

LỚP: IT007.N17.1

MSSV: 21522683

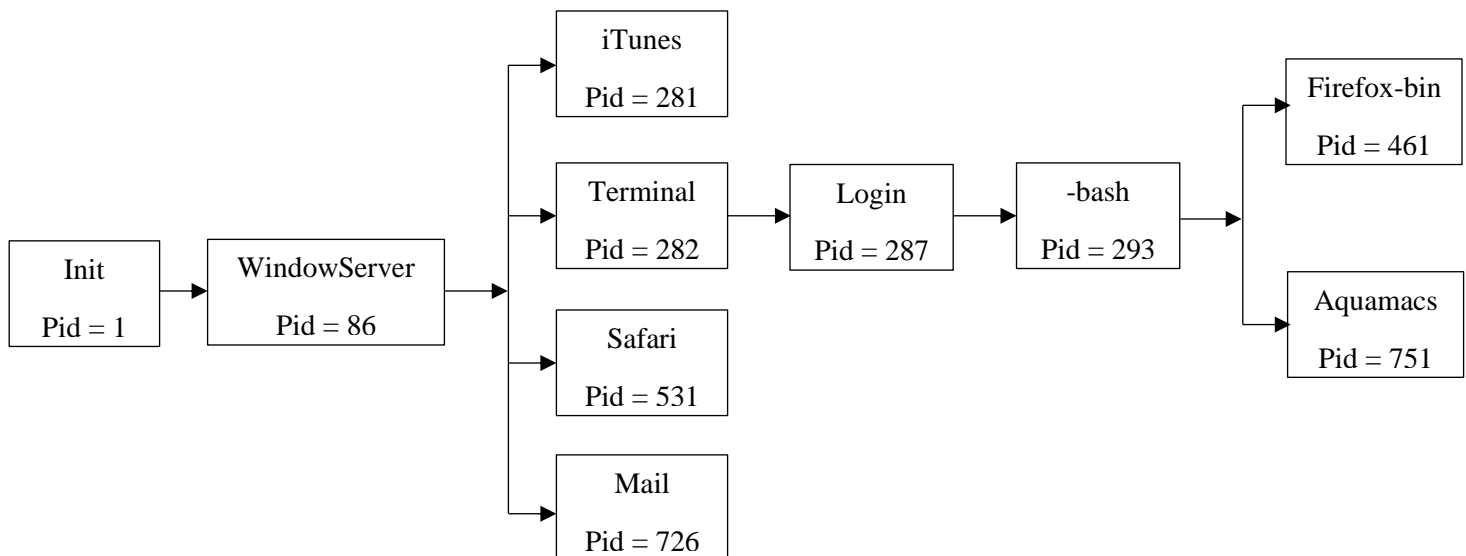
Họ Tên: Phan Trọng Tính

BÁO CÁO HỆ ĐIỀU HÀNH LAB3

Câu 1: Mối quan hệ cha-con giữa các tiến trình.

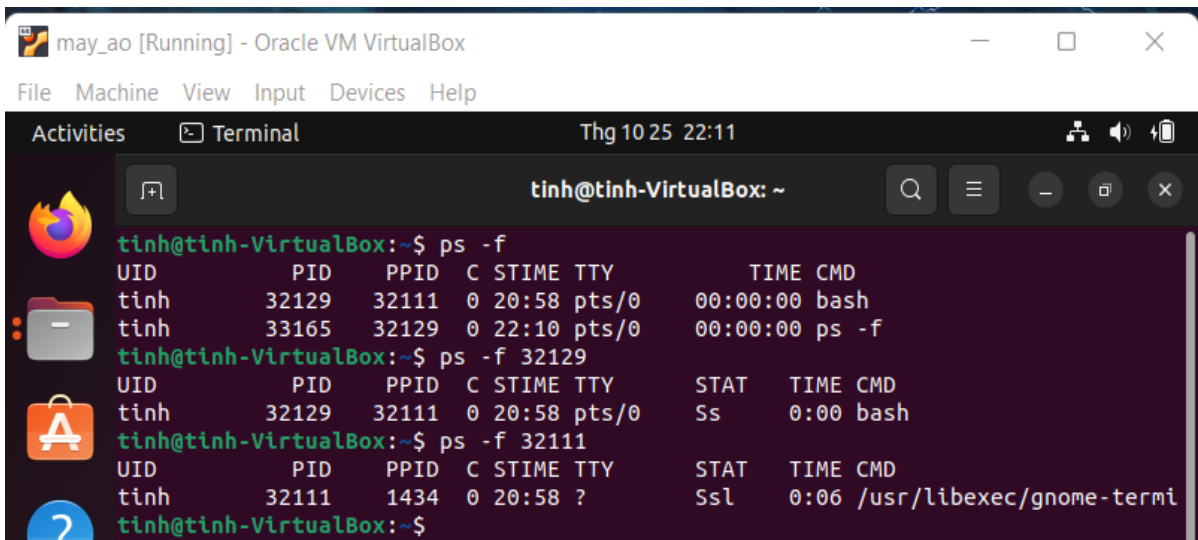
a. Vẽ cây quan hệ parent-child của các tiến trình bên dưới:

UID	PID	PPID	COMMAND
88	86	1	WindowServer
501	281	86	iTunes
501	282	86	Terminal
0	287	282	login
501	461	293	firefox-bin
501	531	86	Safari
501	726	86	Mail
501	751	293	Aquamacs
501	293	287	-bash



b. Trình bày cách sử dụng lệnh ps để tìm tiến trình cha của một tiến trình dựa vào PID của nó.

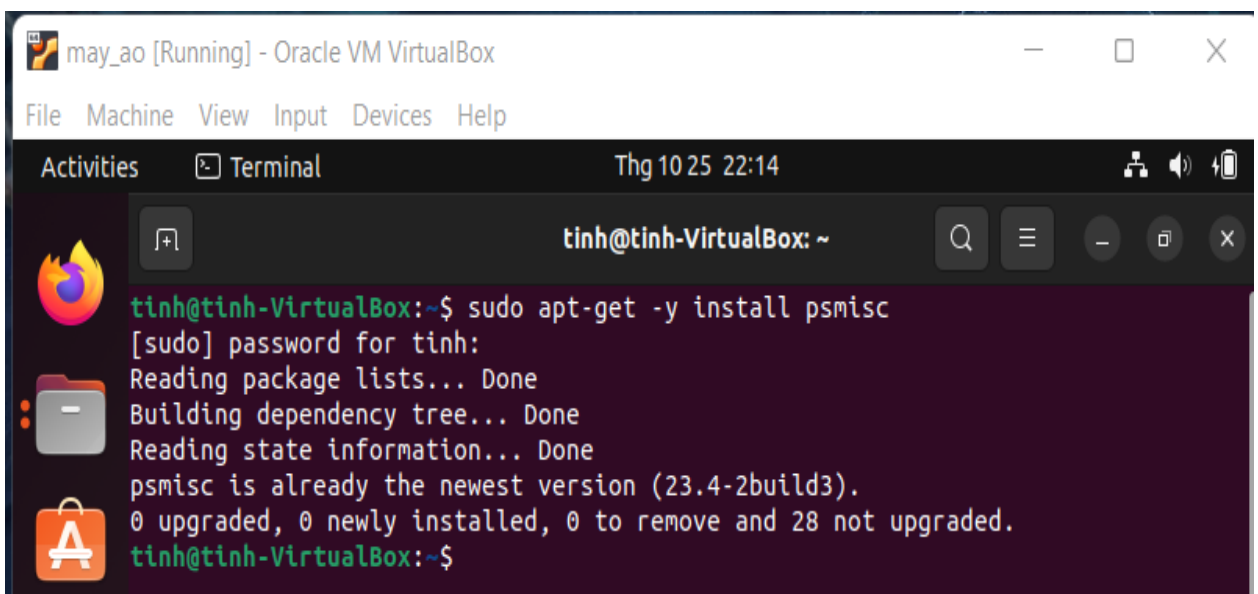
- Ta sẽ sử dụng lệnh `ps -f [pid của tiến trình]`, lúc này thông tin của tiến trình sẽ hiện ra và PPID chính là PID của tiến trình cha.
- Ta sẽ check thông tin của tiến trình cha bằng lệnh `ps -f [PPID]`.



```
tinh@tinh-VirtualBox:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
tinh         32129   32111  0  20:58 pts/0    00:00:00 bash
tinh         33165   32129  0  22:10 pts/0    00:00:00 ps -f
tinh@tinh-VirtualBox:~$ ps -f 32129
UID          PID    PPID  C STIME TTY          TIME CMD
tinh         32129   32111  0  20:58 pts/0    Ss      0:00 bash
tinh@tinh-VirtualBox:~$ ps -f 32111
UID          PID    PPID  C STIME TTY          TIME CMD
tinh         32111   1434  0  20:58 ?        Ssl     0:06 /usr/libexec/gnome-termi
tinh@tinh-VirtualBox:~$
```

c. 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.

Đầu tiên, để sử dụng lệnh pstree ta cần cài đặt lệnh pstree bằng câu lệnh `sudo apt-get -y install psmisc`.



```
tinh@tinh-VirtualBox:~$ sudo apt-get -y install psmisc
[sudo] password for tinh:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
psmisc is already the newest version (23.4-2build3).
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.
tinh@tinh-VirtualBox:~$
```

Ta gõ lệnh **ps tree -p** để xem cây tiến trình hiện tại.

```
tinh@tinh-VirtualBox:~$ ps tree -p
systemd(1)─ModemManager(761)─{ModemManager}(785)
          │                  │{ModemManager}(789)
          └─NetworkManager(672)─{NetworkManager}(704)
                                │{NetworkManager}(708)
          └─accounts-daemon(663)─{accounts-daemon}(714)
                                │{accounts-daemon}(717)
          └─acpid(664)
          └─avahi-daemon(667)─avahi-daemon(732)
          └─colord(1220)─{colord}(1221)
                       │{colord}(1223)
          └─cron(669)
          └─cups-browsed(844)─{cups-browsed}(866)
                             │{cups-browsed}(867)
          └─cupsd(764)
          └─dbus-daemon(671)
          └─gdm-session-wor(1292)─gdm-wayland-ses(1472)─gnome+
                                │                       │{gdm+
                                │                       │{gdm+
                                └─{gdm-session-wor}(1293)
                                │{gdm-session-wor}(1294)
                                └─{gdm3}(793)
                                  │{gdm3}(798)
          └─gnome-keyring-d(1451)─{gnome-keyring-d}(1452)
                                │{gnome-keyring-d}(1453)
                                └─{gnome-keyring-d}(1586)
          └─kerneloops(853)
```

Ta sẽ sử dụng **lệnh ps tree -p -s [pid của tiến trình con cần tìm tiến trình cha]**. Vd: Cần tìm tiến trình cha của tiến trình có PID là 1452 trong cây, ta gõ lệnh **ps tree -p -s 1452**.

```
tinh@tinh-VirtualBox:~$ ps tree -p -s 1452
wpa_supplicant(713)
└─systemd(1)─gnome-keyring-d(1451)─{gnome-keyring-d}(1452)
tinh@tinh-VirtualBox:~$
```

Lúc này dễ dàng thấy được PIDD của PID 1452 là 1451. Và gõ lại lệnh **ps tree -p -s 1451** để kiểm tra.

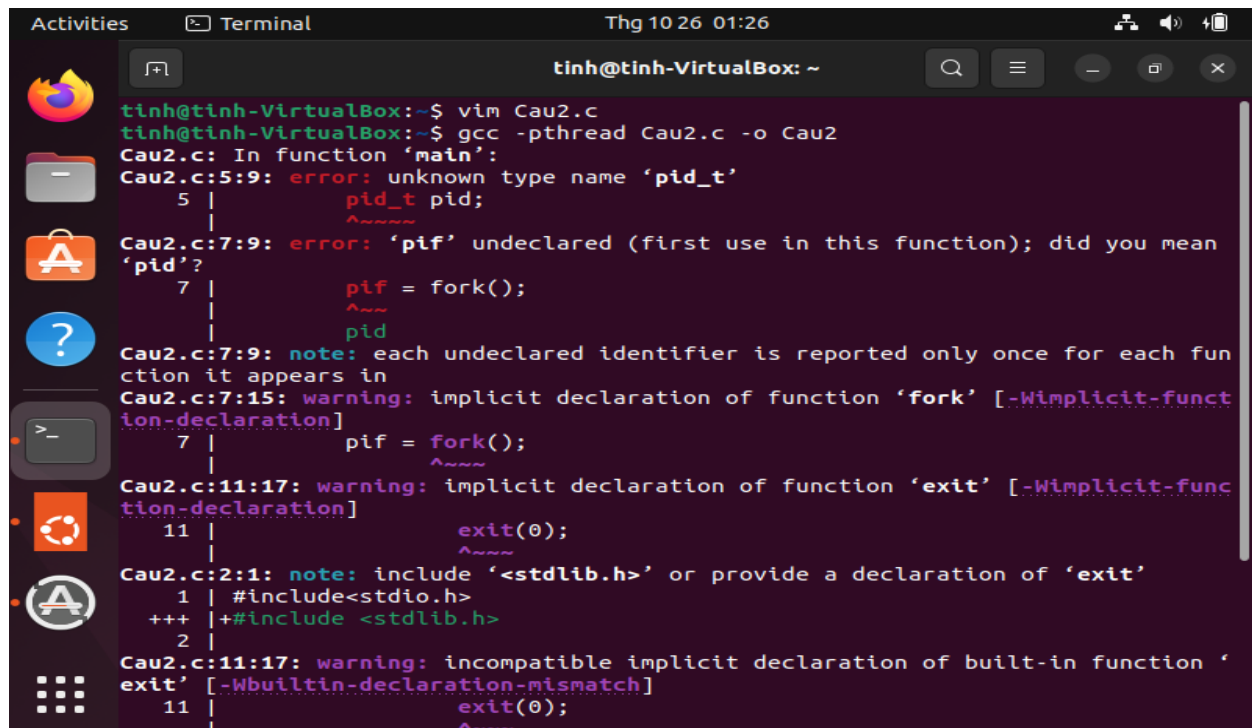
```
tinh@tinh-VirtualBox:~$ ps tree -p -s 1451
wpa_supplicant(713)
└─systemd(1)─gnome-keyring-d(1451)─{gnome-keyring-d}(1452)
tinh@tinh-VirtualBox:~$ ps tree -p -s 1451
systemd(1)─gnome-keyring-d(1451)─{gnome-keyring-d}(1452)
                                │{gnome-keyring-d}(1453)
                                └─{gnome-keyring-d}(1586)
tinh@tinh-VirtualBox:~$
```

Câu 2: Chương trình bên dưới in ra kết quả gì? Giải thích tại sao?

```
tinhh@tinhh-VirtualBox:~$ gedit Cau2
tinhh@tinhh-VirtualBox:~$ rm Cau2
rm: cannot remove 'Cau2': No such file or directory
tinhh@tinhh-VirtualBox:~$ ls
Desktop  Downloads  Pictures  snap      Videos
Documents Music      Public   Templates
tinhh@tinhh-VirtualBox:~$ gedit Cau2.c
tinhh@tinhh-VirtualBox:~$ cat Cau2.c
#include<stdio.h>

int main()
{
    pid_t pid;
    int num_coconuts = 17;
    pif = 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);
}
tinhh@tinhh-VirtualBox:~$
```

Kết quả chương trình chạy bị lỗi do thiếu 1 số thư viện như `stdlib.h`, `unistd.h`, `sys/wait.h`.



```
Activities  Terminal  Thg 10 26 01:26
tinhh@tinhh-VirtualBox: ~
tinhh@tinhh-VirtualBox:~$ vim Cau2.c
tinhh@tinhh-VirtualBox:~$ gcc -pthread Cau2.c -o Cau2
Cau2.c: In function 'main':
Cau2.c:5:9: error: unknown type name 'pid_t'
     5 |         pid_t pid;
       |         ^~~~~
Cau2.c:7:9: error: 'pif' undeclared (first use in this function); did you mean
     7 |         pif = fork();
       |         ^~~
Cau2.c:7:9: note: each undeclared identifier is reported only once for each fun
Cau2.c:7:15: warning: implicit declaration of function 'fork' [-Wimplicit-funct
     7 |         pif = fork();
       |         ^~~~~
Cau2.c:11:17: warning: implicit declaration of function 'exit' [-Wimplicit-func
    11 |         exit(0);
       |         ^~~~~
Cau2.c:2:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
     1 | #include<stdio.h>
+++ |+#include <stdlib.h>
     2 |
Cau2.c:11:17: warning: incompatible implicit declaration of built-in function '
    11 |         exit(0);
       |         ^~~~~
```

Sau khi thêm các thư viện thì kết quả chương trình in ra: “I see 17 coconuts!”.

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/wait.h>
#include<unistd.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);
}
```

```
tinhh@tinhh-VirtualBox:~$ vim Cau2.c
tinhh@tinhh-VirtualBox:~$ gcc -pthread Cau2.c -o Cau2
tinhh@tinhh-VirtualBox:~$ ./Cau2
I see 17 coconuts!
tinhh@tinhh-VirtualBox:~$
```

Kết quả chương trình in ra là “I see 17 coconuts!”. Bởi vì khi vào lệnh fork() thì tiến trình con và cha sẽ chạy. Với pid=0 thì tiến trình con thực hiện chương trình, còn tiến trình cha pid>0 sẽ vào lệnh wait(NULL) vậy nên tiến trình cha sẽ đợi đến khi tiến trình con chạy xong.

Tiến trình con thực hiện lệnh exit(0) trước khi thực hiện lệnh printf của tiến trình cha sẽ được thực thi, 2 tiến trình cha và con có bộ nhớ riêng vậy nên biến num_coconuts = 17. Thực hiện in ra “I see 17 coconuts!”.

Câu 3: Trong phần thực hành, các ví dụ chỉ sử dụng thuộc tính mặc định của pthread, hãy tìm hiểu POSIX thread và trình bày tất cả các hàm được sử dụng để làm thay đổi thuộc tính của pthread, sau đó viết các chương trình minh họa tác động của các thuộc tính này và chú thích đầy đủ cách sử dụng hàm này trong chương trình. (Gợi ý các hàm liên quan đến thuộc tính của pthread đều bắt đầu bởi: pthread_attr_*).

-POSIX thread (pthread) sẽ được sử dụng để lập tiểu trình. Nó cho phép chúng ta tạo ra các ứng dụng chạy song song theo luồng, phù hợp với các hệ thống đa bộ xử lý. POSIX (Portable Operating Systems Interface) là mô tả các API (Application Programming Interface) bao gồm hàm và chức năng của chúng.

-Các hàm được sử dụng để thay đổi thuộc tính của pthread là:

Thuộc tính	Giá trị	Kết quả
Scope	PTHREAD_SCOPE_PROCESS	- Thread mới cạnh tranh với các thread khác trong process này.
Detachstate	PTHREAD_CREATE_JOINABLE	- Trạng thái hoàn thành và thread ID được giữ nguyên sau khi thread thoát
Stackaddr	Null	- Thread mới có địa chỉ ngăn xếp do hệ thống cấp phát.
Stacksize	0	- Thread mới có kích thước ngăn xếp do hệ thống xác định.
Priority	0	- Thread mới có độ ưu tiên bằng 0
Inheritsched	PTHREAD_EXPLICIT_SCHED	- Thread mới không kế thừa quyền ưu tiên từ thread cha.
Schedpolicy	SCHED_OTHER	- Thread mới sử dụng chia sẻ theo thời gian (Solaris time-sharing)
Guardsize	PAGESIZE	- Chống tràn stack

-Các chương trình minh họa tác động của các thuộc tính:

+ Minh họa scope:

```
int pthread_attr_setscope(pthread_attr_t*attr, int scope);  
// đặt thuộc tính phạm vi tranh chấp của đối tượng thuộc //tính luồng  
//được tham chiếu bởi attr đến giá trị được chỉ định //trong phạm vi  
int pthread_attr_getscope(const pthread_attr_t*attr, int scope);  
// thành công thì return 0, ngược lại thì lỗi.
```

+ Minh họa priority

```
int pthread_attr_setshedpolicy(const pthread_attr_t*attr, int policy);  
//Thuộc tính luồng được tham chiếu bởi attr đến giá trị policy.  
int pthread_attr_getshedpolicy(const pthread_attr_t*restrict attr, int*  
restrictpolicy);  
// thành công thì return 0, ngược lại thì lỗi.
```

+ Minh họa setguardsize:

```
S=pthread_attr_getguardsize(pthread_attr_t*attr, &sp);  
//Đặt kích thước bảo vệ của đối tượng attr.
```

+ Minh họa destroy:

```
Int pthread_attr_destroy (pthread_attr_t* tattr);  
//Để xóa bộ nhớ đã được cấp phát trong quá trình khởi tạo. Đối tượng  
//thuộc tính trở nên không hợp lệ.
```

Câu 4: Viết chương trình làm các công việc sau theo thứ tự:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<signal.h>
#include<pthread.h>

long id_thread;
int loop_forever = 2;

void on_sigint()
{
    if(loop_forever==1)
    {
        printf("\nYOU are pressed CTRL+C! Goodbye!\n");
        loop_forever--;
    }
}

void on_sigint2()
{
    if(loop_forever==2)
        system("killall gedit");
    loop_forever--;
}

void *open_gedit(void *args)
{
    system("gedit abcd.txt");
}
```

```
void *open_gedit(void *args)
{
    system("gedit abcd.txt");
    signal(SIGINT, on_sigint2);
    while(loop_forever!=1){};
}

int main()
{
    printf("\nWelcome to IT007, I am 21522683!\n");
    pthread_create(&id_thread, NULL, &open_gedit, (void*)id_thread);
    pthread_join(id_thread, NULL);
    signal(SIGINT, on_sigint);
    while(loop_forever != 0){};
    print("\n");

    return 1;
}
```


a. ***In ra dòng chữ: “Welcome to IT007, I am <your Student ID>!”***

```
tinh@tinh-VirtualBox: ~  
tinh@tinh-VirtualBox:~$ gcc -pthread Cau4.c -o Cau4  
tinh@tinh-VirtualBox:~$ ./Cau4  
  
Welcome to IT007, I am 21522683!
```

b. ***Mở tệp abcd.txt bằng vim editor***



c. ***Tắt vim editor khi người dùng nhấn CTRL+C***

d. ***Khi người dùng nhấn CTRL+C thì in ra dòng chữ: “You are pressed CTRL+C! Goodbye!”***

