122CS0067

Assignment-V
<u>Name:</u> **Amiya Chowdhury**   <u>Roll:</u> **122CS0067**   <u>Date:</u> **09/09/2024**

1. Write a program to implement a two-way continuous chat between two separately executing processes.

lab5_1w.c

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

int main(){
    int res;
    char buffer[50];
//first file
    res=mkfifo("fifo1",0777);
    if(res==0) printf("named pipe created successfully!\n");
//second
    res=mkfifo("fifo2",0777);
    if(res==0) printf("named pipe created successfully!\n");
//opening
    int n,fd,fd2;
    fd=open("fifo1",O_WRONLY);
    fd2=open("fifo2",O_RDONLY);
    while(1){
    fgets(buffer, 50, stdin);
    n=write(fd,buffer,strlen(buffer));
    //
    n=read(fd2,buffer,50);
    if(n) printf("%s",buffer);

    }
    return 0;
}
```
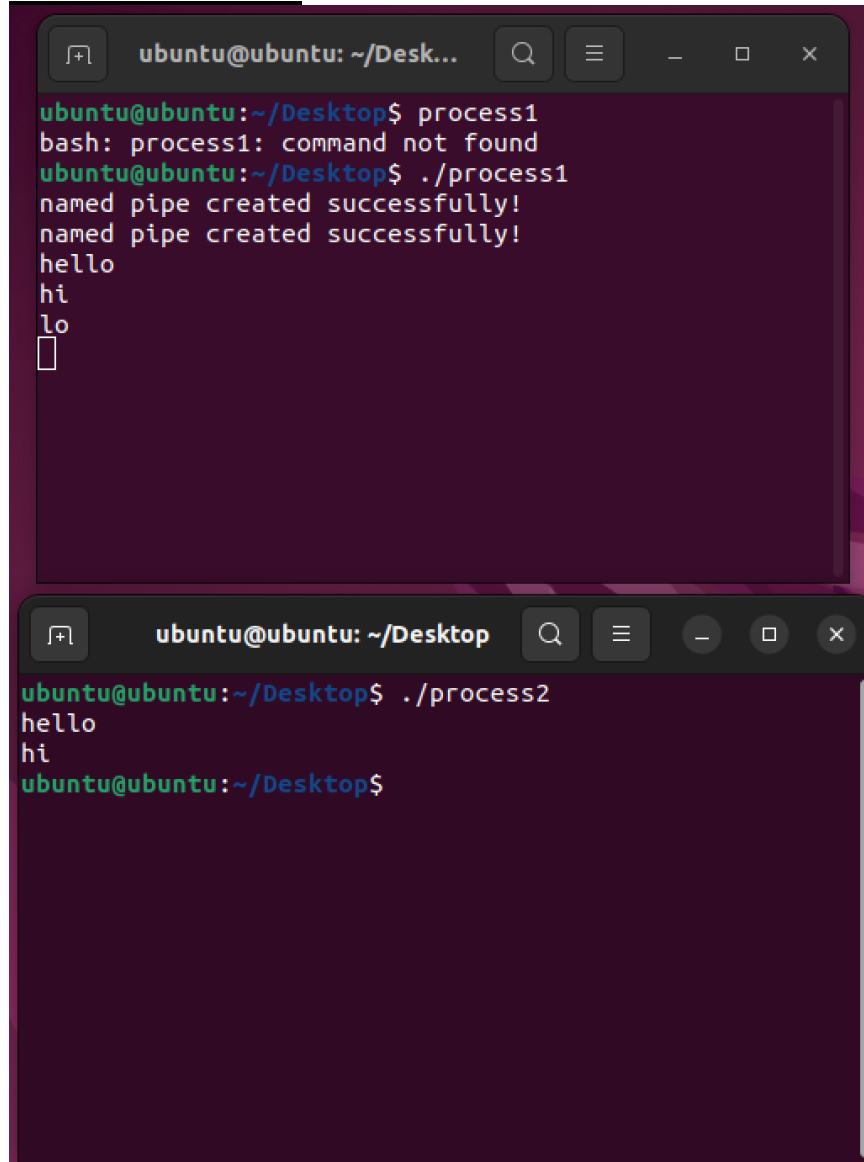
lab5_1r.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int main(){
    int res;
    char buffer[50];
//first
    res=mkfifo("fifo1",0777);
    if(res==0) printf("named pipe created successfully!\n");
//second
    res=mkfifo("fifo2",0777);
    if(res==0) printf("named pipe created successfully!\n");
//open
    int n,fd,fd2;
    fd=open("fifo1",O_RDONLY);
    fd2=open("fifo2",O_WRONLY);
    while(1){
    n=read(fd,buffer,50);
    if(n) printf("%s",buffer);
    if(strcmp(buffer,"exit")==0)break;
```

122CS0067

//
```
    fgets(buffer, 50, stdin);
    n=write(fd2,buffer,strlen(buffer));
    return 0;
}
}
}
```

Terminal Screenshot:

2. Write a program that demonstrates non-blocking I/O with FIFOs. One process writes to the FIFO, while another process reads from it in a non-blocking manner.

lab5_2r.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
int main(){
    int res;
    char buffer[50];
//first
    res=mkfifo("fifo1",0777);
    if(res==0) printf("named pipe created successfully!\n");
//open
    int n,fd;
    fd=open("fifo1",O_RDONLY);
    while(1){
    n=read(fd,buffer,50);
    if(n) printf("%s",buffer);
    }
return 0;
}
```

lab5_2w.c

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

int main(){
    int res;
    char buffer[50];
//first file
    res=mkfifo("fifo1",0777);
    if(res==0) printf("named pipe created successfully!\n");

//opening
    int n,fd;
    fd=open("fifo1",O_WRONLY);

    while(1){
    fgets(buffer, 50, stdin);
    n=write(fd,buffer,strlen(buffer));
    }
    return 0;
}
```

Terminal Screenshot:

3. Write a program where multiple child processes write data to a single FIFO, and a parent process reads the data from the FIFO.

lab5_3.c

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

int main(){
    int res;
    char buffer[50];
//first file
    res=mkfifo("fifo1",0777);
    if(res==0) printf("named pipe created successfully!\n");
int a=fork();
int b=fork();
//opening
if(a>0 && b>0){
    int n,fd;
    fd=open("fifo1",O_RDONLY|O_NONBLOCK);
    while(1){
    n=read(fd,buffer,50);
    if(n) printf("%s",buffer);
    }

}

else if(a==0 && b>0){
    int n,fd;
    fd=open("fifo1",O_WRONLY);
    n=write(fd,"FIRST\n",6);
```
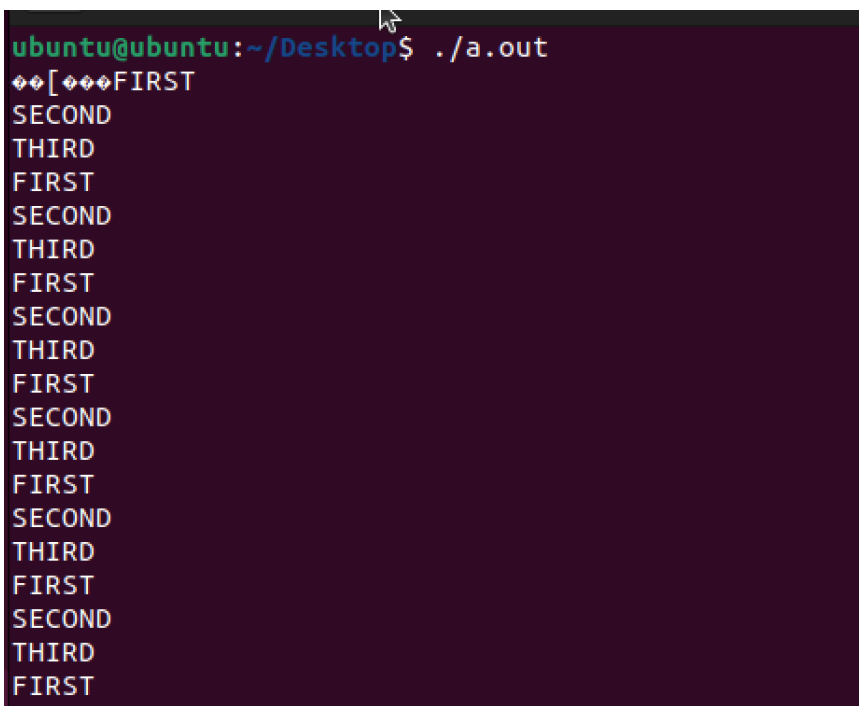
```
    }
else if(a>0 && b==0){
    int n,fd;
    fd=open("fifo1",O_WRONLY);
    n=write(fd,"SECOND\n",7);

    }
else if(a==0 && b==0){
    int n,fd;
    fd=open("fifo1",O_WRONLY);
    n=write(fd,"THIRD\n",8);
    }


    return 0;
}
```

Terminal Screenshot: