`popen` with Signal Handling

Homework Assignment #10 (Last)

ELEC462-002 System Programming

(Instructor: Prof. Suh, Young-Kyoon)

Due: 01:59:59 pm, Thursday, May 25, 2023

Write down your own `popen` code (called `popen2`) as follows:

`popen2()` must support two modes (**read** and **write**) and ignore two signals (**CTRL-c** and **CTRL-**\) and `kill` command. Below is an output for execution.

[Expected output]

```
yksuh@localhost:~/courses/ELEC462/homeworks/hw10$ gcc -o hw10 hw10.c
yksuh@localhost:~/courses/ELEC462/homeworks/hw10$ ./hw10
PID: 4253
^C      SIGINT received but ignored...
^\      SIGQUIT received but ignored...
test
sample
hw10.c
hw10
helloworld
elec462_scratch.txt
        SIGKILL received but ignored...
yksuh@localhost:~/courses/ELEC462/homeworks/hw10$
```

```
yksuh@localhost:~/courses/ELEC462/homeworks/hw10$ kill 4253
yksuh@localhost:~/courses/ELEC462/homeworks/hw10$
```

Complete the code in `popen2()`. Following is a skeleton code of `popen2()`.

[Outline of `popen2()`]

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

FILE *popen2(const char*, const char*);

#define BUFF_SIZE 100

int main(){
    FILE *pipein_fp, *pipeout_fp;
    char readbuf[BUFF_SIZE];

    if( (pipein_fp = popen2("ls","r")) == NULL){
        perror("popen2");
        exit(1);
    }

    if( (pipeout_fp = popen2("sort -r", "w")) == NULL){
        perror("popen2");
        exit(1);
    }

    while( fgets(readbuf, BUFF_SIZE, pipein_fp))
        fputs(readbuf, pipeout_fp);

    pclose(pipein_fp);
    pclose(pipeout_fp);
    sleep(1);

    return 0;
}

FILE *popen2(const char *command, const char *mode){
    //  declare an array of pipes and will be using it
    //  if mode == 'r' then
    //      parent's end is READ and child_end is WRITE
    //  if mode == 'w' then
    //      parent's end is WRITE and child_end is READ
    //  Now get a pipe via pipe()

    //  Invoke fork()
    //  If (parent) then
    //      it needs to close one end and fdopen the other end
    //  If (child)  then
    //      it needs to close the other one end and redirect
    //  via dup2() stdin or stdout to another_end then exec cmd
    //  Now run cmd via execl() with "sh -c"

    //  otherwise, exit(1)
}
```

<Precautions!!>

- Name your source code like: `hw10_s<StudentID>.c`.
- Zip the code and name it like `hw10_s<StudentID>.zip`.
  Ex) If your student ID is 2022123456, your file name would be `hw10_s2022123456.zip`.
- Upload your zip file into LMS assignments tab.

Q & A

If you have a question, then contact TA by email (rinyo0126@knu.ac.kr). Or, leave your messages on LMS Q&A board.

Late Day Policy

All exercises are due at 1:59:59 pm on the assigned due date. A grading penalty will be applied to late assignments. Any assignment turned in late will be penalized 50% per late day.

Plagiarism

No plagiarism will be tolerated. Also, NEVER use ChatGPT. If the assignment is to be worked on your own, please respect it. If the instructor determines that there are substantial similarities exceeding the likelihood of such an event, he will call the two (or more) students to explain them and possibly to take an immediate test (or assignment, at the discretion of the instructor) to determine the student's abilities related to the offending work.