OS HW1

Operation system 107 fall

W.J. TSAI 蔡文錦 教授

TA 蘇聖雅 莊侑穎 劉晏 盧彥廷 黃資捷

PREWORK

Login Tools

- PuTTY

Editors

- vim

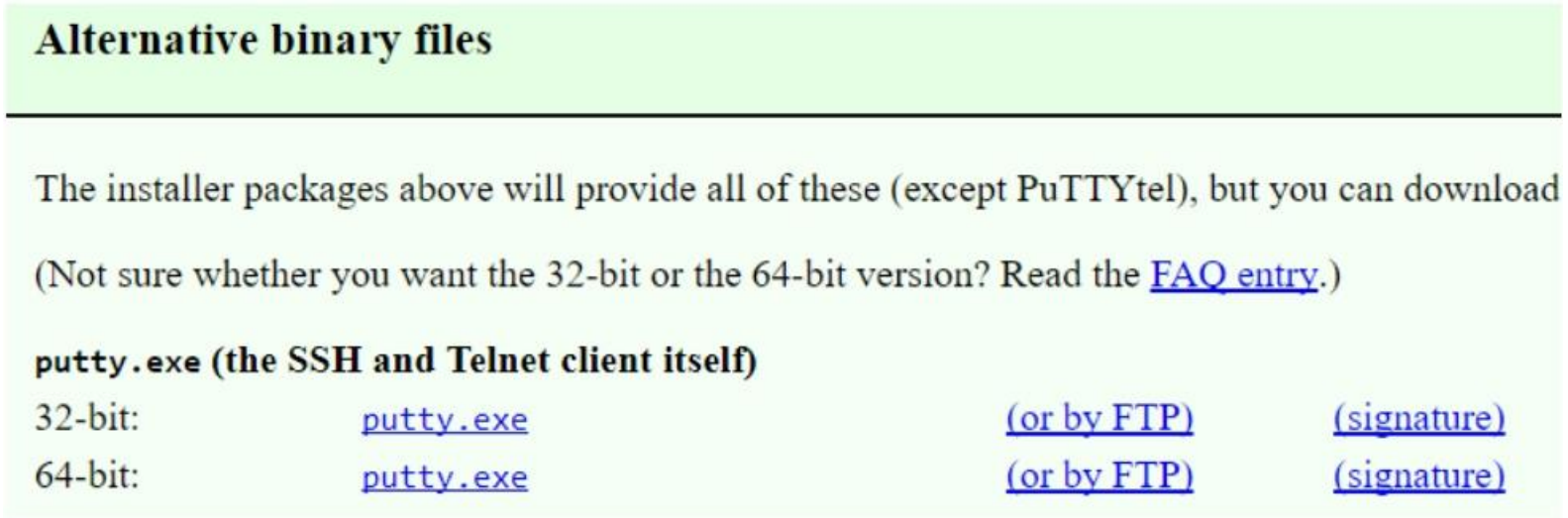
FTP Tools

- FileZilla Client

PuTTY

Download PuTTY

https://goo.gl/rM4Scb



Alternative binary files

The installer packages above will provide all of these (except PuTTYtel), but you can download

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

putty.exe (the SSH and Telnet client itself)

32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(signature)

PuTTY

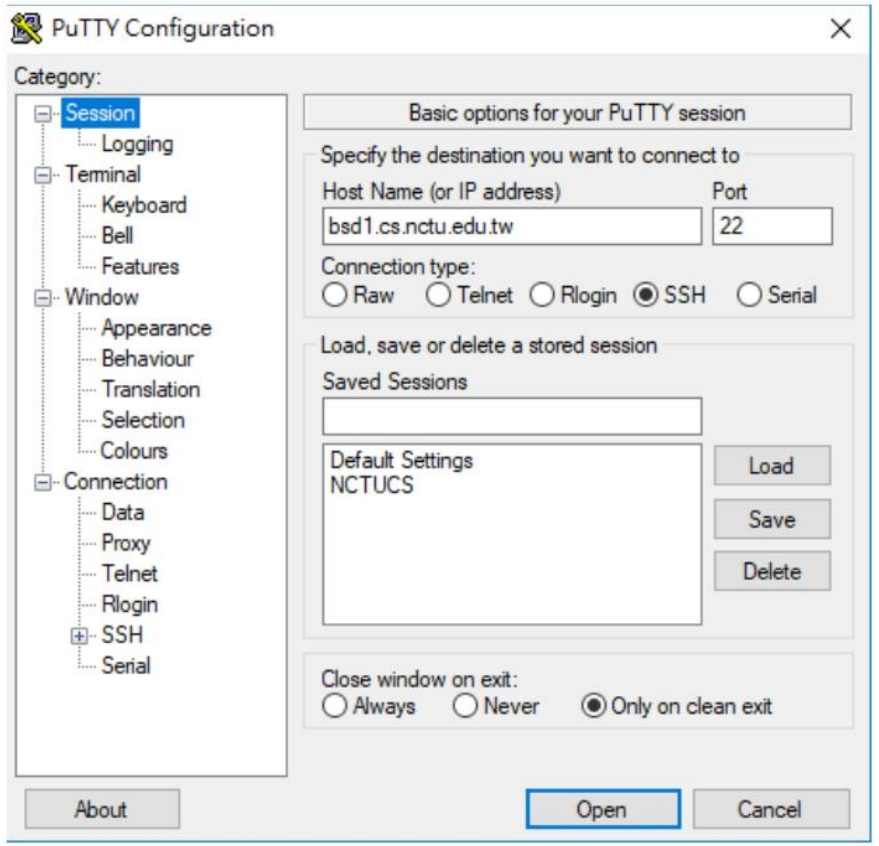
How to Use PuTTY

https://goo.gl/8AJsPL

Login

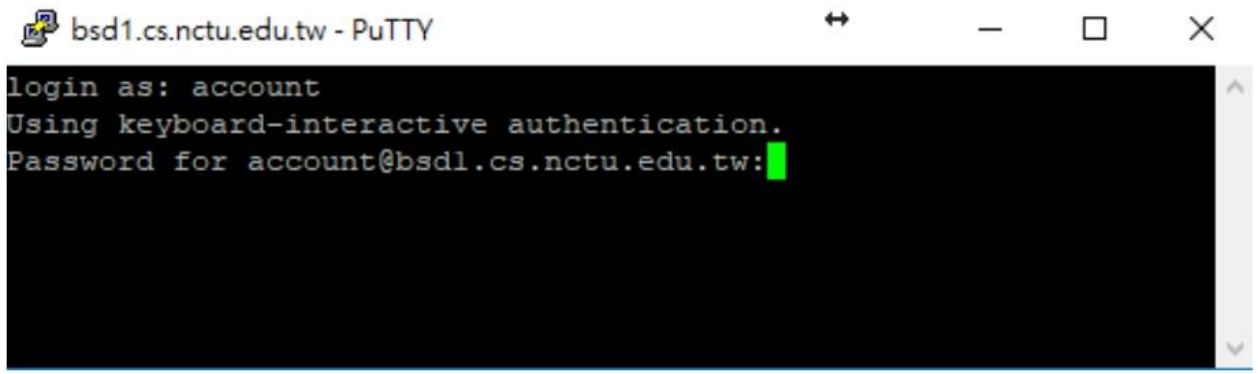
The default for SSH service is port 22

- bsd1.cs.nctu.edu.tw – bsd5.cs.nctu.edu.tw
- linux1.cs.nctu.edu.tw – linux6.cs.nctu.edu.tw



PuTTY

Command



- clear – clear the screen
- ls – list directory contents
- mv – move files or directories
- mkdir – create directories
- rm – remove files or directories
- chmod – change file system modes of files or directories

- Other instruction Reference
  - http://linux.vbird.org/linux_basic/redhat6.1/linux_06command.php#filesystem

FileZilla

- Upload File to Workstation
- Login

主機: bsd1.cs.nctu.edu.tw

協定: SFTP

登入型態: 一般

使用者: 計中申請帳號

密碼: 計中申請密碼

HW 1-1

Finish “hw1_1.c” in order to design a C program to serve as a shell interface that accepts user commands then execute each command in a separate process.

UNIX shells typically allow the child process to run in the background or concurrently, so if a ampersand(&) at the end of the command means the parent and child processes will run concurrently.

You will use:

read(STDIN_FILENO, inputBuffer, MAX_LINE): read command line

fork(): create child process

execvp(char *command, char *params[]): execute system calls

wait()

waitpid(pid)

…

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

#defile MAX_LINE 80

int main(void)
{
    char *arg[MAX_LINE/2+1]; /*command line arguments*/
    int should_run = 1; /*flag to determine when to exit program*/

    while(should_run){
        print("osh>");
        fflush(stdout);

        /**
        * your code!
        * After reading user input, the step are:
        * (1) fork a child process using fork()
        * (2) the child process will invoke execvp()
        * (3) if command included &, parent will invoke wait()
        */
    }

    return 0;
}
```

HW 1-1

- Change directory

  $cd your/folder/

- Compile

  $gcc -o shell hw1_1.c

- Execute

  $./shell

- **You need**

1. finish “hw1_1.c” as a shell interface.

2. user can keep entering the command until he/she enters “exit”.(a command include the command itself and its parameters).

3. if a user enter “&”, the shell should let child run in the background (means child and parent run concurrently).

Example

Show the content of hi.txt file

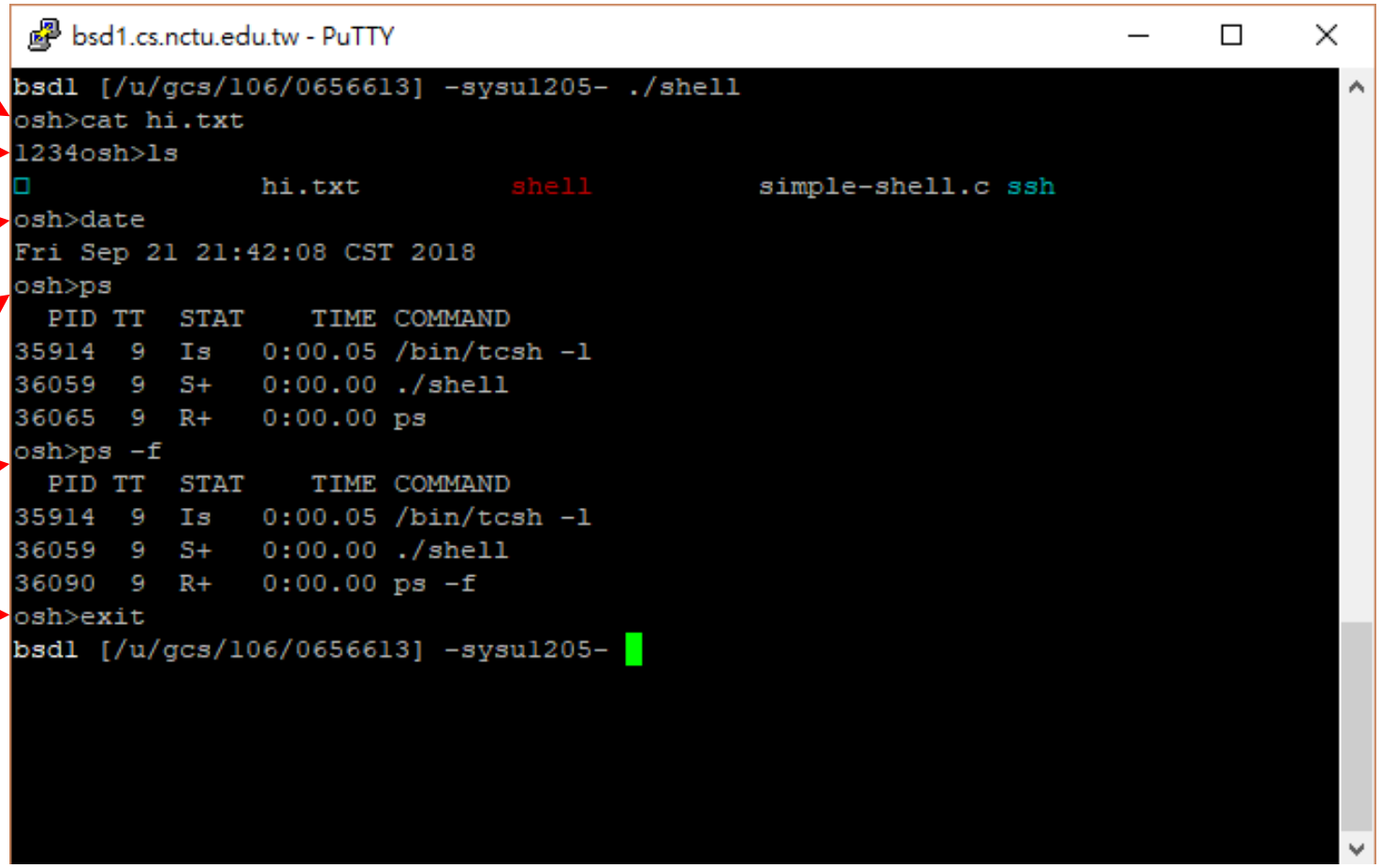
Show the file in the directory

Show the date

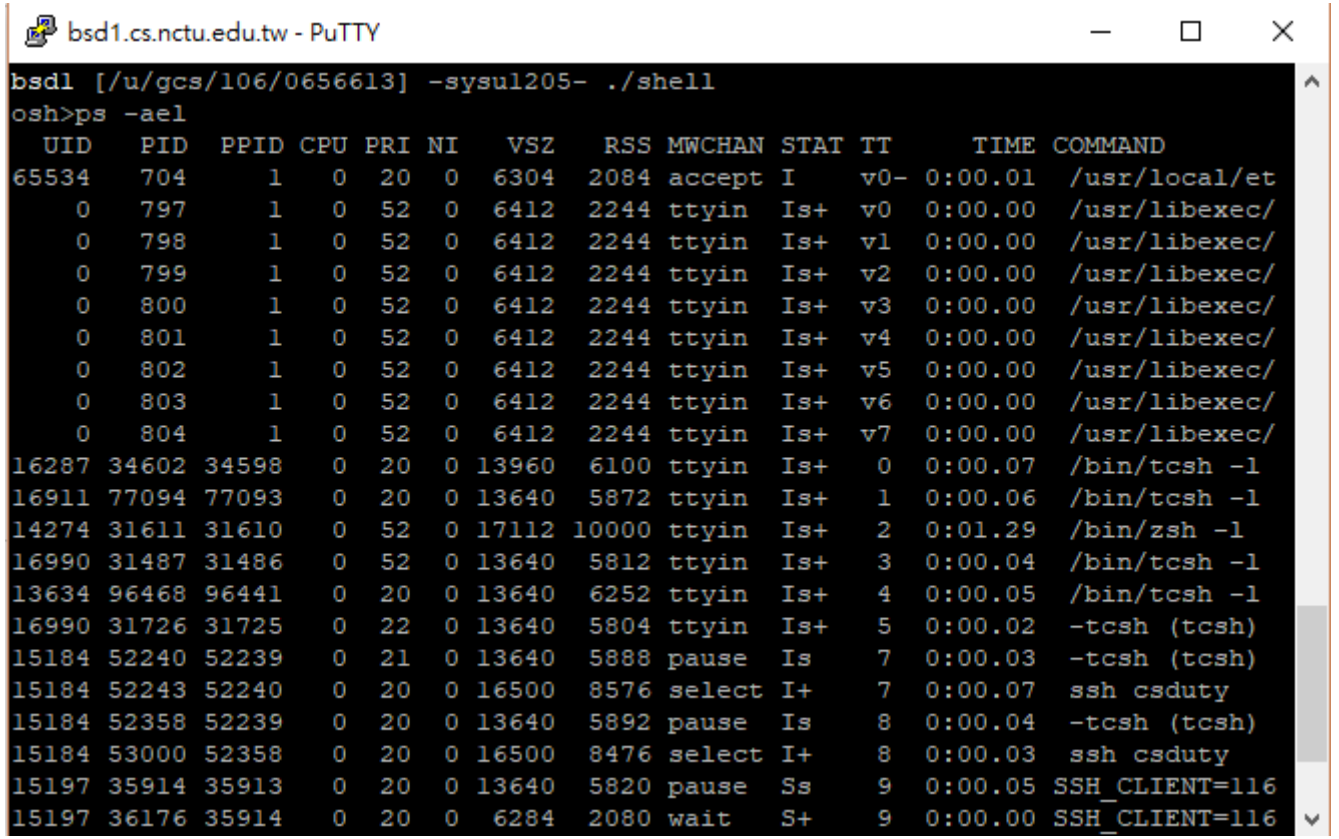
List all processes in current shell

Perform a full-format processes list

Enter “exit” to finish shell



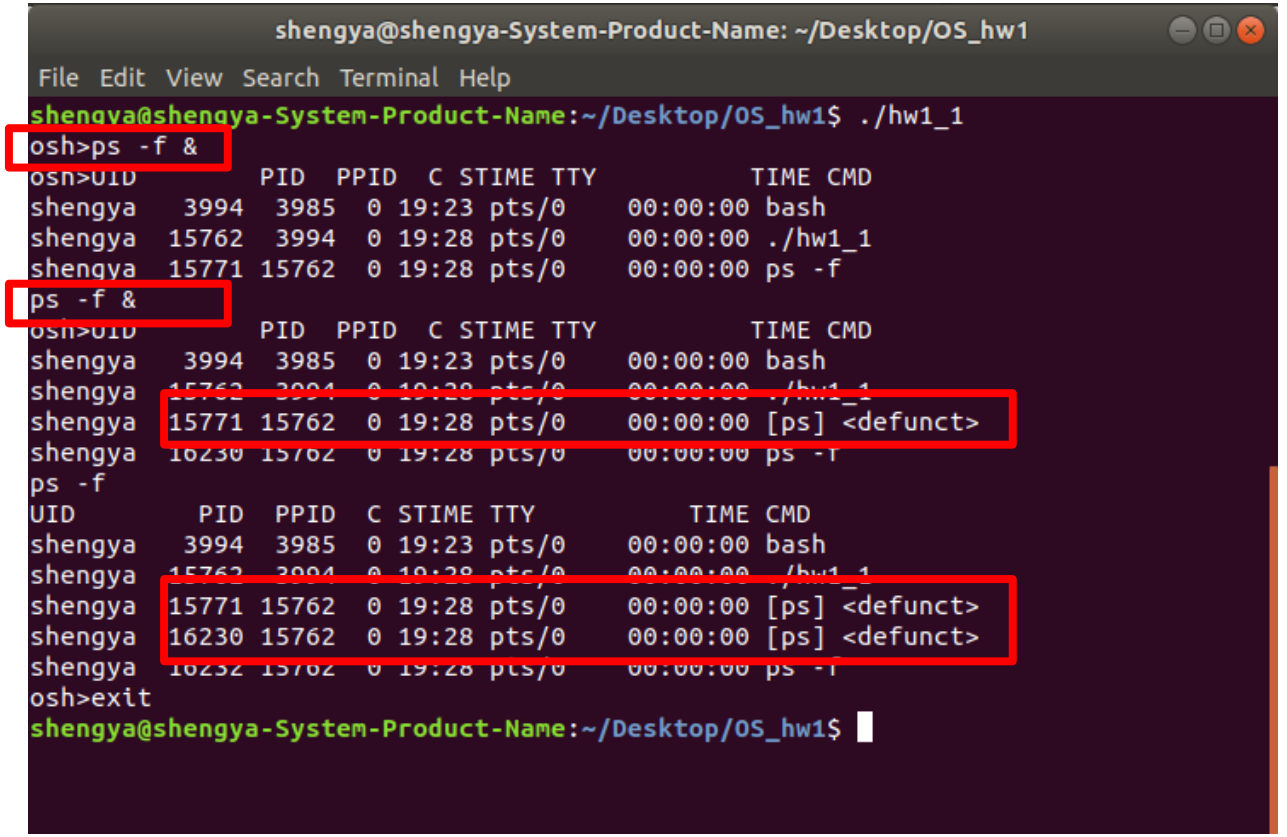
Example



Receive “-ael” as args and execute

Example

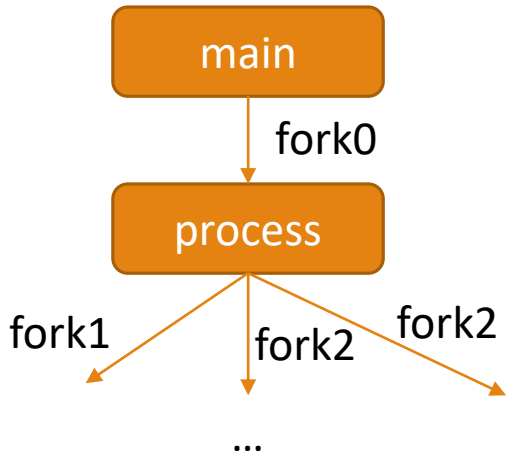
PID 15771&16230 becomes a zombie
(because ps –f **&** will let child process and parent process run concurrently, meaning that the parent process didn't call "wait" for the child)



HW 1-2

Please draw the tree format according the code on the report(OS_document.docx).

You need to clarify which fork(fork0, fork1, fork2 or fork3) the process been made by, for instance:

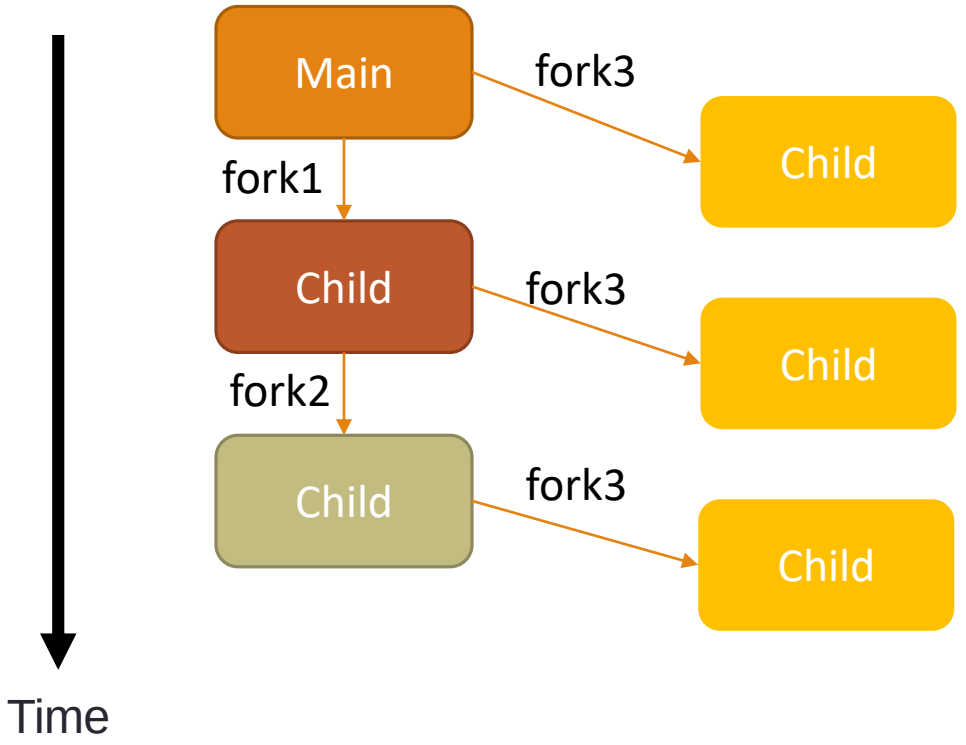



```c
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    pid_t pid;
    pid = fork();//fork0
    for(int i=0;i<2;i++)
    {
        if (pid==0){
            pid = fork();//fork1
            pid = fork();//fork2
        }else if(pid>0){
            pid = fork();//fork3
        }else{
            printf("Error!");
        }
    }
    return 0;
}
```

HW 1-3

Write a program which uses fork() to produce the following tree format (namely, your code should have only 3 fork())

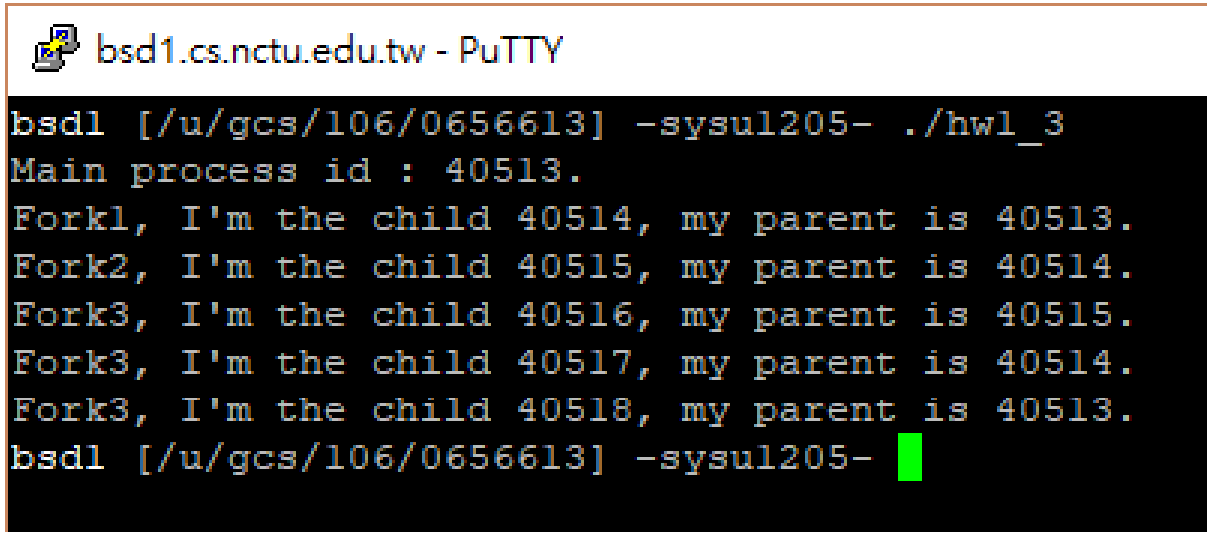


HW 1-3

Output format

Main process

Total 5 Children
The format & fork order
have to be same.
(Fork 1 2 3 3 3)

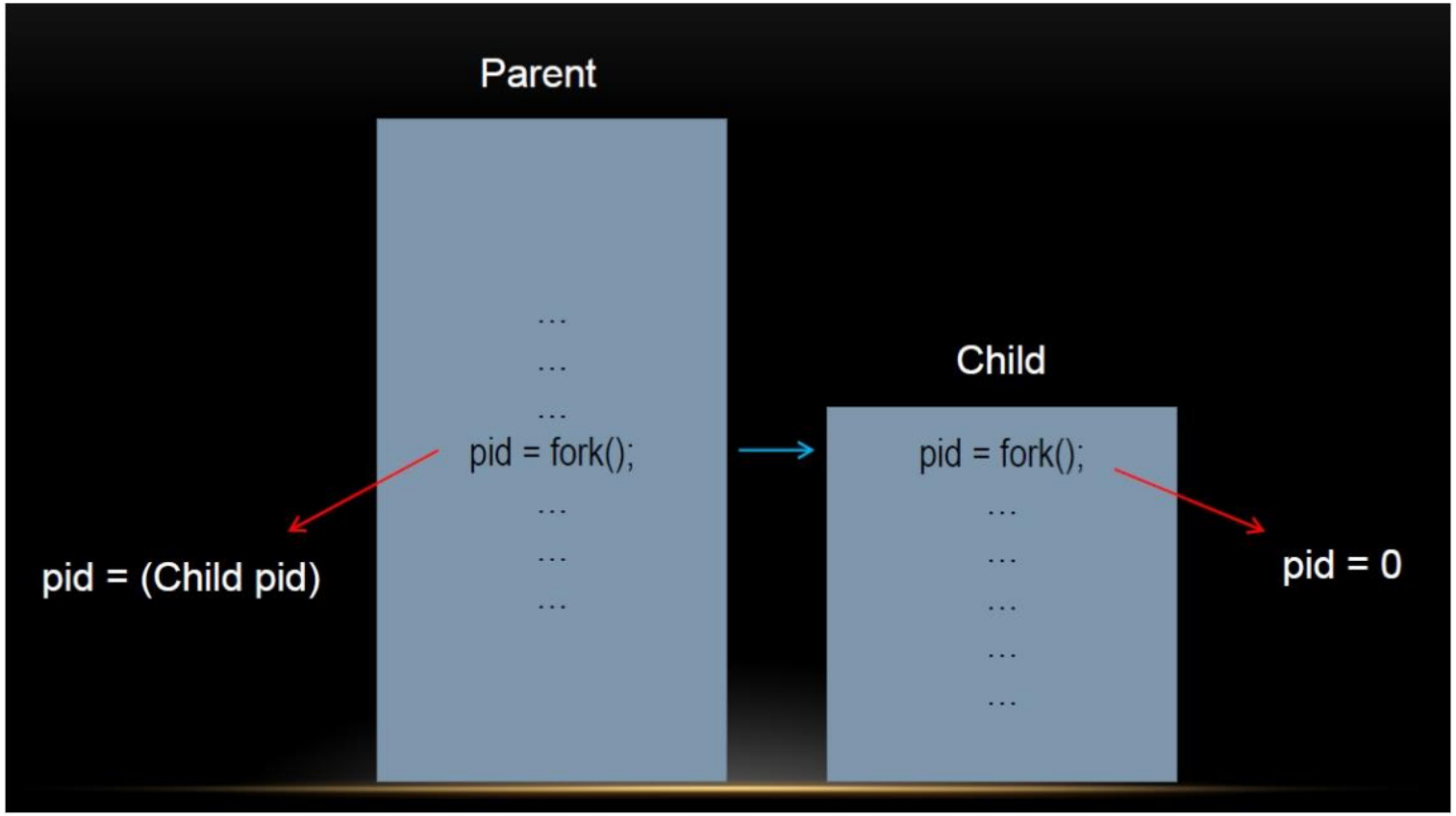


Hint:

Parent Process has to wait until Child Process finishes, then exit.

Use PID to identify parent and child.

Hint



Submission and Grade

Filename format please according : hw1-1.c, hw1-3.c (or .cpp), OS_report.docx.
Put two *.c(*.cpp) files and a *.docx report into same compressed file named **StudentID_hw1.zip** (ex : 0000000_hw1.zip).

Deadline: 2018/10/14 (SUN) PM11:59

a. Total score: 100pts. **COPY WILL GET A 0 POINT!**

b. hw1-1 score: code 40pts, report Q1 10pts

c. hw1-2 score: report Q2 20pts

d. hw1-3 score: code 20pts, report Q3 10pts

e. Report: format is in OS_report.docx. YOU NEED TO FINISH EVERY PART OF REPORT TO GET SCORE!

Rules

0. Use NCTU CS Workstation as your programming environment
1. Use only C/C++, OTHER LANGUAGES WILL GET 0 POINT!
2. **Incorrect filename format** will get -5 pts
3. **Incorrect output format** will get -5 pts
4. DELAYED SUBMISSION WILL GET 0 POINT!

*If you have any question, just send email to TAs.