Name : Muhammad Ishraf Shafiq Zainuddin

ID : 200342741

Lab Assgn : 4

Part 1 (1)

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

int main(int argc, char *argv[])
{

 //Reference : https://stackoverflow.com/questions/3411018/system-call-to-copy-files

 FILE *from, *to;
 char ch;

 if(argc!=3)
 {
  printf("Error!! Not enough arguments~ :D \n"); //Print error if not enough arguments
  exit(1);
 }

 //filename1
 if((from = fopen(argv[1], "rb"))==NULL) {
  printf("Error~ Can't open filename1 \n");   //Print error if unable to find filename1
  exit(1);
 }

 //filename2 = opening new file to copy and paste
 if((to = fopen(argv[2], "wb"))==NULL) {
  printf("Error!! Not enough arguments \n"); //print error if there's no enough arguments
  exit(1);
 }

 //Copy function
 while(!feof(from))
 {
  ch = fgetc(from);
  if(ferror(from))
  {
   printf("Error reading filename1.\n");
   exit(1);
  }
```

```
  if(!feof(from)) fputc(ch, to);
  if(ferror(to))
  {
    printf("Error writing filename2.\n");
    exit(1);
  }
}
```

//Reference : (1) http://c.happycodings.com/c-on-unix/code14.html, (2) CS330 Lab Page

```
pid_t pid, ppid;
pid_t fork_return;
gid_t gid;

fork_return = fork();

//Getting Process ID
if (fork_return == 0)
{
  printf("The Process ID is %d", pid); //print process ID
  printf("\n\n");
}
else if (fork_return == -1)
{
   perror("Error~");  //print error if unable to get process ID
}


//Getting Parent ID
if (fork_return > 0)
{
  printf("The Parent ID is %d", ppid); //print parent ID
  printf("\n\n");
}
else if (fork_return == -1)
{
  perror("Error~"); //print error if unable to get parent ID
}

//Getting Process Group ID
if (fork_return == 0)
{
  printf("The group id is %d", gid); // print process group ID
  printf("\n\n");
}
else if (fork_return == -1)
{
```

```
    perror("Error~");  //print error if unable to get process group ID
   }

  return 0;
}
```

Part 1 (2)

(a) It will not print.

(b) Getting parent ID is the parent of the executable program

(c) By using wait() function.

(d) -p = preserving specified attributes such as mode,ownership and timestamp. -i = prompting the user before overwrite something.

Part 2

//Reference : https://stackoverflow.com/questions/18905905/c-programing-parent-child-process

Because what happened int the parent process is different and does not affect the child's process due to memory protection. The parent and the child's process have separate virtual address space.

Script

Script started on 2018-05-30 02:08:17-0600

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ ls -l

total 32

drwxr-xr-x 2 shafiqzain shafiqzain 4096 May 30 01:31 #[0m#[01;34mAssignment1#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1347 May 29 19:23 #[01;32mfilename1#[0m

-rw-r--r-- 1 shafiqzain shafiqzain   0 May 30 02:08 lab4.log

drwxr-xr-x 4 shafiqzain shafiqzain 4096 May 18 00:11 #[01;34mLabs#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 12776 May 30 01:53 #[01;32mmyCp#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1951 May 30 01:50 #[01;32mmyCp.cpp#[0m

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ cat filename1

The following is taken from page 1 of Interprocess Communications in Unix:

Fundamental to all operating systems is the concept of a process. While somewhat abstract, a process consists of an executing (running) program, its current values, state information, and the resources used by the operating system to manage the execution of the process. A process is a dynamic entity. In a UNIX-based operating system, at any given point in time, multiple processes appear to be executing concurrently. From the viewpoint of each of the processes involved it appears they have access to, and control of, all system resources as if they were in their own stand-alone setting. Both viewpoints are an illusion. The majority of UNIX operating systems run on platforms that have a single processing unit capable of supporting many active processes. However, at any point in time only one process is actually being worked upon. By rapidly changing the process it is currently executing, the UNIX operating system gives the appearance of concurrent process execution. The ability of the operating system to multiplex its resources among multiple processes in various stages of execution is called multiprogramming (or multitasking). Systems with multiple processing units, which by definition can support true concurrent processing are called multiprocessing.

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$.?##[K/myCp

Error!! Not enough arguments~ :D

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ ./myCp filename2##[K1

Error!! Not enough arguments~ :D

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ ./myCp filename1 filename2

The Parent ID is -1238681664

The Process ID is 21892

The group id is 21892

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ ls -l

total 36

drwxr-xr-x 2 shafiqzain shafiqzain 4096 May 30 01:31 #[0m#[01;34mAssignment1#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1347 May 29 19:23 #[01;32mfilename1#[0m

-rw-r--r-- 1 shafiqzain shafiqzain 2694 May 30 02:09 filename2

-rw-r--r-- 1 shafiqzain shafiqzain   0 May 30 02:08 lab4.log

drwxr-xr-x 4 shafiqzain shafiqzain 4096 May 18 00:11 #[01;34mLabs#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 12776 May 30 01:53 #[01;32mmyCp#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1951 May 30 01:50 #[01;32mmyCp.cpp#[0m

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ cat filename2

The following is taken from page 1 of Interprocess Communications in Unix:

Fundamental to all operating systems is the concept of a process. While somewhat abstract, a process consists of an executing (running) program, its current values, state information, and the resources used by the operating system to manage the execution of the process. A process is a dynamic entity. In a UNIX-based operating system, at any given point in time, multiple processes appear to be executing concurrently. From the viewpoint of each of the processes involved it appears they have access to, and control of, all system resources as if they were in their own stand-alone setting. Both viewpoints are an illusion. The majority of UNIX operating systems run on platforms that have a single processing unit capable of supporting many active processes. However, at any point in time only one process is actually being worked upon. By rapidly changing the process it is currently executing, the UNIX operating system gives the appearance of concurrent process execution. The ability of the operating system to multiplex its resources among multiple processes in various stages of execution is called multiprogramming (or multitasking). Systems with multiple processing units, which by definition can support true concurrent processing are called multiprocessing.

The following is taken from page 1 of Interprocess Communications in Unix:

Fundamental to all operating systems is the concept of a process. While somewhat abstract, a process consists of an executing (running) program, its current values, state information, and the resources used by the operating system to manage the execution of the process. A process is a dynamic entity. In a UNIX-based operating system, at any given point in time, multiple processes appear to be executing concurrently. From the viewpoint of each of the processes involved it appears they have access to, and control of, all system resources as if they were in their own stand-alone setting. Both viewpoints are an illusion. The majority of UNIX operating systems run on platforms that have a single processing unit capable of supporting many active processes. However, at any point in time only one process is actually being worked upon. By rapidly changing the process it is currently executing, the UNIX operating system gives the appearance of concurrent process execution. The ability of the operating system to multiplex its resources among multiple processes in various stages of execution is called multiprogramming (or multitasking). Systems with multiple processing units, which by definition can support true concurrent processing are called multiprocessing.

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$.,##[K?##[K/myCp myCp /net/share/aliases/CS330-001.20

01820 filename2#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ cat filename2#[K

#[K#[A#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C######
#########./myCp myCp /net/share/aliases/CS330-001.20

01820 filename2##############

#[A#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#[C#####################################[CmyCp /net/share/aliases/CS330-001.201#[1P#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ #[C#[C#[C#[C#[C#[CyCp /net/share/aliases/CS330-001.2018#[1P#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ #[C#[C#[C#[C#[C#[CCp /net/share/aliases/CS330-001.20182#[1P#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ #[C#[C#[C#[C#[C#[Cp /net/share/aliases/CS330-001.201820#[1P filename2#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ #[C#[C#[C#[C#[C#[C /net/share/aliases/CS330-001.201820#[1Pfilename2#[A

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ #[C#[C#[C#[C#[C#[C

Error~ Can't open filename1

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ ls -l

total 40

drwxr-xr-x 2 shafiqzain shafiqzain 4096 May 30 01:31 #[0m#[01;34mAssignment1#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1347 May 29 19:23 #[01;32mfilename1#[0m

-rw-r--r-- 1 shafiqzain shafiqzain 2694 May 30 02:09 filename2

-rw-r--r-- 1 shafiqzain shafiqzain 4096 May 30 02:10 lab4.log

drwxr-xr-x 4 shafiqzain shafiqzain 4096 May 18 00:11 #[01;34mLabs#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 12776 May 30 01:53 #[01;32mmyCp#[0m

-rwxr-xr-x 1 shafiqzain shafiqzain 1951 May 30 01:50 #[01;32mmyCp.cpp#[0m

#]0;shafiqzain@shafiqzain-X450CC: ~/CS330##[01;32mshafiqzain@shafiqzain-X450CC#[00m:#[01;34m~/CS330#[00m$ exit

exit

Script done on 2018-05-30 02:11:13-0600