OS HW1
Operation system 110 fall

W.J. TSAI 蔡文錦 教授
TA 姚淨云 張皓雲 林孟學 王彥珽

PREWORK

- Login Tools
  - PuTTY
- FTP Tools
  - FileZilla Client

PuTTY

Download PuTTY

Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel), but you can download standalone binaries one by one if you prefer.

(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)

64-bit x86:	`putty.exe`	(or by FTP)	(signature)
64-bit Arm:	`putty.exe`	(or by FTP)	(signature)

PuTTY

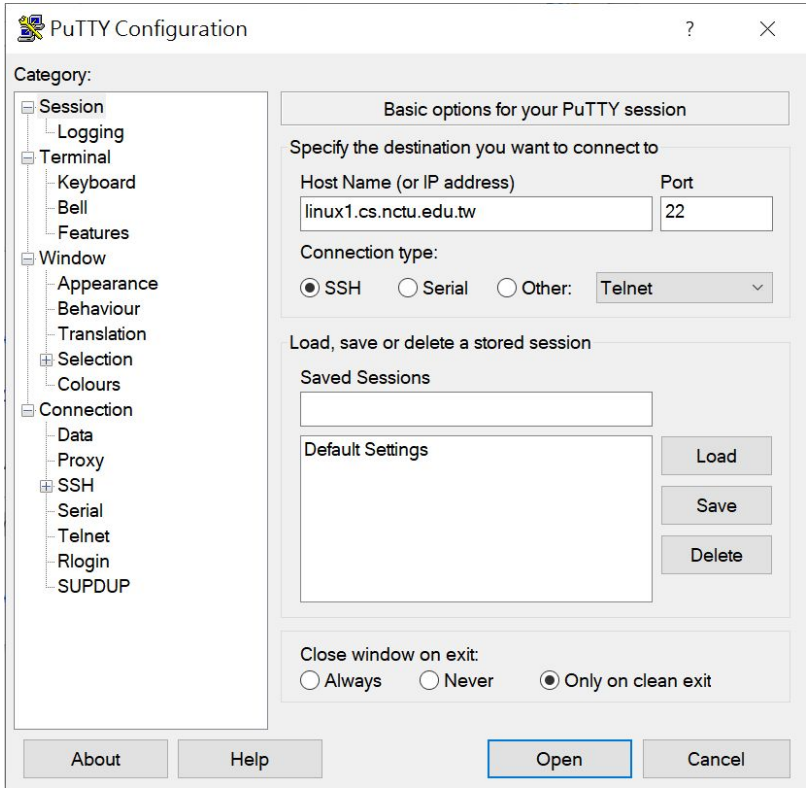
How to Use PuTTY

[國立陽明交通大學資工系資訊中心]

Login

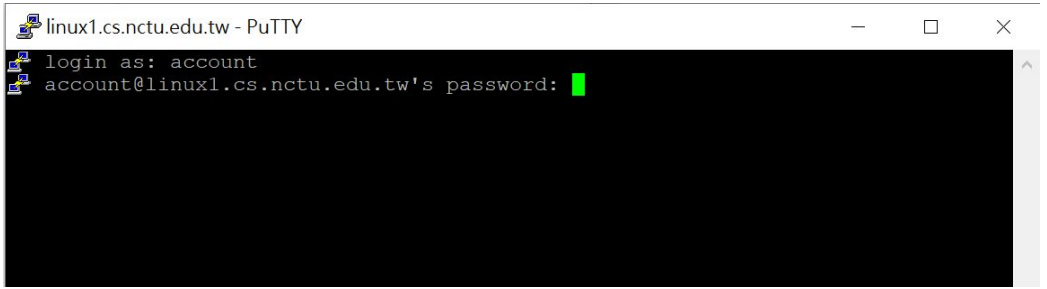
The default for SSH service is port 22

- bsd1.cs.nctu.edu.tw－bsd4.cs.nctu.edu.tw
- linux1.cs.nctu.edu.tw－linux4.cs.nctu.edu.tw



PuTTY

Command

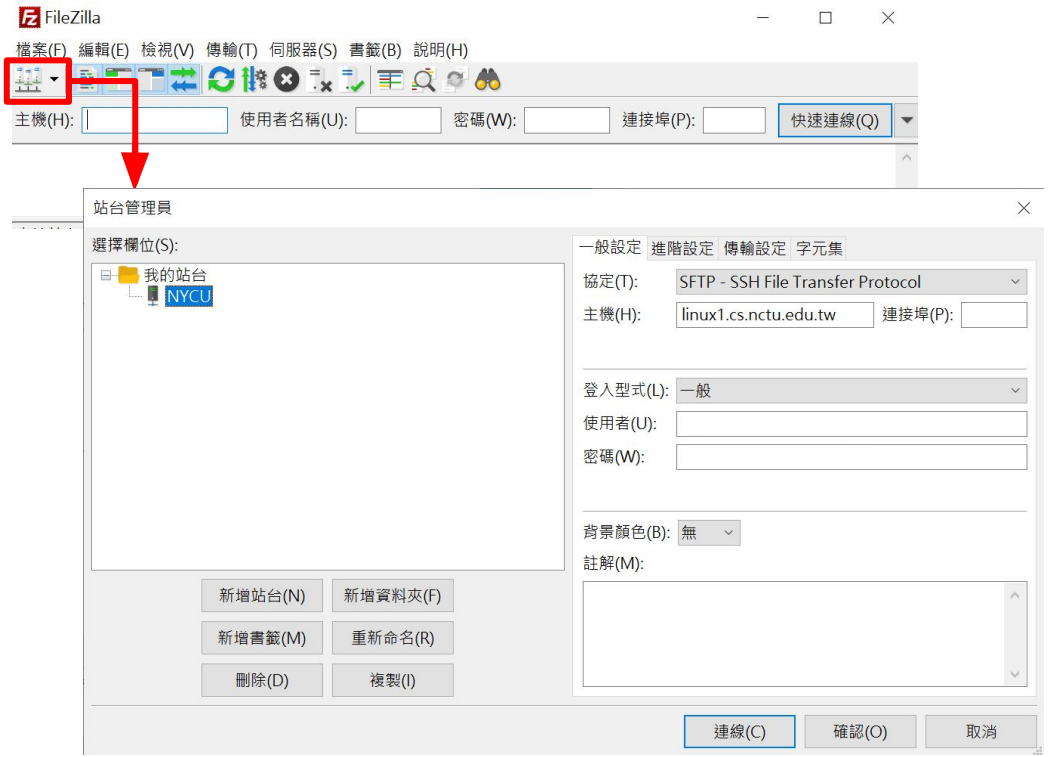


- clear－clear the screen
- ls－list directory contents
- mv－move files or directories
- mkdir－create directories
- rm－remove files or directories
- chmod－change file system modes of files or directories

- Other instruction Reference
  - 鳥哥的Linux私房菜

FileZilla

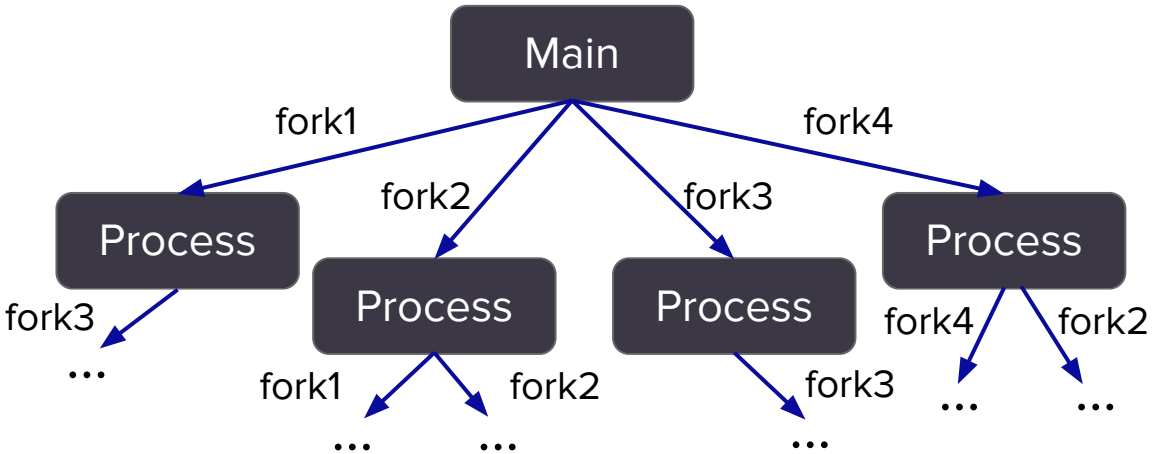
- Upload File to Workstation
- Login
  - 主機:linux1.cs.nctu.edu.tw
  - 協定:SFTP
  - 登入型態:一般
  - 使用者:計中申請帳號
  - 密碼:計中申請密碼



HW 1-1

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

You need to clarify which fork(fork0, fork1, fork2, fork3 and fork4) the process been made by,
For example:




```c
#include <stdio.h>
#include <unistd.h>
#include<sys/wait.h>

int main(){
    pid_t pid;
    pid = fork(); //fork0
    if(pid == 0){
        pid = fork(); //fork1
        if(pid > 0)
            wait(NULL);
        pid = fork(); //fork2
        if(pid > 0)
            wait(NULL);
    }
    else if(pid > 0){
        wait(NULL);
        pid = fork(); //fork3
        if(pid > 0)
            wait(NULL);
    }
    else{
        printf("Error!");
    }
    pid = fork(); //fork4
    if(pid > 0)
        wait(NULL);
    return 0;
}
```

Hint

System Call:

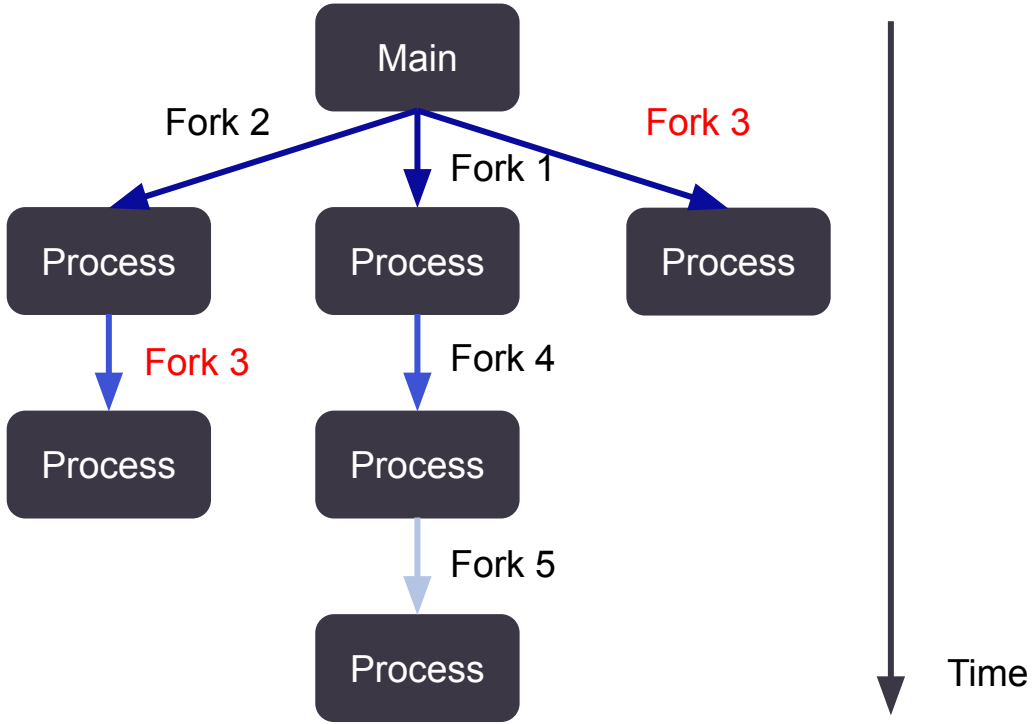
- getpid(): Get the process ID of the current process.
- getppid(): Get process ID of parent process.

Additions:

- You can use above system calls to help you complete this part.
- Draw which fork (fork0, fork1, fork2, fork3, fork4) the process been made by after the code is executed.

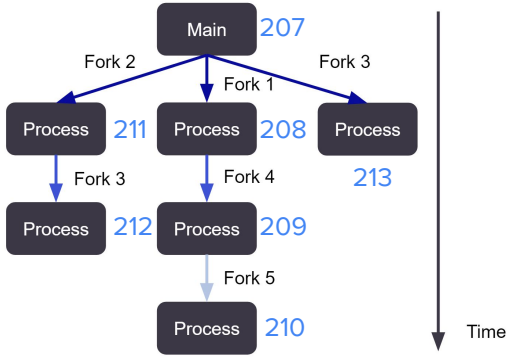
HW 1-2

Write a program which uses fork() to produce the following tree format.



HW 1-2

Your program must output messages in the format as below.



Main Process ID →

Total 6 Children

The format have to be same.

```
Main Process ID : 2575207

Fork 1. I'm the child 2575208, my parent is 2575207.
Fork 4. I'm the child 2575209, my parent is 2575208.
Fork 5. I'm the child 2575210, my parent is 2575209.
Fork 2. I'm the child 2575211, my parent is 2575207.
Fork 3. I'm the child 2575212, my parent is 2575211.
Fork 3. I'm the child 2575213, my parent is 2575207.
```

Hint:
Parent Process has to wait until Child Process finish.
Use PID to identify parent and child.

HW 1-3

Finish “hw1_3.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.

Important System Call:

- read(STDIN_FILENO, inputBuffer, MAX_LINE): read command line
- fork(): create child process
- execvp(char *command, char *params[]): execute system calls
- waitpid(pid)
- wait()

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

#define 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){
        printf("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()
        */
    }

    return 0;
}
```

HW 1-3

- **Change directory**

  $cd your/folder/

- **Compile**

  $gcc -o shell hw1_3.c

  or

  $g++ -o shell hw1_3.cpp

- **Execute**

  $./shell

You need

1. Finish “hw1_3.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. Your shell needs to support following commands: cat, ls, date, ps –f, exit (you can refer to pages 13).

Example

Note: ”usr” is your user name when you sign up

Show the content of hi.txt file

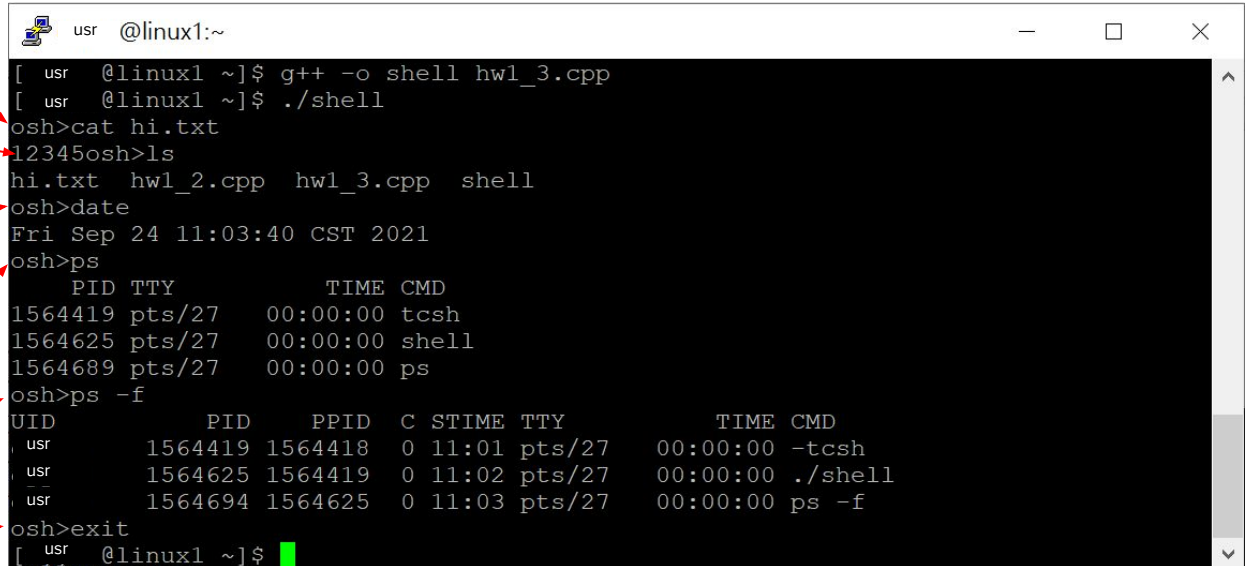
Show the file in the directory

Show the date

List all processes in current shell

Perform a full-format processes list

Enter “exit” to finish shell



Submission and Grade

Filename format please according : **hw1_2.c**, **hw1_3.c** (or .cpp), **OS_report.pdf**.
Put two *.c (*.cpp) files and a *.pdf report into same compressed file named **StudentID_hw1.zip** (ex : 0000000_hw1.zip).

Deadline: 2021/10/17 (Sun.) PM11:59

a. Total score: 100pts. **COPY WILL GET A 0 POINT!**
b. hw1-1 score: report Q1 30pts
c. hw1-2 score: code 25pts, report Q2 10pts
d. hw1-3 score: code 25pts, report Q3 10pts
e. Report: format is in OS_report.docx. YOU NEED TO FINISH EVERY PART OF REPORT TO GET SCORE!

Rules

- Use only C/C++, OTHER LANGUAGES WILL GET A 0 POINT
- Incorrect filename format will get -5 pts
- Incorrect output format will get -5 pts
- DELAYED SUBMISSION WILL GET A -20% POINT A Day!

* If you have any question, just send email to TAs by new E3.