1024);
    printf("text daryaft shode : %s\n", buf);

    return 0;
}
```



```c
int main(){

    int fd;
    char * myfifo = "/tmp/myfifo";
    char buf[1024];

    fd = open(myfifo, O_RDONLY);
    read(fd, buf, 1024);
    printf("text daryaft shode : %s\n", buf);
    for (int i = 0; i < 1024; ++i){
        if(buf[i] >96 && buf[i]<123){
            // a - z
            buf[i] = buf[i]-32;
        }else if (buf[i] >64 && buf[i]<91){
            //A - Z
            buf[i] = buf[i]+32;
        }
    }
    close(fd);
    printf("text tolid shode : %s\n", buf);

    fd = open(myfifo, O_WRONLY);
    write(fd, buf, 1024);
    close(fd);

    return 0;
```



```
mmnazari@MMNazari1380:~/Desktop/OSLab04$ gcc -o Part03-Pipe02 Part03-Pipe02.c
mmnazari@MMNazari1380:~/Desktop/OSLab04$ ./Part03-Pipe02
```

در روش دوم ترمینال پیغامی چاپ نمیکند.