

Name: Nguyễn Hữu Tứ

ID:19522453

Class:IT007.L21.1

OPERATING SYSTEM LAB 03 REPORT

SUMMARY

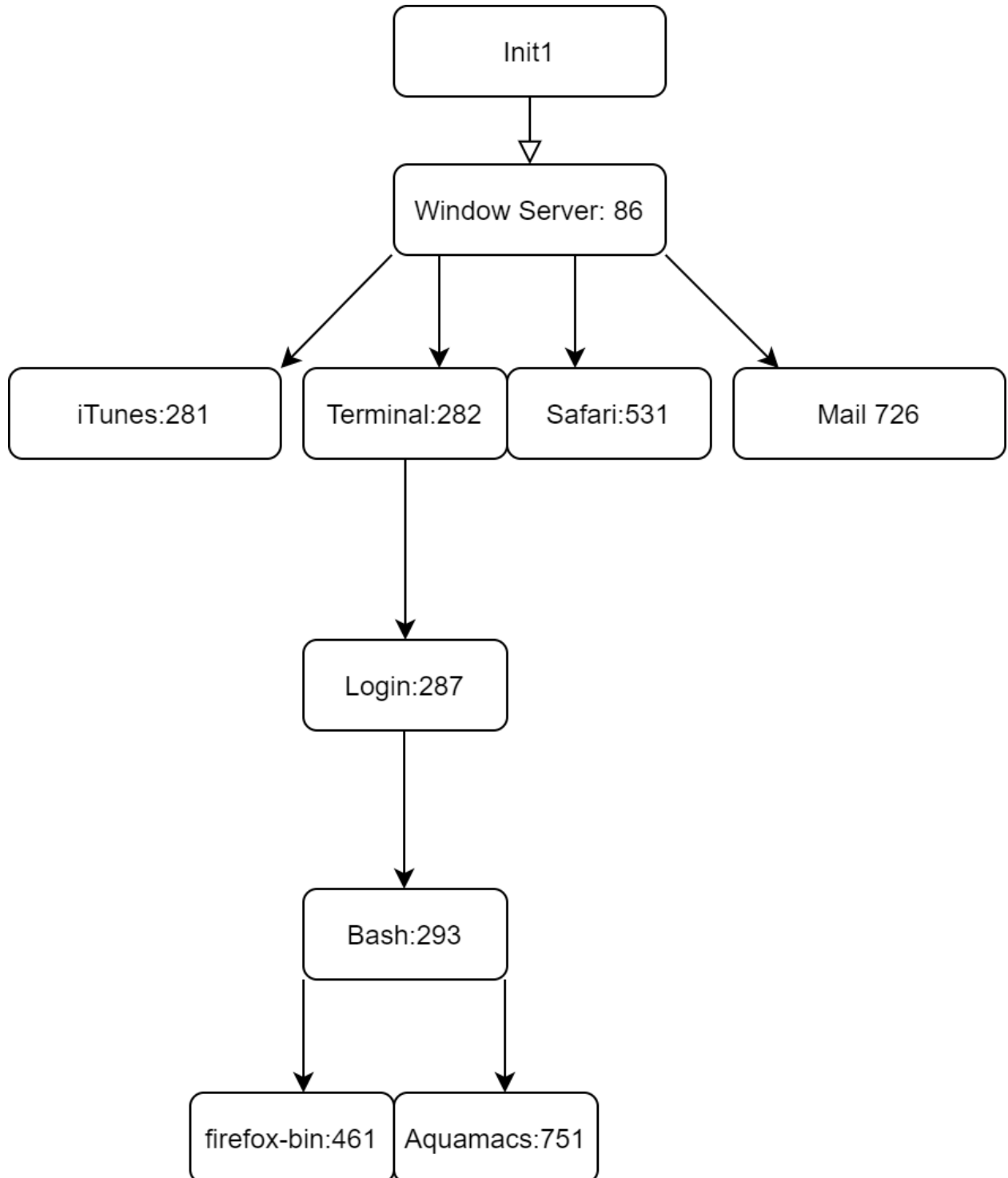
Task		Status	Page
	Task name 1	Done	2
	Task name 2	Done	3
	Task name 3	Done	4
	Task name 4	Done	7

Self-scores: 10.0

Section 1.5

1. Task name 1: Môi quan hệ cha-con giữa các tiến trình

Vẽ cây quan hệ parent-child của các tiến trình:



Cách sử dụng lệnh ps để tìm tiến trình cha của một tiến trình dựa vào PID của nó :

- Sử dụng `ps -f -p` để hiển thị đầy đủ thông tin của một tiến trình dựa vào PID của nó:

- PID: ID của tiến trình
- PPID: ID của tiến trình cha của nó

```
tu-19522453@ubuntu:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
tu-1952+      3128      3120  0  10:32 pts/0        00:00:00 bash
tu-1952+      4190      3128  0  13:02 pts/0        00:00:00 ps -f
tu-19522453@ubuntu:~$ ps -f -p 3128
UID          PID     PPID  C  STIME TTY          TIME CMD
tu-1952+      3128      3120  0  10:32 pts/0        00:00:00 bash
tu-19522453@ubuntu:~$
```

Tìm hiểu và cài đặt lệnh `pstree` (nếu chưa được cài đặt), sau đó trình bày cách sử dụng lệnh này để tìm tiến trình cha của một tiến trình dựa vào PID của nó

- Sử dụng `pstree -s -p` để hiển thị một phần của cây tiến trình, cho phép xem PID tiến trình cha của một tiến trình chỉ định

```
tu-19522453@ubuntu:~$ ps -f
UID          PID     PPID  C  STIME TTY          TIME CMD
tu-1952+      3128      3120  0  10:32 pts/0        00:00:00 bash
tu-1952+      4963      3128  0  13:15 pts/0        00:00:00 ps -f
tu-19522453@ubuntu:~$ pstree -s -p 3128
systemd(1)──systemd(1333)──gnome-terminal-(3120)──bash(3128)──pstree(4964)
tu-19522453@ubuntu:~$
```

Pid tiến trình cha Pid tiến trình chỉ định

2. Task name 2:

```
tu-19522453@ubuntu: ~/LAB03

/*#####
#University Of Information Technology #
# Nguyen Huu Tu , 19522453           #
# File : excercise_2.c               #
# IT007 Operating System             #
#####*/
#include <stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/wait.h>

int main()
{
    pid_t pid;
    int num_coconuts = 17;
    pid = fork();

    if(pid == 0)
    {
        num_coconuts = 42 ;
        exit(0);
    } else {
        wait(NULL); /*wait until the child terminates*/
    }
    printf("I see %d coconuts!\n",num_coconuts);
    exit(0);
}

tu-19522453@ubuntu:~/LAB03$ ./excercise_2
I see 17 coconuts!
tu-19522453@ubuntu:~/LAB03$
```

- Chương trình sẽ in ra dòng chữ “I see 17 coconuts !”
- Giải thích :

Đầu tiên num_coconuts = 17

Khi đến dòng lệnh pid = fork() thì sinh ra 1 tiến trình con. Gán pid = giá trị trả về của hàm fork() => pid của tiến trình con = 0, và pid của tiến trình cha > 0.

Sau đó, cả 2 tiến trình cùng chạy, tiến trình cha đến dòng lệnh wait(NULL) sẽ đợi cho đến khi tiến trình con terminated.

Tiến trình con gán num_coconuts = 42 nhưng khi chạy đến dòng lệnh exit(0) đã bị thoát ra. Lúc này giá trị num_coconuts ở tiến trình cha vẫn là 17.

Dòng lệnh printf("I see %d coconuts!\n", num_coconuts); chỉ có tiến trình cha thực hiện vì tiến trình con sau khi thực hiện lệnh exit(0) đã bị thoát ra.

3. Task name 3: Các hàm được sử dụng để làm thay đổi thuộc tính thread

pthread_attr_init: dùng để khởi tạo đối tượng thuộc tính với giá trị mặc định của chúng. Các thuộc tính có thể là: scope, detachstate, stackaddr, stacksize, priority, inheritshed, shedpolicy.

pthread_attr_destroy: dùng giải phóng tài nguyên của đối tượng thuộc tính đã được tạo.

pthread_attr_setdetachstate: thiết lập thuộc tính tách được

pthread_attr_getdetachstate: lấy thuộc tính tách được của đối tượng

pthread_attr_setguardsize: thiết lập guard size (??) của đối tượng thuộc tính

pthread_attr_setscope: để tạo thread có giới hạn (bound) hoặc thread không giới hạn (unbound).

pthread_attr_getscope: lấy về giá trị thread scope, từ đó biết được là thread là bound hay unbound.

pthread_attr_setschedpolicy: để thiết lập thuật toán lập lịch tiến trình. Có thể là SHED_FIFO (first-in-first-out), SCHED_RR (round-robin) or SCHED_OTHER.

pthread_attr_getschedpolicy: lấy về thuật toán lập lịch tiến trình hiện tại của hệ thống.

pthread_attr_setschedparams: thiết lập các tham số lập lịch.

pthread_attr_getschedparams: lấy về các tham số lập lịch đã được định nghĩa bởi hàm pthread_attr_setschedparam().

pthread_attr_setstacksize: thiết lập kích thước stack của thread.

pthread_attr_getstacksize: lấy về kích thước stack của thread được thiết lập bởi thread_attr_setstacksize().

pthread_attr_setstackaddr: thiết lập địa chỉ stack của thread.

pthread_attr_getstackaddr: trả về địa chỉ stack của thread đã được thiết lập bởi hàm pthread_attr_setstackaddr.

4. Task name 4: Viết chương trình theo yêu cầu của đề bài :

*Giải thích code :

Đầu tiên ta vào hàm main() in ra dòng chữ “ Welcom to IT007, I am 19522453” Sau đó dùng system để thao tác lệnh với Shell dùng lệnh gnome để bật file vim abcd.txt lên một terminal mới. Dùng hàm signal để nhận phím Ctrl+C nếu không nhấn thì chương trình sẽ vào hàm while và không làm gì. Đợi cho người dùng nhập Ctrl+C thì thoát cửa sổ terminal abcd.txt. Sau đó cho loop=1 để thực hiện lại vòng lặp. Trong vòng lặp ta không làm gì, để chờ người nhập nhấn Ctrl+C để xuất ra dòng chữ “You are pressed Ctrl + C ! Goodbye”.

```

# Name: Nguyen Huu Tu #
# ID: 19522453 #
#####*/
#include <stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<signal.h>
int loop =1;
void main_exit()
{
    printf("\nYou are pressed CTRL+C! Goodbye\n");
    loop = 0;
}

void vim_exit()
{
    loop = 0;
    system("pkill vim");
}

int main()
{
    printf("Welcome to IT007, I am 19522453\n");
    system("gnome-terminal --tab -e 'vim abcd.txt'");
    signal(SIGINT,vim_exit);
    while(loop){}
    loop=1;
    signal(SIGINT,main_exit);
    while (loop) {}
    return 0;
}
-- INSERT --

```

a) In ra dòng chữ: “Welcome to IT007, I am !”

```

tu-19522453@ubuntu:~/LAB03/THREAD$ vim excercise_4.c
tu-19522453@ubuntu:~/LAB03/THREAD$ chmod +x excercise_4.c
tu-19522453@ubuntu:~/LAB03/THREAD$ gcc excercise_4.c -o excercise_4
tu-19522453@ubuntu:~/LAB03/THREAD$ ./excercise_4
Welcome to IT007, I am 19522453

```

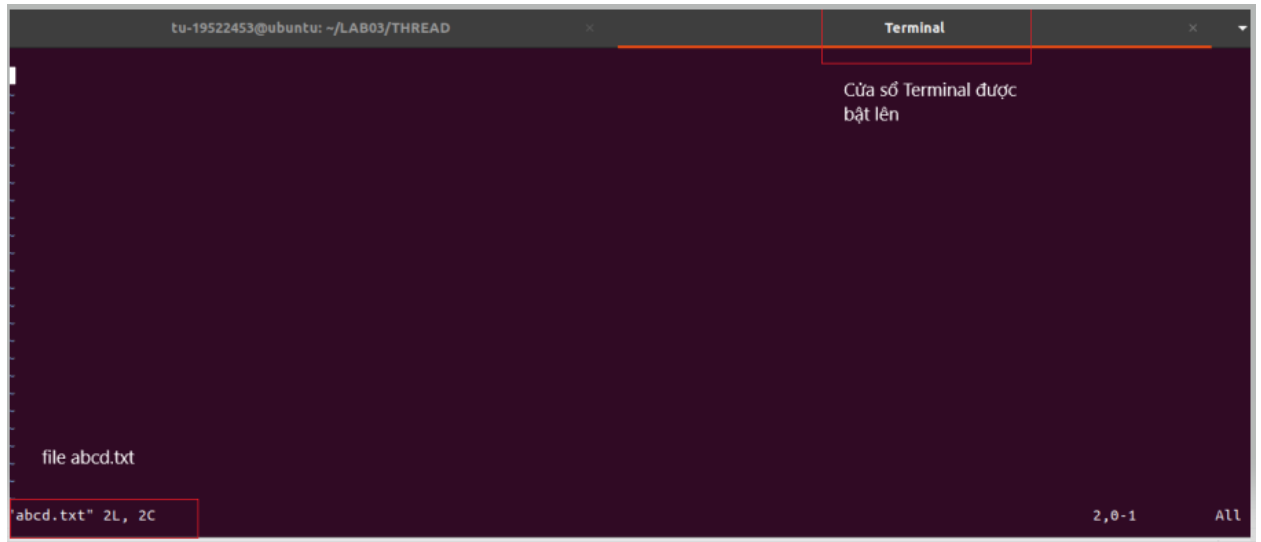
b) Mở tệp tệp abcd.txt bằng vim editor

```

tu-19522453@ubuntu:~/LAB03/THREAD$ vim excercise_4.c
tu-19522453@ubuntu:~/LAB03/THREAD$ gcc excercise_4.c -o excercise_4
tu-19522453@ubuntu:~/LAB03/THREAD$ ./excercise_4
Welcome to IT007, I am 19522453

```

Sau đó terminal của vim abcd.txt được bật lên



- c) Tắt vim editor khi người dùng nhấn CTRL+C

```
tu-19522453@ubuntu:~/LAB03/THREAD$ ./excercise_4
Welcome to IT007, I am 19522453
^C
```

- d) Khi người dùng nhấn CTRL+C thì in ra dòng chữ: "You are pressed CTRL+C! Goodbye!"

```
tu-19522453@ubuntu:~/LAB03/THREAD$ vim excercise_4.c
tu-19522453@ubuntu:~/LAB03/THREAD$ gcc excercise_4.c -o excercise_4
tu-19522453@ubuntu:~/LAB03/THREAD$ ./excercise_4
Welcome to IT007, I am 19522453
# Option "-e" is deprecated and might be removed in a later version of gnome-terminal.
# Use "--" to terminate the options and put the command line to execute after it.
^C
^C
You are pressed CTRL+C! Goodbye
tu-19522453@ubuntu:~/LAB03/THREAD$
```