# SP19-BCS-048 Shazil Zafar

# SP19-BCS-078 Saud ul Hassan

# Question No 1:

Construct two-way communication between Parent and Child processes in order to perform the following tasks: (15)

a) Parent writes content of current working directory at its writing end.

b) Child reads the list of contents at its reading end and sorts it into alphabetical order.

c) Child writes the sorted list of contents at its writing end.

d) Parents reads the sorted list at its reading end.

# Code:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<sys/wait.h>

#include<sys/types.h>

#include<dirent.h>

int main(void) {

char file[50][50];

char fileout[50][50];

char output[50][50];

pid\_t childpid;

int fd[2];

int i = 0;

if (pipe(fd) == -1) {

perror("Failed to create the pipe");

return 1;

}

childpid = fork();

if (childpid == -1) {

perror("Failed to fork");

return 1;

}

if (childpid){

DIR \*dir;

struct dirent \*rd;

dir = opendir(".");

if(dir == NULL)

{

printf("Error opening directory!");

exit(0);

}

while((rd = readdir(dir)) != NULL)

{

strcpy(file[i],rd->d\_name);

i++;

}

closedir(dir);

// printf("Parent Process Unsorted files \n");

// for(int j = i;j>=0;j--)

// printf("%s \n",file[j]);

write(fd[1], file,sizeof(file));

wait(NULL);

read(fd[0], output,sizeof(file));

printf("Parent Process sorted files \n");

for(int j = 0;j<=i;j++)

printf("%s \n",output[j]);

}

else {

read(fd[0], fileout,sizeof(file));

int len = 0;

while(strcmp(fileout[++len],"\0")!=0);

// printf("\nChild unsorted Files: \n");

// for(int k = 0;k<=len;k++)

// printf("%s \n",fileout[k]);

char temp[50];

for(int i=0; i<len; i++){

for(int j=0; j<len-1-i; j++){

if(strcmp(fileout[j], fileout[j+1]) > 0){

strcpy(temp, fileout[j]);

strcpy(fileout[j], fileout[j+1]);

strcpy(fileout[j+1], temp);

}

}

}

// printf("\nChild Sorted Files: \n");

// for(int k = 0;k<=len;k++)

// printf("%s \n",fileout[k]);

//writing in child

write(fd[1], fileout,sizeof(fileout));

}

return 0;

}

# Output:

Text

Description automatically generated

# Question No 2

Develop a C program that creates four child processes. Each of which does exactly one of the following (10)

a) Child I creates a text file using cat command.

b) Child 2 list down the content of current working directory.

c) Child 3 creates a new directory using mkdir command

d) Child 4 removes the directory that child 3 has created.

# Code:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<sys/wait.h>

#include<sys/types.h>

#include<dirent.h>

int j = 0,i=0;

int main()

{

int childid;

for(i=0;i<4;i++)

{

childid = fork();

if( childid == 0){

if(j==0){

//Child 1

execl("/bin/sh", "sh", "-c", "/bin/cat saud.txt", NULL);

}else if(j==1){

//Child 2

DIR \*dir;

struct dirent \*rd;

dir = opendir(".");

if(dir == NULL)

{

printf("Error opening directory!");

exit(0);

}

while((rd = readdir(dir)) != NULL)

{

printf("Dir Name: %s\n",rd->d\_name);

}

closedir(dir);

}else if(j==2){

//child 3

char \*args[] = {"mkdir","hello.txt",NULL};

execvp("/bin/mkdir",args);

}else if(j==3){

//child 4

char \*args[] = {"rmdir","hello.txt", NULL};

execvp("/bin/rmdir",args);

}

exit(0);

}

j++;

wait(NULL);

}

return 0;

}

# Output:

Text

Description automatically generated with medium confidence