

Tên: Lê Anh Thư

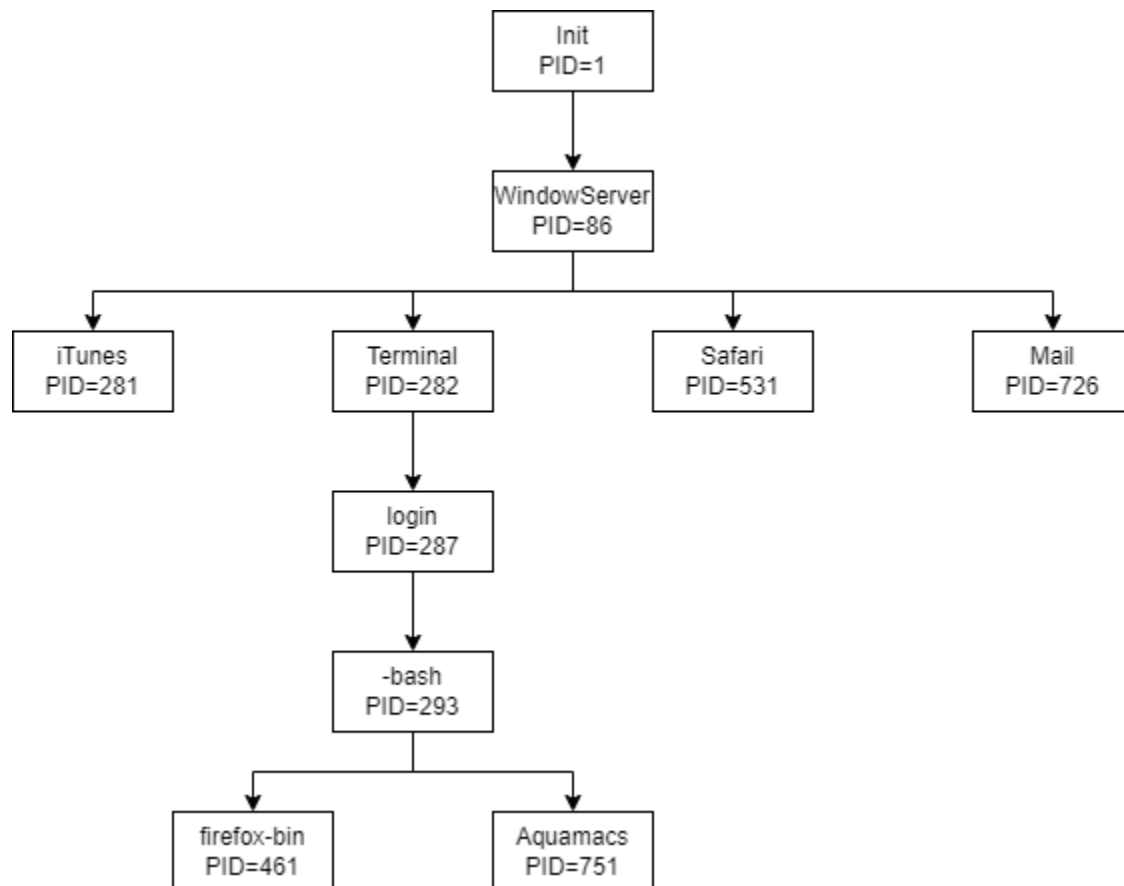
MSSV: 20521985

BÀI TẬP THỰC HÀNH BUỔI 3

Môn: Hệ Điều Hành

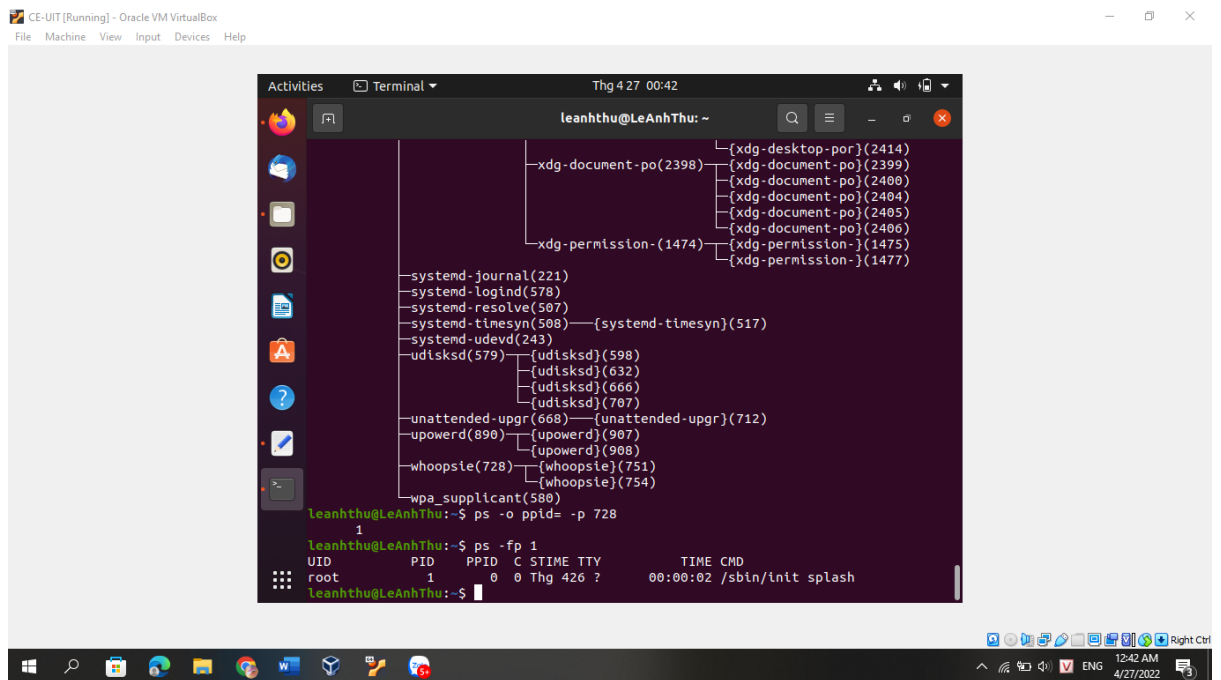
Câu 1

a. Cây quan hệ parent-child



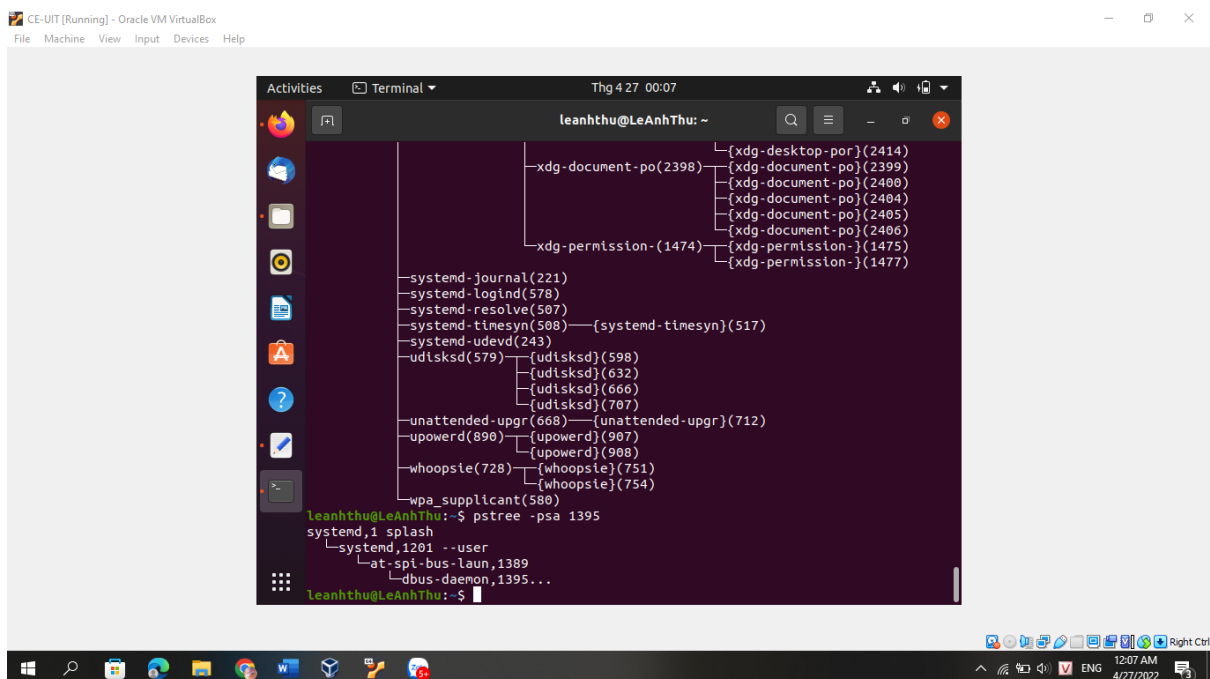
b. Sử dụng lệnh ps để tìm tiến trình cha của một tiến trình dựa vào PID

- ❖ Dùng lệnh “ps -o ppid= -p [PID của tiến trình con]” để tìm PID của tiến trình cha.
- ❖ Dùng lệnh “ps -fp [PID của tiến trình cha]” để lấy thông tin tiến trình cha.



c. Sử dụng pstree để tìm tiến trình cha dựa vào PID của một tiến trình

❖ Dùng lệnh “pstree -psa [PID của tiến trình con]”



Câu 2

Source code:

```

1 /*****
2 # University of Information Technology #
3 # IT007 Operating System #
4 # Le Anh Thu, 20521985 #
5 # File: exercise_2.c #
6 *****/
7
8 #include<stdio.h>
9 #include<unistd.h>
10 #include<stdlib.h>
11 #include<sys/wait.h>
12
13 int main()
14 {
15     pid_t pid;
16     int num_coconuts = 17;
17
18     pid = fork();
19     if(pid == 0){
20         num_coconuts = 42;
21         exit(0);
22     }
23     else{
24         wait(NULL);
25     }
26
27     printf("I see %d coconuts\n", num_coconuts);
28
29     exit(0);
30 }

```

Kết quả: Chương trình in ra “I see 17 coconuts!”

Giải thích: Biến `num_coconuts` trong process con thay đổi không ảnh hưởng biến `num_coconuts` ở process cha. Process con gặp `exit` nên kết thúc và không in ra màn hình. Process cha gặp `wait` đợi process kết thúc và in kết quả ra màn hình.

Câu 3

❖ Tất cả các hàm được sử dụng để làm thay đổi thuộc tính của pthread

Hàm	Khai báo	Công dụng
pthread_attr_t init	<code>int pthread_attr_init(pthread_attr_t *tattr);</code>	Tạo thuộc tính cho thread
pthread_attr_t destroy	<code>int pthread_attr_destroy(pthread_attr_t *tattr);</code>	Xóa thuộc tính của thread
pthread_attr_t setdetachstate	<code>int pthread_attr_setdetachstate(pthread_attr_t *tattr, int detachstate);</code>	Chọn trạng thái thread sẽ được tạo là joinable hay detached
pthread_attr_t setguardsize	<code>int pthread_attr_setguardsize(pthread_attr_t *attr, size_t guardsize);</code>	Thay đổi kích thước tối thiểu của vùng bảo vệ stack
pthread_attr_t setscope	<code>int pthread_attr_setscope(pthread_attr_t *tattr, int scope);</code>	Chọn phạm vi tranh chấp lập lịch cho thread

pthread_attr_setschedpolicy	int pthread_attr_setschedpolicy(pthread_attr_t *tattr, int policy);	Chọn giải thuật lập lịch cho thread
pthread_attr_setinheritsched	int pthread_attr_setinheritsched(pthread_attr_t *tattr, int inherit);	Chọn giải thuật lập lịch của thread sẽ kế thừa từ process cha hay tự xác định.
pthread_attr_setschedparam	int pthread_attr_setschedparam(pthread_attr_t *tattr, const struct sched_param *param);	Thay đổi độ ưu tiên lập lịch cho thread
pthread_attr_setstacksize	int pthread_attr_setstacksize(pthread_attr_t *tattr, int size);	Thay đổi kích thước stack được cấp cho thread
pthread_attr_setstackaddr	int pthread_attr_setstackaddr(pthread_attr_t *tattr, void *stackaddr);	Cung cấp địa chỉ cho stack của thread

❖ Ví dụ

• pthread_attr_setstackaddr

```
#include <pthread.h>

pthread_attr_t tattr;
void *base;
int ret;

base = (void *) malloc(PTHREAD_STACK_MIN + 0x4000);

/* setting a new address */
ret = pthread_attr_setstackaddr(&tattr, base);
```

• pthread_attr_init

```
#include <pthread.h>

pthread_attr_t tattr;
int ret;

/* initialize an attribute to the default value */
ret = pthread_attr_init(&tattr);
```

• pthread_attr_setdetachstate

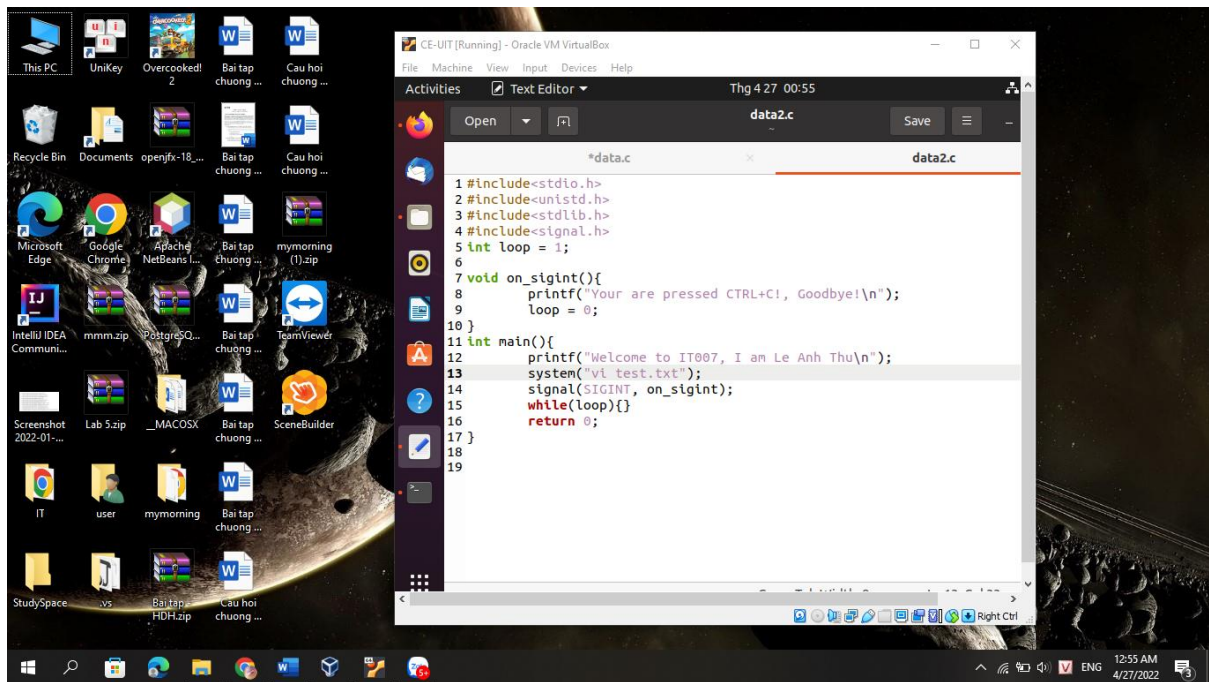
```
#include <pthread.h>
```

```
pthread_attr_t tattr;
pthread_t tid;
void *start_routine;
void arg;
int ret;

/* initialized with default attributes */
ret = pthread_attr_init() (&tattr);
ret =
pthread_attr_setdetachstate() (&tattr, PTHREAD_CREATE_DETACHED);
ret = pthread_create() (&tid, &tattr, start_routine, arg);
```

Câu 4

❖ Cách 1



Source code:

```
#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<signal.h>

int loop = 1;

void on_sigint(){

    printf("Your are pressed CTRL+C!, Goodbye!\n");

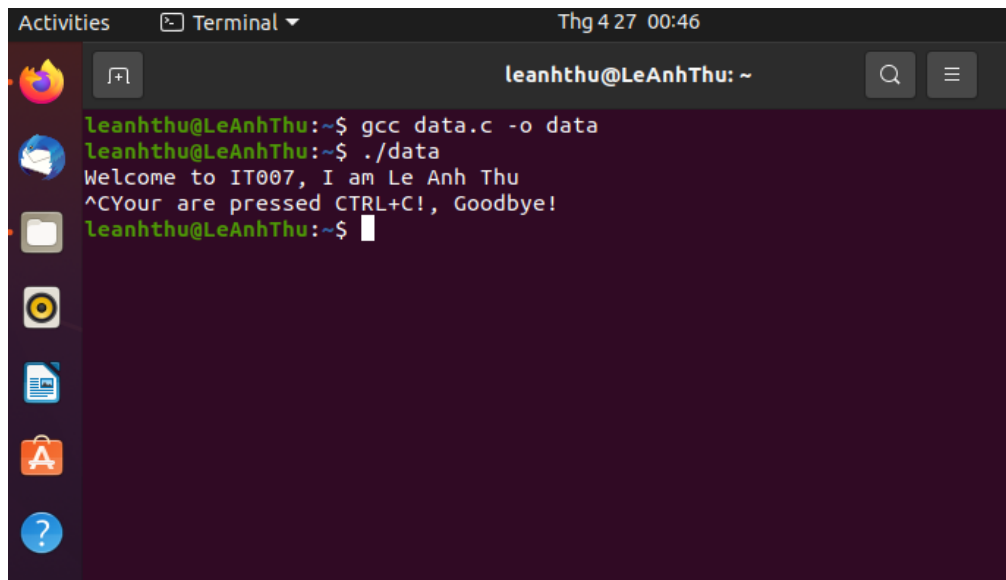
    loop = 0;

}

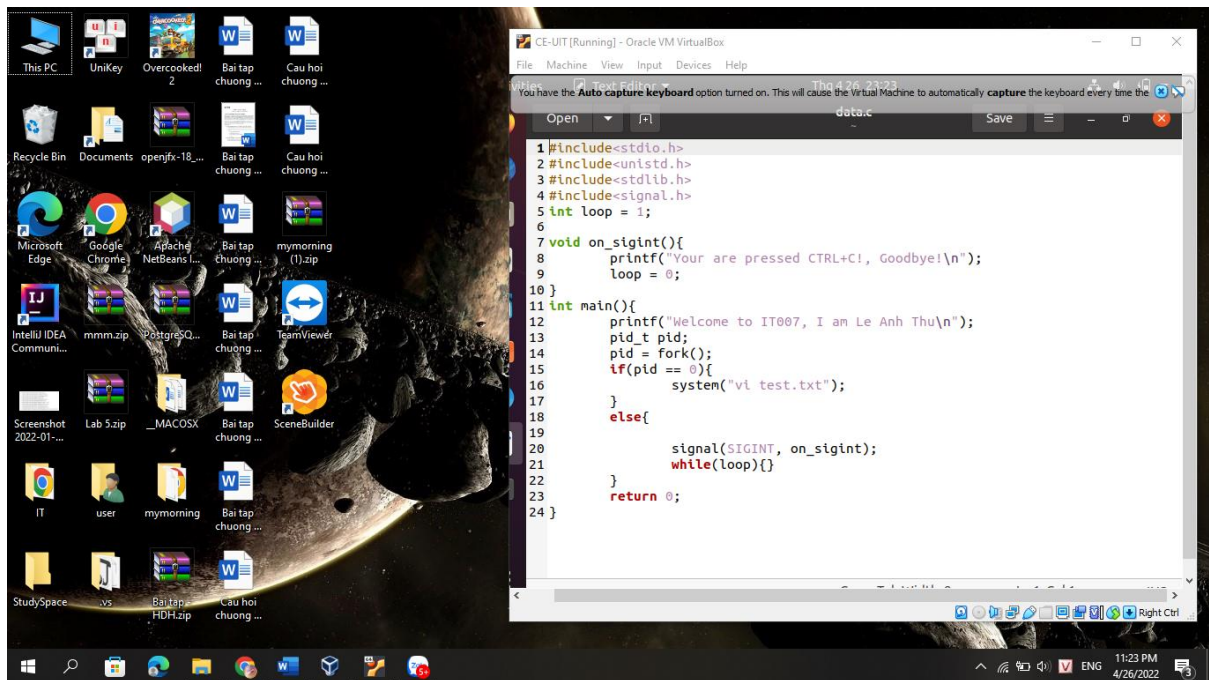
int main(){

    printf("Welcome to IT007, I am Le Anh Thu\n");
```

Kết quả :



❖ **Cách 2**



Source code:

```
#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<signal.h>

int loop = 1;

void on_sigint(){

    printf("Your are pressed CTRL+C!, Goodbye!\n");

    loop = 0;

}

int main(){

    printf("Welcome to IT007, I am Le Anh Thu\n");

    pid_t pid;

    pid = fork();

    if(pid == 0){

        system("vi test.txt"); }

    else{

        signal(SIGINT, on_sigint);

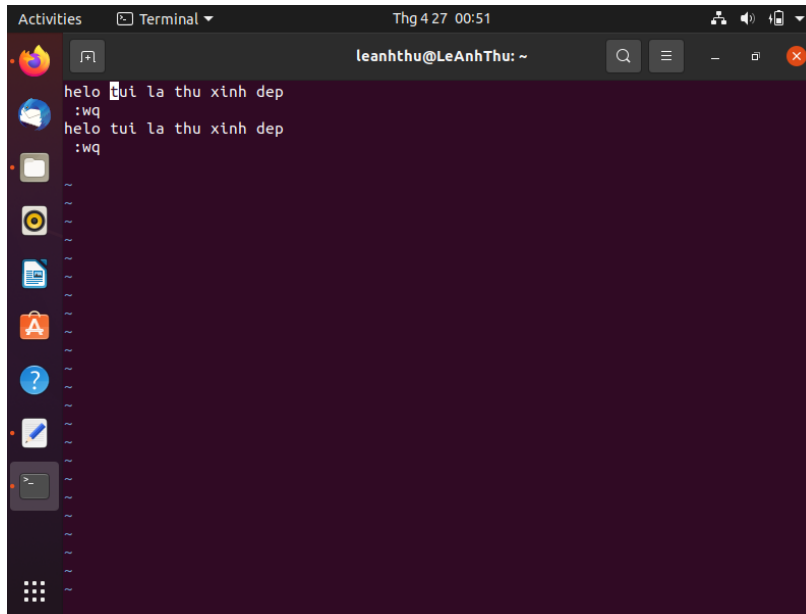
        while(loop){}

    }

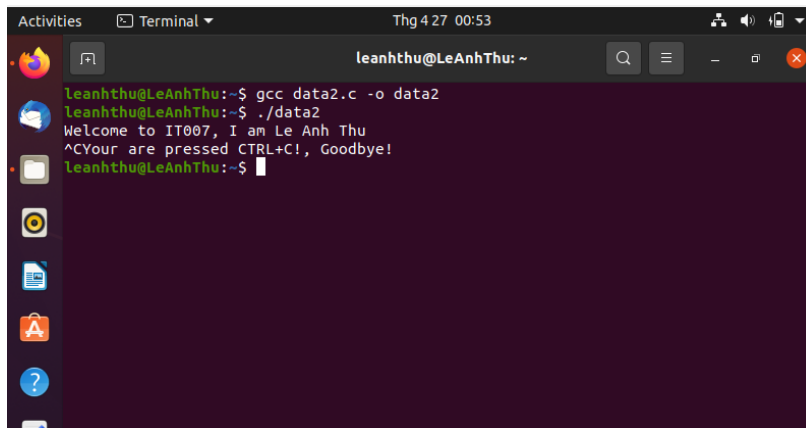
}
```

```
    return 0;
}
```

Kết quả:



```
leanhthu@LeAnhThu: ~  
helo tui la thu xinh dep  
:wq  
helo tui la thu xinh dep  
:wq
```



```
leanhthu@LeAnhThu:~$ gcc data2.c -o data2  
leanhthu@LeAnhThu:~$ ./data2  
Welcome to IT007, I am Le Anh Thu  
^CYour are pressed CTRL+C!, Goodbye!  
leanhthu@LeAnhThu:~$
```