INTER-PROCESS COMMUNICATION

LAB # 10

Fall 2020

CSE302L System Programming Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _______________

Submitted to:

Engr. Madiha Sher

Tuesday, February 23rd, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task # 01:

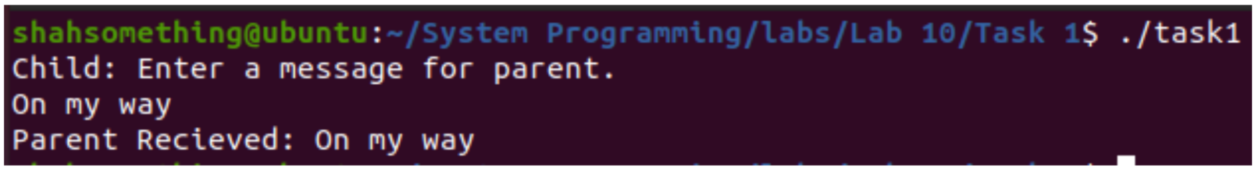
A program in which a child writes a string to a pipe and the parent reads the string.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <string.h>

int main(int argc, char *argv[])
{
   int fd[2];
   int ret = pipe(fd);
   if(ret==-1)
   {
      perror("Failed to create a pipe");
      return -1;
   }
   int x = fork();
   if(x==-1)
   {
      perror("Failed to create a child process");
      return -1;
   }
   if(x==0)
   {
      char buffer[100];
      printf("Child: Enter a message for parent.\n");
      fgets(buffer,200,stdin);
      write(fd[1],buffer,strlen(buffer));
   }else{
      wait(NULL);
      char buffer[200];
      read(fd[0],buffer,sizeof(buffer));
      printf("Parent Recieved: %s",buffer);
   }
}
```

Output/Results:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 10/Task 1$ ./task1
Child: Enter a message for parent.
On my way
Parent Recieved: On my way
```

Task # 02:

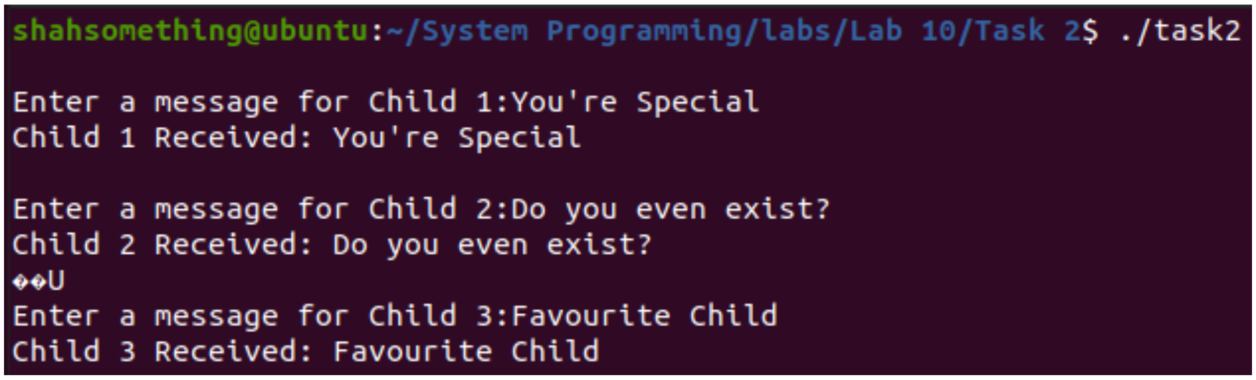
Write a program that creates a process fan. Parent process writes to the pipe and all the child processes read the message from pipe and display it on stdout.

Code:

```
#include <stdio.h>
#include <unistd.h>
#include<sys/wait.h>
#include<string.h>

int main(int argc, char *argv[])
{
    int fd[2];
    int ret = pipe(fd);
    char buffer[100];
    if(ret==-1)
    {
        perror("Failed to create a pipe");
        return -1;
    }
    for(int i=0;i<3;i++)
    {
        int x = fork();
        if(x==-1)
        {
            perror("Failed to create a child process");
            return -1;
        }
        if(x>0)
        {
            printf("\nEnter a message for Child %d:",i+1);
            fgets(buffer,sizeof(buffer),stdin);
            write(fd[1],buffer,strlen(buffer));
            wait(NULL);
        }else{
            read(fd[0],buffer,sizeof(buffer));
            printf("Child %d Received: %s",i+1,buffer);
            break;
        }
    }
}
```

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 10/Task 2$ ./task2

Enter a message for Child 1:You're Special
Child 1 Received: You're Special

Enter a message for Child 2:Do you even exist?
Child 2 Received: Do you even exist?
��U
Enter a message for Child 3:Favourite Child
Child 3 Received: Favourite Child
```

Task # 03:

Chatting between two process using FIFO

Write a Chatting application in which two processes can communicate using FIFO. Your program should satisfy the following specifications. The program should take the name of FIFO, will create the FIFO (if not created yet) and should open it for reading and writing. Program should take input from standard input and write it to FIFO and should read from FIFO and write to standard output in another process. Both reading and writing shall be done concurrently.

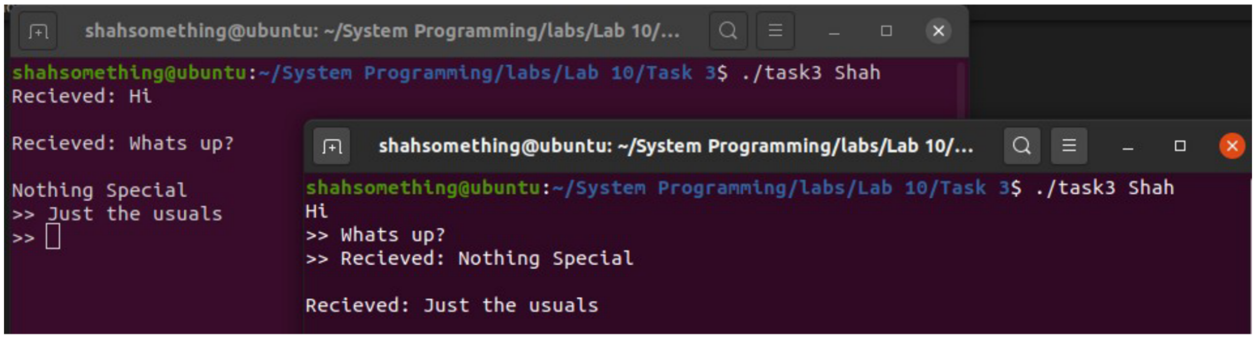
Code:

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <sys/select.h>
#include <string.h>
#include <sys/stat.h>
#include <errno.h>

int main(int argc, char *argv[])
{
    if(argc!=2)
    {
        printf("Fifo name not provided\n");
        return -1;
    }
    int fd,w,r,ret;
    if(mkfifo(argv[1],S_IRWXU)==-1)
    {
        if(errno!=EEXIST)
        {
            perror("Failed to create fifo\n");
            return -1;
        }
    }
    fd = open(argv[1],O_RDWR);
    if(fd==-1)
    {
        perror("Failed to open fifo\n");
        return -1;
```

```
}
char str[512];
char str2[4] = ">> ";
fd_set readset;
sleep(4);
while(1)
{
    FD_ZERO(&readset);
    FD_SET(fd,&readset);
    FD_SET(STDIN_FILENO,&readset);
    ret = select(fd+1,&readset,NULL,NULL,NULL);
    if(ret==-1)
    {
        perror("Error Using Select");
        return -1;
    }
    if(FD_ISSET(STDIN_FILENO,&readset))
    {
        fprintf(stderr, ">> ");
        r = read(STDIN_FILENO,str,sizeof(str));
        w = write(fd,str,r);
        if(w==-1)
        {
            perror("Error while writing to fifo");
            return -1;
        }
        sleep(4);
        continue;
    }
    if(FD_ISSET(fd,&readset))
    {
        r = read(fd,str,sizeof(str));
        if(r==-1)
        {
            perror("Error while reading from fifo");
            return -1;
        }else if(r==0)
            continue;
        fprintf(stderr,"Recieved: %s\n",str);
    }
    }
}
```

Output:



```
shahsomething@ubuntu:~/System Programming/labs/Lab 10/Task 3$ ./task3 Shah
Recieved: Hi

Recieved: Whats up?

Nothing Special
>> Just the usuals
>> 
```

```
shahsomething@ubuntu:~/System Programming/labs/Lab 10/Task 3$ ./task3 Shah
Hi
>> Whats up?
>> Recieved: Nothing Special

Recieved: Just the usuals
```