**ASSIGNMENT NO. 7 (a)//parent n child process??**

**INTER PROCESS COMMUNICATION USING PIPES**

#include<unistd.h>

#include<stdio.h>

#include<sys/types.h>

#include<sys/wait.h>

#include<string.h>

int main()

{

pid\_t pid;

int p1[2],p2[2], nbytes, i=0;

char str[50]="\0", fName1[20]="\0",fName[20]="\0", ch[50]="\0";

FILE \*fp;

pipe(p1);

pipe(p2);

pid=fork();

if(pid<0)

{

printf("\n\nError in executing fork\n\n");

}

else

{

if(pid==0)

{

sleep(10);

printf("\n\nControl in Child Process..");

close(p1[1]); //Closing the Writing End

nbytes=read(p1[0], fName1, sizeof(fName1));

close(p1[0]); //Closing the Reading End

printf("\n\nFilename Received: %s", fName1);

fp=fopen(fName1, "r");

ch[0]=fgetc(fp);

while(!feof(fp))

{

i++;

ch[i]=fgetc(fp);

}

//ch[i]='\0';

ch[strlen(ch)-1]= '\0';

fclose(fp); //because last character is end of file

printf("\n\nFetching Data From File..");

close(p2[0]); //Closing the Reading End

write(p2[1], ch, sizeof(ch));

close(p2[1]); //Closing the Writing End

printf("\n\nData Transferred..");

}

else

{

printf("\nControl in Parent Process..");

printf("\nEnter the File Name(with Extension): ");

scanf("%s", fName);

if(!(fopen(fName,"r")))

{

printf("\n\nFile Doesn't Exist!! Create the file and run again..");

return 0;

}

close(p1[0]); //Closing the Reading End

write(p1[1], fName, sizeof(fName));

close(p1[1]); //Closing the Writing End

sleep(10);

printf("\n\n\nControl back in Parent Process..");

close(p2[1]);

read(p2[0], str, sizeof(str));

close(p2[0]);

printf("\n\nFile Data:\n%s", str);

}

}

return 0;

}

**OUTPUT**

[it@localhost ~]$ gcc UnnamedPipe.c

[it@localhost ~]$ ./a.out

Control in Parent Process..

Enter the File Name(with Extension): test.txt

Control in Child Process..

Filename Received: test.txt

Fetching Data From File..

Data Transferred..

Control back in Parent Process..

File Data:

Hii..

Bye..

Hello..

Good Day

Night..

[it@localhost ~]$