OS Lab 9

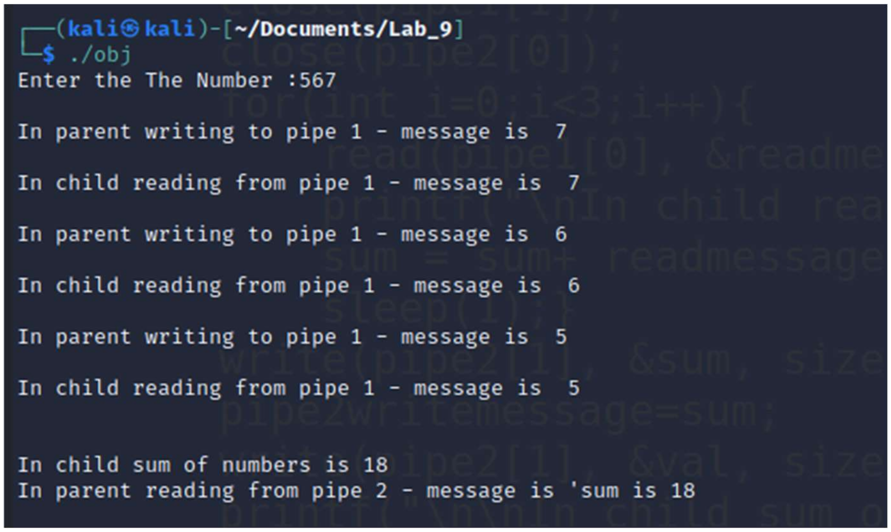
11762 Muhammad Kashif

Task 1

```
#include <stdio.h>
#include<unistd.h>
int main(){
    int pipe1[2], pipe2[2];
    int returnstatus1, returnstatus2;
    int pid;
    int pipe1writemessage;
    int pipe2writemessage;
    int readmessage;
    int val;
    int sum =0;
    returnstatus1 = pipe(pipe1);
    if(returnstatus1 == -1){
        printf("Unable to create pipe1");
        return 1;}
    returnstatus2 = pipe(pipe2);
    if(returnstatus2 == -1){
        printf("Unable to create pipe2");
        return 1;}
    pid = fork();
    if(pid != 0){
        close(pipe1[0]);
        close(pipe2[1]);
        printf("Enter the The Number :");
        scanf("\n%d",&pipe1writemessage);
        for(int i=0;i<3;i++){
            val =pipe1writemessage %10;
            write(pipe1[1], &val, sizeof(val));
            pipe1writemessage=pipe1writemessage/10;
            printf("\nIn parent writing to pipe 1 - message is  %d\n", val);
            sleep(1);}
            read(pipe2[0], &readmessage, sizeof(readmessage));
            printf("\nIn parent reading from pipe 2 - message is 'sum is %d\n", readmessage);}
    else{
        close(pipe1[1]);
        close(pipe2[0]);
        for(int i=0;i<3;i++){
            read(pipe1[0], &readmessage, sizeof(pipe1writemessage));
            printf("\nIn child reading from pipe 1 - message is  %d\n", readmessage);
            sum = sum+ readmessage;
            sleep(1);}
        write(pipe2[1], &sum, sizeof(sum));
        pipe2writemessage=sum;
        write(pipe2[1], &val, sizeof(pipe2writemessage));
        printf("\n\nIn child sum of numbers is %d", sum);}
    return 0;}#include <stdio.h>
#include<unistd.h>
```

```
int main(){
    int pipe1[2], pipe2[2];
    int returnstatus1, returnstatus2;
    int pid;
    int pipe1writemessage;
    int pipe2writemessage;
    int readmessage;
    int val;
    int sum =0;
    returnstatus1 = pipe(pipe1);
    if(returnstatus1 == -1){
        printf("Unable to create pipe1");
        return 1;}
    returnstatus2 = pipe(pipe2);
    if(returnstatus2 == -1){
        printf("Unable to create pipe2");
        return 1;}
    pid = fork();
    if(pid != 0){
        close(pipe1[0]);
        close(pipe2[1]);
        printf("Enter the The Number :");
        scanf("\n%d",&pipe1writemessage);
        for(int i=0;i<3;i++){
            val =pipe1writemessage %10;
            write(pipe1[1], &val, sizeof(val));
            pipe1writemessage=pipe1writemessage/10;
            printf("\nIn parent writing to pipe 1 - message is  %d\n", val);
            sleep(1);}
            read(pipe2[0], &readmessage, sizeof(readmessage));
            printf("\nIn parent reading from pipe 2 - message is 'sum is %d\n", readmessage);}
    else{
        close(pipe1[1]);
        close(pipe2[0]);
        for(int i=0;i<3;i++){
            read(pipe1[0], &readmessage, sizeof(pipe1writemessage));
            printf("\nIn child reading from pipe 1 - message is  %d\n", readmessage);
            sum = sum+ readmessage;
            sleep(1);}
        write(pipe2[1], &sum, sizeof(sum));
        pipe2writemessage=sum;
        write(pipe2[1], &val, sizeof(pipe2writemessage));
        printf("\n\nIn child sum of numbers is %d", sum);}
    return 0;}
```

Output



```
┌──(kali㉿kali)-[~/Documents/Lab_9]
└─$ ./obj
Enter the The Number :567

In parent writing to pipe 1 - message is  7

In child reading from pipe 1 - message is  7

In parent writing to pipe 1 - message is  6

In child reading from pipe 1 - message is  6

In parent writing to pipe 1 - message is  5

In child reading from pipe 1 - message is  5


In child sum of numbers is 18
In parent reading from pipe 2 - message is 'sum is 18
```