

Name: Aditya Somani

Roll No: T1851061

Div: A

PRN NO. 71901204L

ASSIGNMENT NO. 7A

TITLE: Inter-process Communication (IPC) in Linux using Pipes.

Program:

```
#include<stdio.h>
```

```
#include<string.h>
```

```
#include<sys/types.h>
```

```
#include<unistd.h>
```

```
#include<sys/wait.h>
```

```
int main()
```

```
{
```

```
    int pipe1[2],pipe2[2],i,j,data_pro,cpid,k=0;
```

```
    char buffer[100],path[500]="/home/anuj/Desktop/oslab/%a/test.txt";
```

```
    char pathbuf[500],ch;
```

```
    FILE * f1;
```

```
    if(pipe(pipe1)<0)
```

```
        printf("PIPE1 ERROR \n");
```

```
    else
```

```
        printf("PIPE1 CREATED \n");
```

```
    if(pipe(pipe2)<0)
```

```
        printf("PIPE2 ERROR \n");
```

```
    else
```

```
        printf("PIPE2 CREATED \n");
```

```
    pid_t ppid,ppid,apid;
```

```

cpid=fork();

if(cpid==0)
{
    sleep(5);
    printf("\n\t ===== Child Process =====");
    printf("\n\t cpid %d\n",cpid);
    printf("\n\t I AM CHILD PROCESS CPID %d\n",getpid());
    printf("\n\t MY PARENT ID IS CPPID %d\n",getppid());
    close(pipe1[1]);
    close(pipe2[0]);
    data_pro=read(pipe1[10],path,sizeof(path));
    printf("\n\t Child Wrote %d bytes\n",data_pro);
    f1=fopen(path,"r");
    if(f1==NULL)
        printf("\n\t FILE DOES NOT EXIST\n");
    else{
        printf("\n\t FILE FOUND BY CHILD PROCESS \n");

        while((ch=getc(f1))!=EOF)
        {
            buffer[k]=ch;
            k++;
        }
    }

    printf("\n\t ===== Child Process =====");
    data_pro=write(pipe2[1],buffer,strlen(buffer));
    printf("\n\t Child Wrote %d bytes \n",data_pro);
    close(pipe2[1]);
}

```


```

        fclose(f1);
    }
    else{
        printf("\n\t===== Parent Process =====");
        printf("\n\tcpid %d\n",cpid);
        printf("\n\tI AM PARENT PROCESS PID %d \n",getpid());
        printf("\n\tMY PARENT ID IS PPID %d \n",getppid());

        close(pipe1[0]);
        data_pro=write(pipe1[1],path,strlen(path));
        printf("\n\tParent Wrote %d bytes \n",data_pro);
        close(pipe1[1]);
        sleep(10);
        printf("\n\t===== Parent Process =====");
        printf("\n\tPARENT READS\n");
        data_pro=read(pipe2[0],pathbuf,sizeof(pathbuf));
        printf("\n\tParent Read %d bytes:\n %s\n",data_pro,pathbuf);
    }
    return 0;
}

```

Output:

**OnlineGDB** beta
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE

My Projects




Classroom new


Learn Programming

Programming Questions

Sign Up

Login

 44.4K



Scale up, retract, hold steady: Hybrid cloud gives you flexibility to do what times demand.

ADS VIA CARBON

[About](#) • [FAQ](#) • [Blog](#) • [Terms of Use](#) • [Contact Us](#) • [GDB Tutorial](#) • [Credits](#) • [Privacy](#)
© 2016 - 2020 GDB Online

input

```
main.c:32:10: warning: unknown escape sequence: '\040'
PIPE1 CREATED
PIPE2 CREATED

===== Parent Process =====
cpid 3374

I AM PARENT PROCESS PID 3373

MY PARENT ID IS PPID 3368

Parent Wrote 36 bytes

===== Child Process =====
cpid 0

I AM CHILD PROCESS CPID 3374

MY PARENT ID IS CPPID 3373

Child Wrote -1 bytes

FILE DOES NOT EXIST


===== Child Process =====
Child Wrote 6 bytes

===== Parent Process =====
PARENT READS

Parent Read 6 bytes:

0zR0l

...Program finished with exit code 0
Press ENTER to exit console.
```



ENG 14:30

