

Lab 3

BÁO CÁO BÀI THỰC HÀNH SỐ 3

TIẾN TRÌNH VÀ TIỂU TRÌNH

Môn học: Hệ điều hành (IT007)

Sinh viên thực hiện	Nguyễn Đức Tấn
Thời gian thực hiện	13/04/2024

THỰC HÀNH:

1. Cài đặt trình soạn thảo văn bản vim và trình biên dịch gcc:

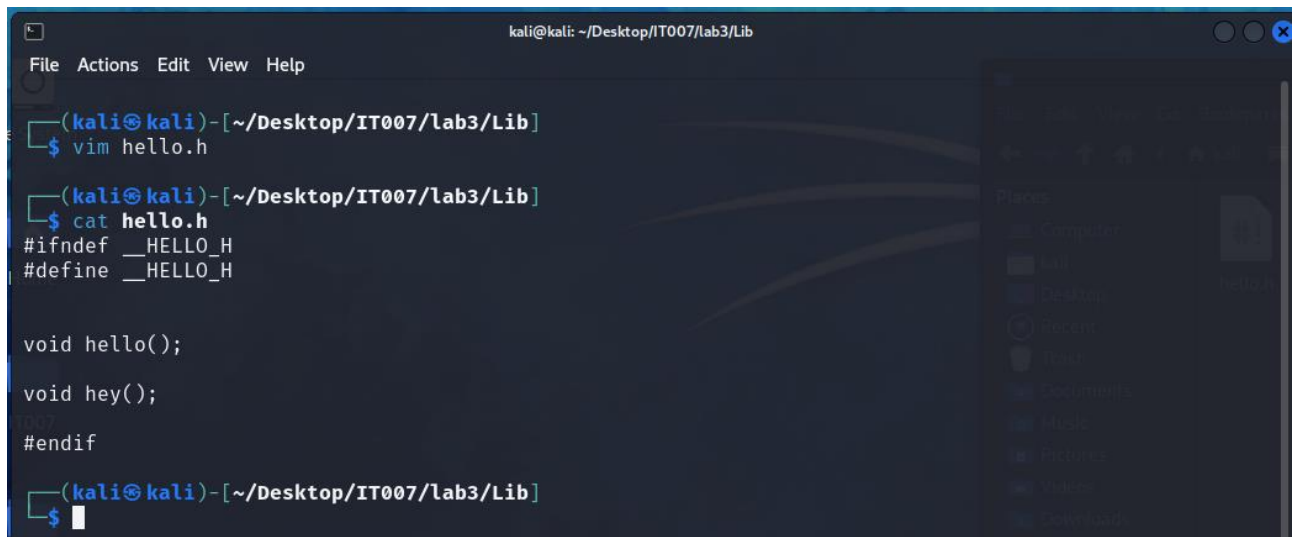
```
Processing triggers for man-db (2.12.0-1) ...  
  
(root@kali)-[/home/kali]  
# which vim  
/usr/bin/vim  
  
(root@kali)-[/home/kali]  
# which gcc  
/usr/bin/gcc  
  
(root@kali)-[/home/kali]  
#
```

Sử dụng vim để soạn thảo chương trình hello.c:

```
kali@kali: ~/Desktop/IT007/lab3  
File Actions Edit View Help  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$ touch file1.c  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$ vim fil1.c  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$ cat fil1.c  
#include <stdio.h>  
  
int main() {  
    printf("Hello, I am Duc Tan\n");  
    printf("Welcome to IT007!\n");  
    return 0;  
}  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$ gcc fil1.c -o f1  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$ ./f1  
Hello, I am Duc Tan  
Welcome to IT007!  
  
(kali@kali)-[~/Desktop/IT007/lab3]  
$
```

Chuyển chương trình hello.c thành 1 thư viện:

Lab 3: Tiến trình và tiểu trình



The screenshot shows a terminal window with the title bar "kali@kali: ~/Desktop/IT007/lab3/Lib". The terminal output is as follows:

```
kali@kali:~/Desktop/IT007/lab3/Lib$ vim hello.h

(kali@kali)~[~/Desktop/IT007/lab3/Lib]
$ cat hello.h
#ifndef __HELLO_H
#define __HELLO_H

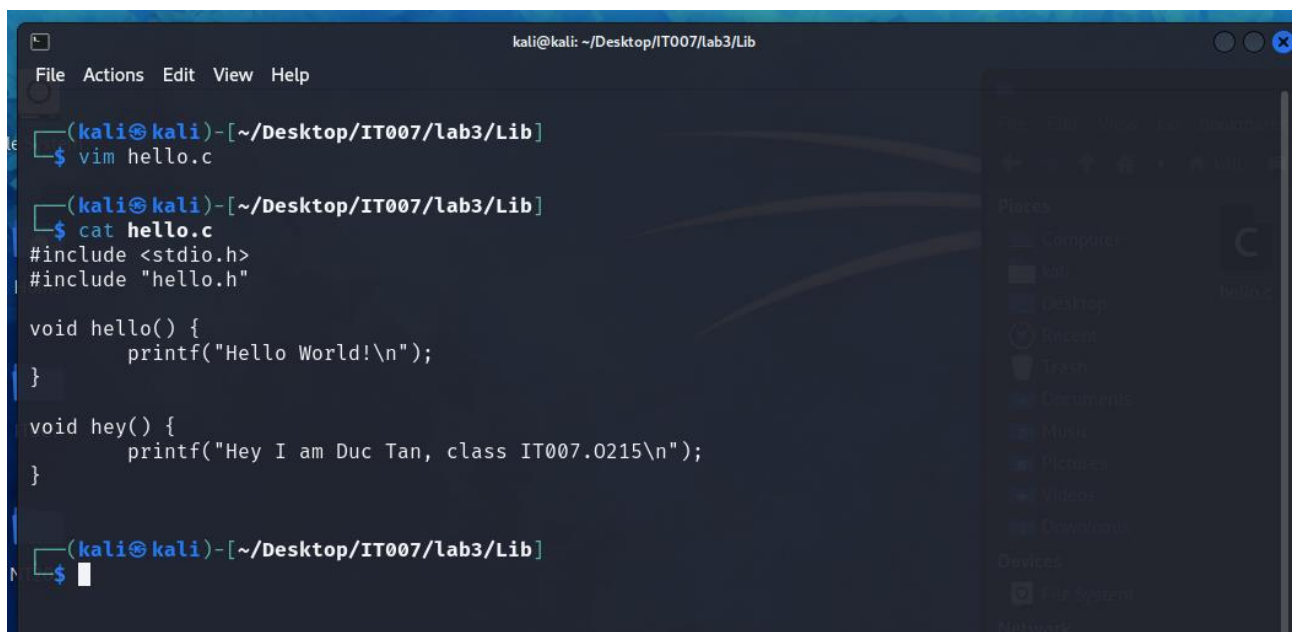
void hello();

void hey();

#endif

(kali@kali)~[~/Desktop/IT007/lab3/Lib]
$
```

A file manager sidebar is visible on the right, showing the current directory and its contents, including the file "hello.h".



The screenshot shows a terminal window with the title bar "kali@kali: ~/Desktop/IT007/lab3/Lib". The terminal output is as follows:

```
kali@kali:~/Desktop/IT007/lab3/Lib$ vim hello.c

(kali@kali)~[~/Desktop/IT007/lab3/Lib]
$ cat hello.c
#include <stdio.h>
#include "hello.h"

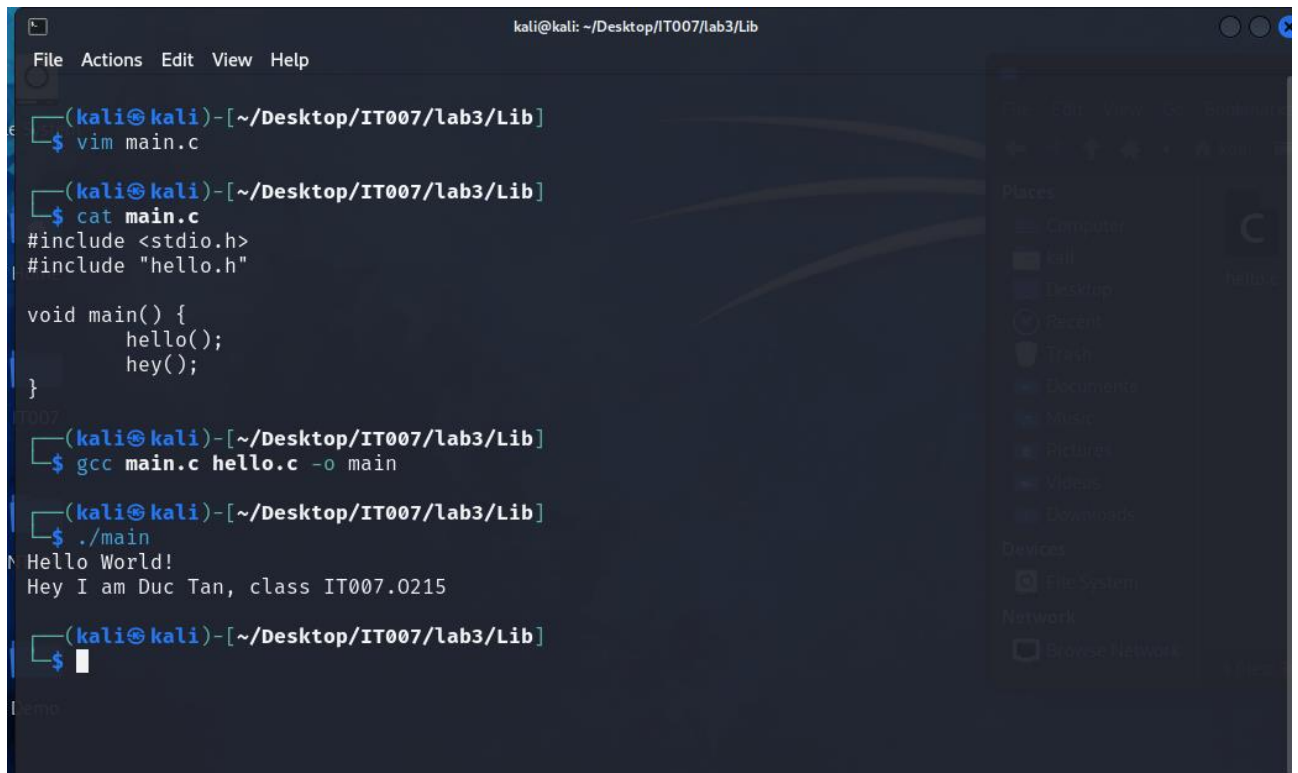
void hello() {
    printf("Hello World!\n");
}

void hey() {
    printf("Hey I am Duc Tan, class IT007.0215\n");
}

(kali@kali)~[~/Desktop/IT007/lab3/Lib]
$
```

A file manager sidebar is visible on the right, showing the current directory and its contents, including the file "hello.c".

Lab 3: Tiến trình và tiểu trình



```
kali@kali: ~/Desktop/IT007/lab3/Lib
File Actions Edit View Help

(kali@kali)-[~/Desktop/IT007/lab3/Lib]
$ vim main.c

(kali@kali)-[~/Desktop/IT007/lab3/Lib]
$ cat main.c
#include <stdio.h>
#include "hello.h"

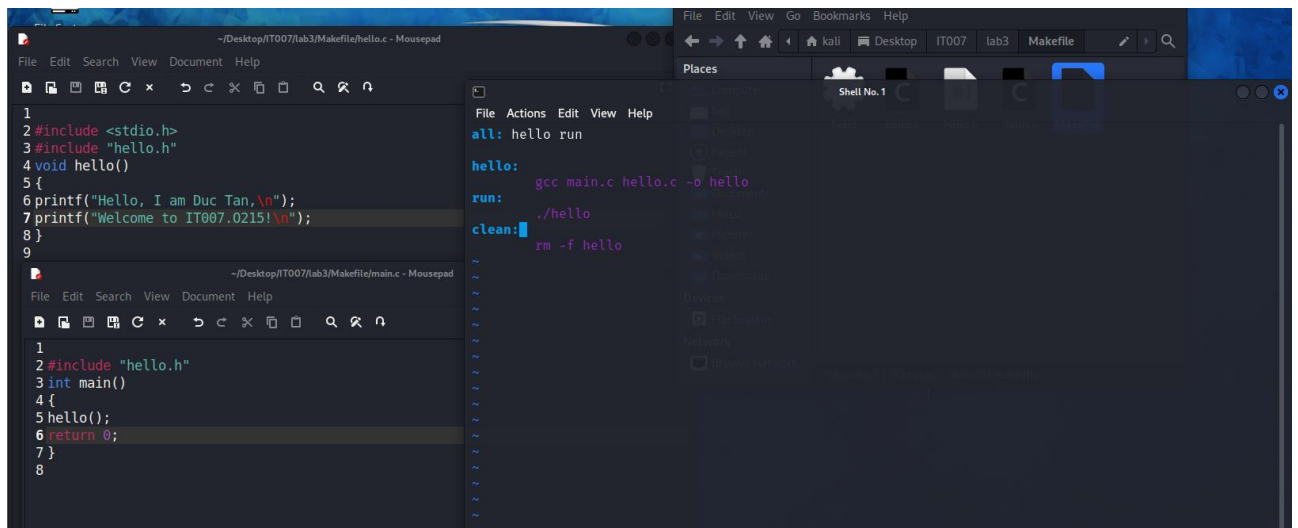
void main() {
    hello();
    hey();
}

(kali@kali)-[~/Desktop/IT007/lab3/Lib]
$ gcc main.c hello.c -o main

(kali@kali)-[~/Desktop/IT007/lab3/Lib]
$ ./main
Hello World!
Hey I am Duc Tan, class IT007.0215

(kali@kali)-[~/Desktop/IT007/lab3/Lib]
$
```

Makefile:



```
~/Desktop/IT007/lab3/Makefile/hello.c - Mousepad
File Edit Search View Document Help

1
2 #include <stdio.h>
3 #include "hello.h"
4 void hello()
5 {
6     printf("Hello, I am Duc Tan,\n");
7     printf("Welcome to IT007.0215!\n");
8 }
9

~/Desktop/IT007/lab3/Makefile/main.c - Mousepad
File Edit Search View Document Help

1
2 #include "hello.h"
3 int main()
4 {
5     hello();
6     return 0;
7 }
8

~/Desktop/IT007/lab3/Makefile
File Actions Edit View Help

all: hello run

hello:
    gcc main.c hello.c -o hello

run:
    ./hello

clean:
    rm -f hello
```



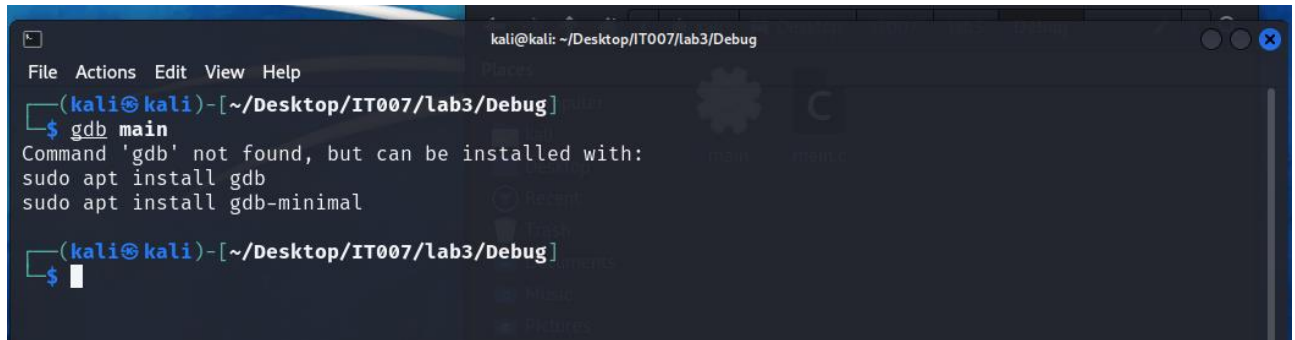
```
kali@kali: ~/Desktop/IT007/lab3/Makefile
File Actions Edit View Help

(kali@kali)-[~/Desktop/IT007/lab3/Makefile]
$ make all
./hello
Hello, I am <your name>,
Welcome to IT007!

(kali@kali)-[~/Desktop/IT007/lab3/Makefile]
$
```

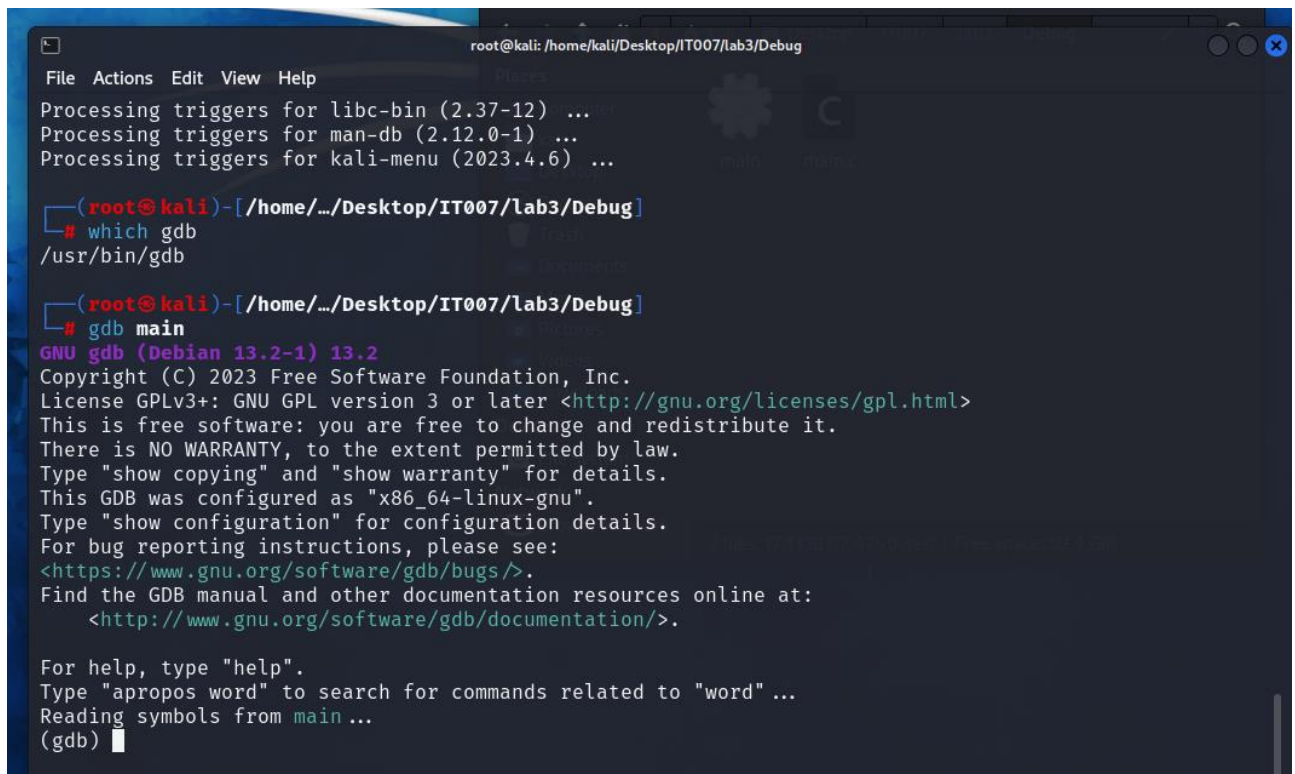
Trình gỡ lỗi (GDB)

Lỗi:



```
kali@kali: ~/Desktop/IT007/lab3/Debug
File Actions Edit View Help
(kali@kali)-[~/Desktop/IT007/lab3/Debug]
$ gdb main
Command 'gdb' not found, but can be installed with:
sudo apt install gdb
sudo apt install gdb-minimal
(kali@kali)-[~/Desktop/IT007/lab3/Debug]
$
```

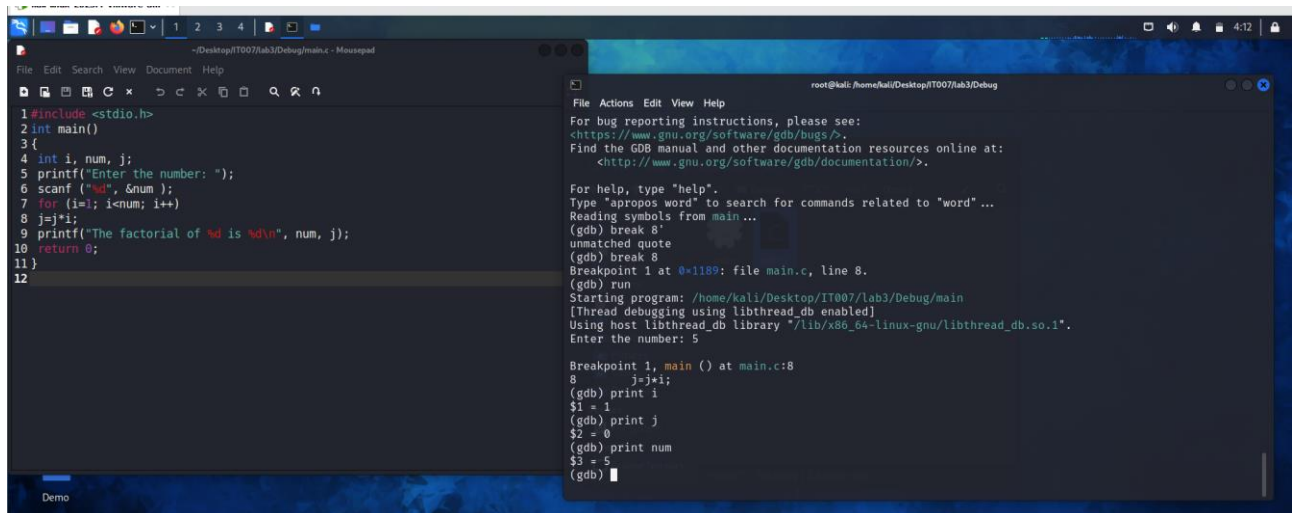
Giải quyết: cài đặt thêm công cụ gdb



```
root@kali: /home/kali/Desktop/IT007/lab3/Debug
File Actions Edit View Help
Processing triggers for libc-bin (2.37-12) ...
Processing triggers for man-db (2.12.0-1) ...
Processing triggers for kali-menu (2023.4.6) ...
(root@kali)-[/home/.../Desktop/IT007/lab3/Debug]
# which gdb
/usr/bin/gdb
(root@kali)-[/home/.../Desktop/IT007/lab3/Debug]
# gdb main
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from main...
(gdb)
```

Lab 3: Tiến trình và tiểu trình



```
1#include <stdio.h>
2int main()
3{
4    int i, num, j;
5    printf("Enter the number: ");
6    scanf ("%d", &num );
7    for (i=1; i<num; i++)
8        j=j*i;
9    printf("The factorial of %d is %d\n", num, j);
10   return 0;
11}
12
```

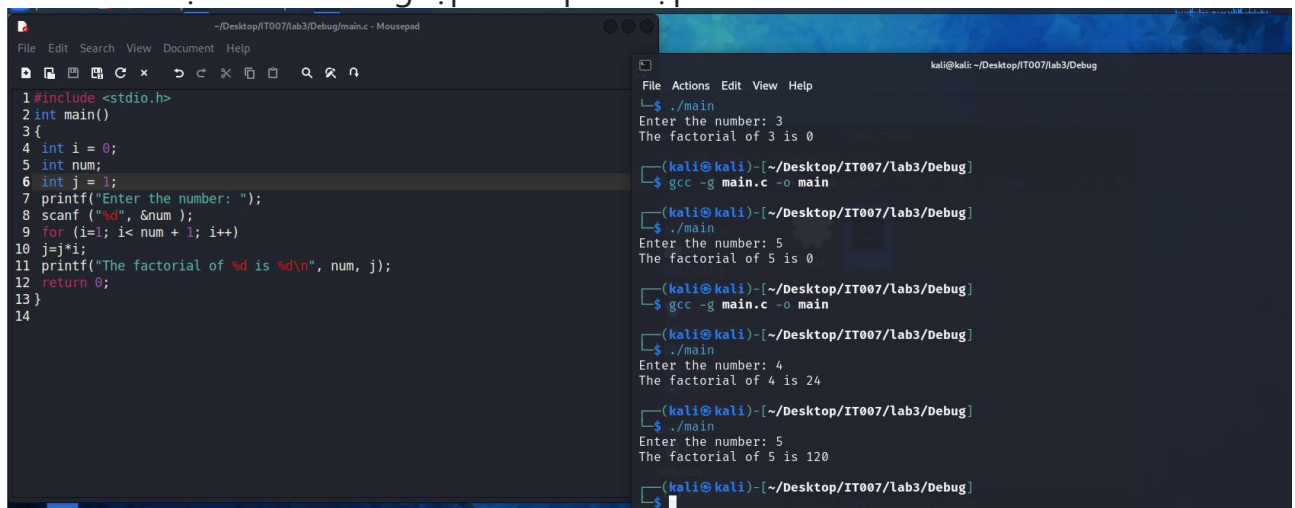
```
File Actions Edit View Help
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from main ...
(gdb) break 8
unmatched quote
(gdb) break 8
Breakpoint 1 at 0x1189: file main.c, line 8.
(gdb) run
Starting program: /home/kali/Desktop/IT007/Lab3/Debug/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Enter the number: 5

Breakpoint 1, main () at main.c:8
8         j=j*i;
(gdb) print i
$1 = 1
(gdb) print j
$2 = 0
(gdb) print num
$3 = 5
(gdb)
```

Sử dụng gdb debug:

- Gán cho j giá trị.
- Giá trị kết thúc vòng lặp chưa phù hợp



```
1#include <stdio.h>
2int main()
3{
4    int i = 0;
5    int num;
6    int j = 1;
7    printf("Enter the number: ");
8    scanf ("%d", &num );
9    for (i=1; i< num + 1; i++)
10        j=j*i;
11    printf("The factorial of %d is %d\n", num, j);
12    return 0;
13}
14
```

```
(kali@kali)~/Desktop/IT007/Lab3/Debug
$ ./main
Enter the number: 3
The factorial of 3 is 0

(kali@kali)~/Desktop/IT007/Lab3/Debug
$ gcc -g main.c -o main

(kali@kali)~/Desktop/IT007/Lab3/Debug
$ ./main
Enter the number: 5
The factorial of 5 is 0

(kali@kali)~/Desktop/IT007/Lab3/Debug
$ gcc -g main.c -o main

(kali@kali)~/Desktop/IT007/Lab3/Debug
$ ./main
Enter the number: 4
The factorial of 4 is 24

(kali@kali)~/Desktop/IT007/Lab3/Debug
$ ./main
Enter the number: 5
The factorial of 5 is 120

(kali@kali)~/Desktop/IT007/Lab3/Debug
$
```

Sử dụng lệnh top:

Lab 3: Tiến trình và tiểu trình

```
kali@kali: ~/Desktop/IT007/lab3/Debug
File Actions Edit View Help
top - 04:16:05 up 29 min, 1 user, load average: 0.14, 0.27, 0.22
Tasks: 195 total, 2 running, 193 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.8 us, 1.7 sy, 0.0 ni, 96.4 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 1956.1 total, 476.1 free, 886.2 used, 760.8 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1069.9 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
  901 root        20   0 405772 129576 56236 S   6.6   6.5   2:06.58 Xorg
15283 kali        20   0 455812 105416 86028 S   1.3   5.3   0:00.81 qterminal
  482 root        20   0 241352 12568 7424 S   0.7   0.6   0:07.56 vmttoolsd
 1212 kali        20   0 1321860 107648 78116 S   0.7   5.4   0:18.41 xfwm4
 1273 kali        20   0 304632 34648 19072 S   0.7   1.7   0:11.02 panel-13-cpugra
16908 kali        20   0 11728 5632 3456 R   0.7   0.3   0:00.05 top
1275 kali        20   0 424060 29968 20724 R   0.3   1.5   0:08.73 panel-15-genmon
 1398 kali        39  19 927348 26232 14700 S   0.3   1.3   0:00.58 tumblerd
 1435 kali        20   0 221156 47260 30028 S   0.3   2.4   0:08.50 vmttoolsd
13192 root        20   0      0      0      0 I   0.3   0.0   0:00.12 kworker/3:0-mm_percpu_wq
13193 root        20   0      0      0      0 I   0.3   0.0   0:00.38 kworker/2:1-mm_percpu_wq
    1 root        20   0 21040 12400 9328 S   0.0   0.6   0:02.59 systemd
    2 root        20   0      0      0      0 S   0.0   0.0   0:00.02 kthreadd
    3 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    5 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
    6 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 netns
    8 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   10 root        20   0      0      0      0 I   0.0   0.0   0:00.63 kworker/u64:0-writeback
   11 root         0 -20      0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
```

Tạo tiến trình:

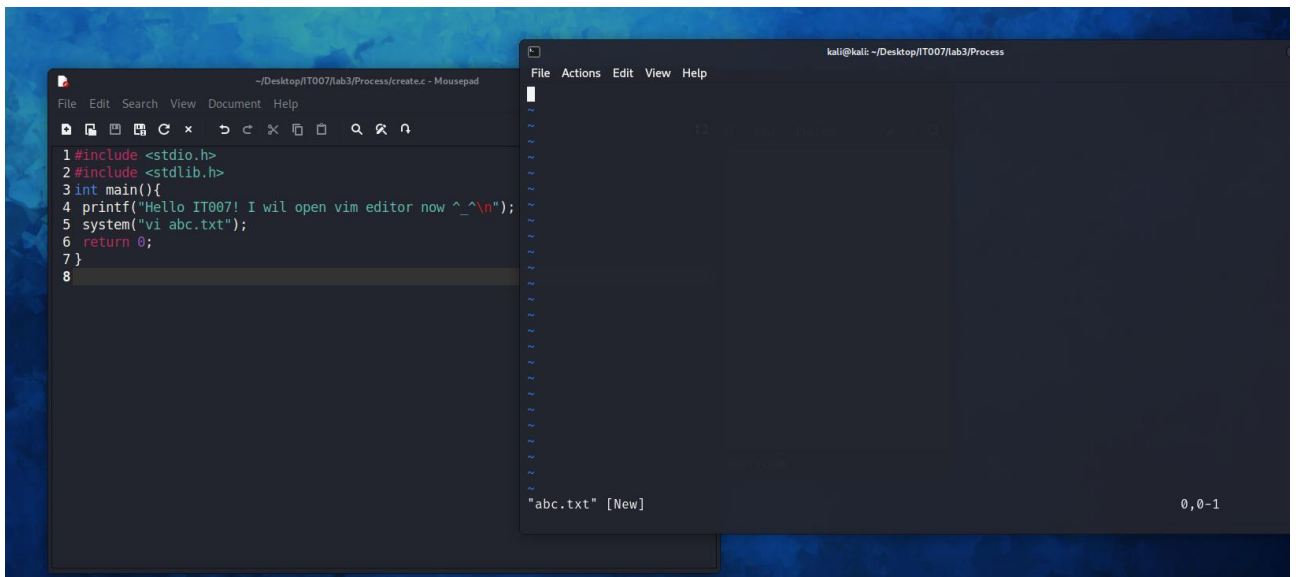
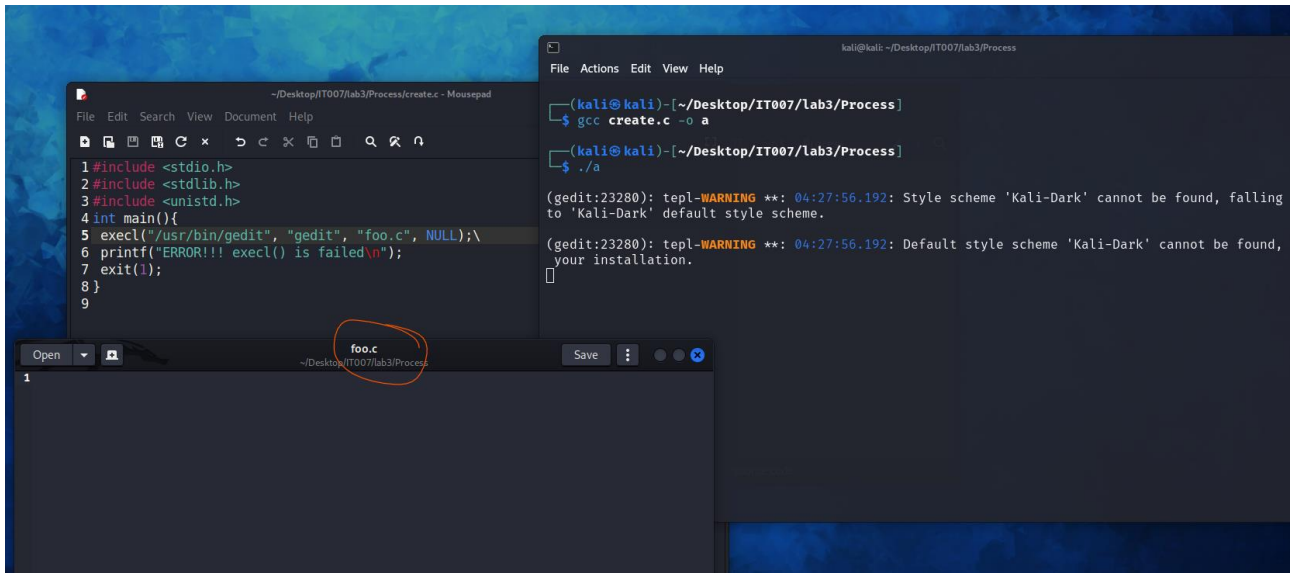
```
~/Desktop/IT007/lab3/process/create.c
File Edit Search View Document Help
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 int main(){
5     pid_t pid;
6     pid=fork();
7     if(pid==0)
8         printf("Child Process, pid=%d\n",pid);
9     else
10        printf("Parent Process, pid=%d\n",pid);
11    exit(0);
12 }
13
14

(kali@kali)~/Desktop/IT007/lab3/Process
$ gcc create.c -o a

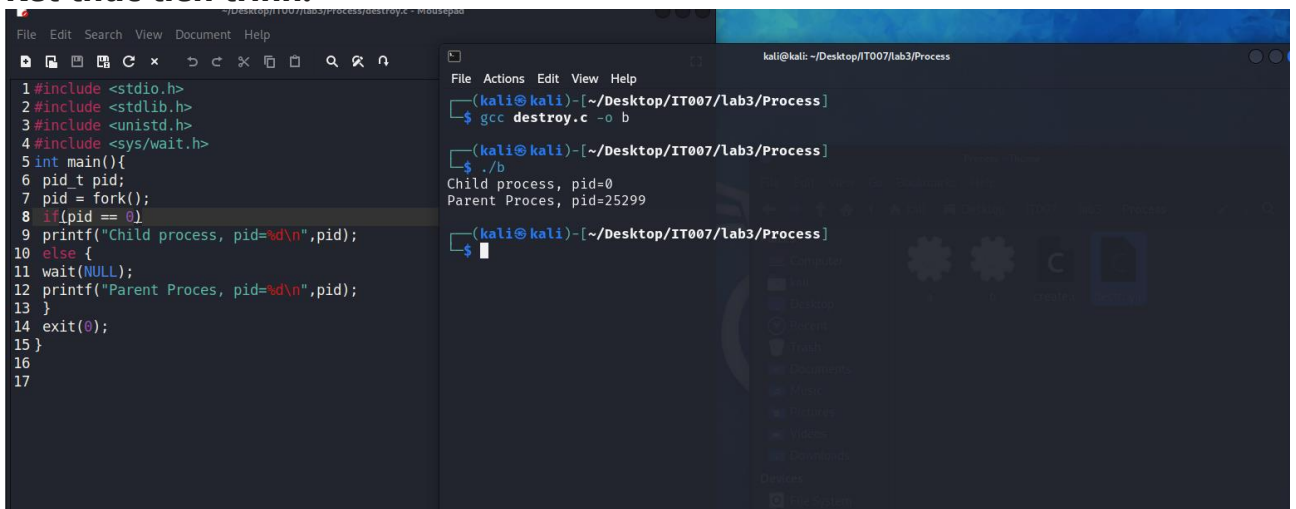
(kali@kali)~/Desktop/IT007/lab3/Process
$ ./a
Parent Process, pid=20707
Child Process, pid=0

(kali@kali)~/Desktop/IT007/lab3/Process
$
```

Lab 3: Tiến trình và tiểu trình

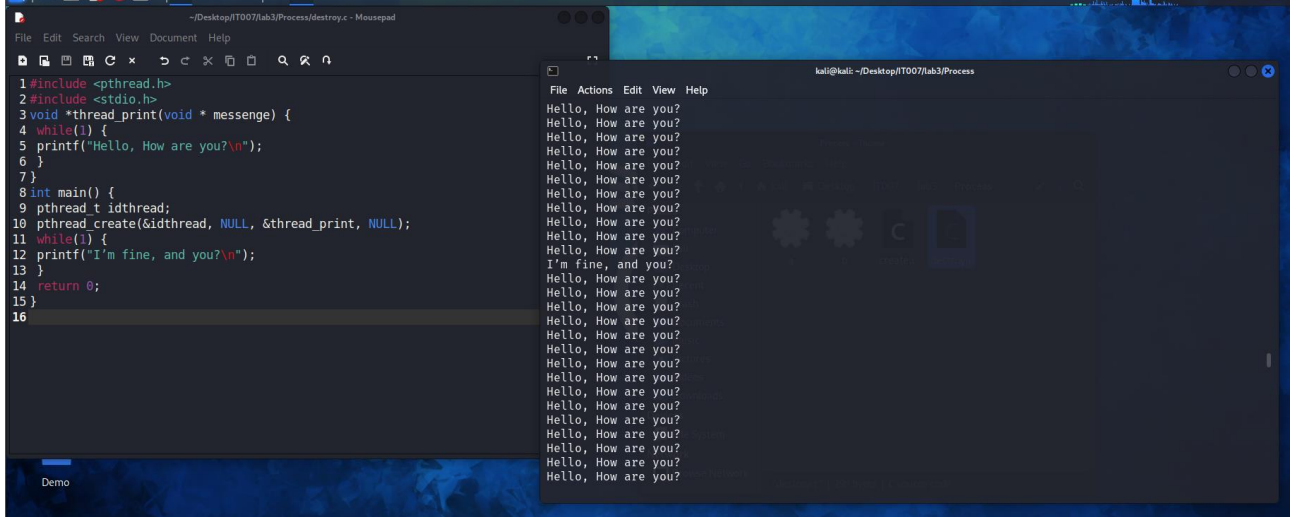


Kết thúc tiến trình:



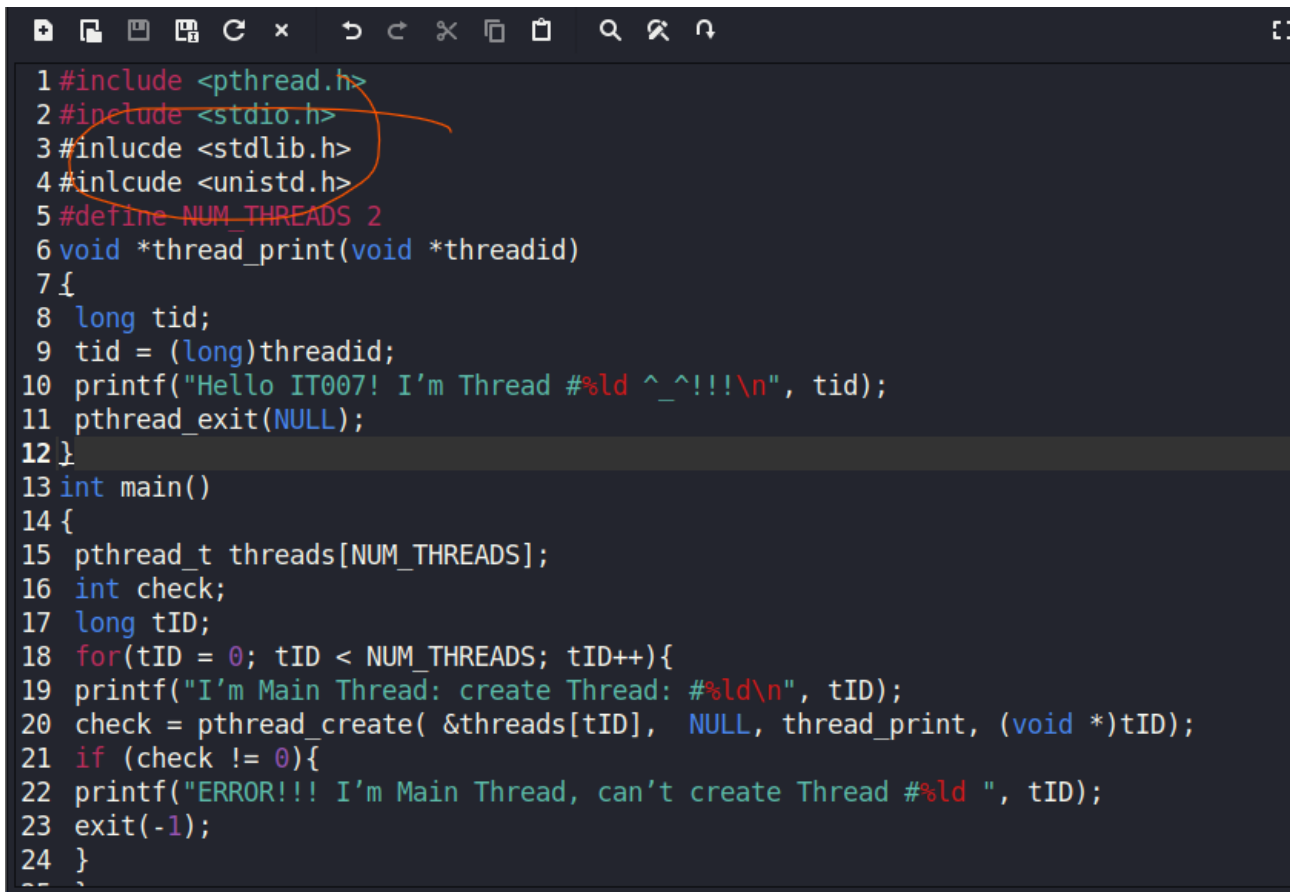
Lab 3: Tiến trình và tiểu trình

Tạo tiểu trình:

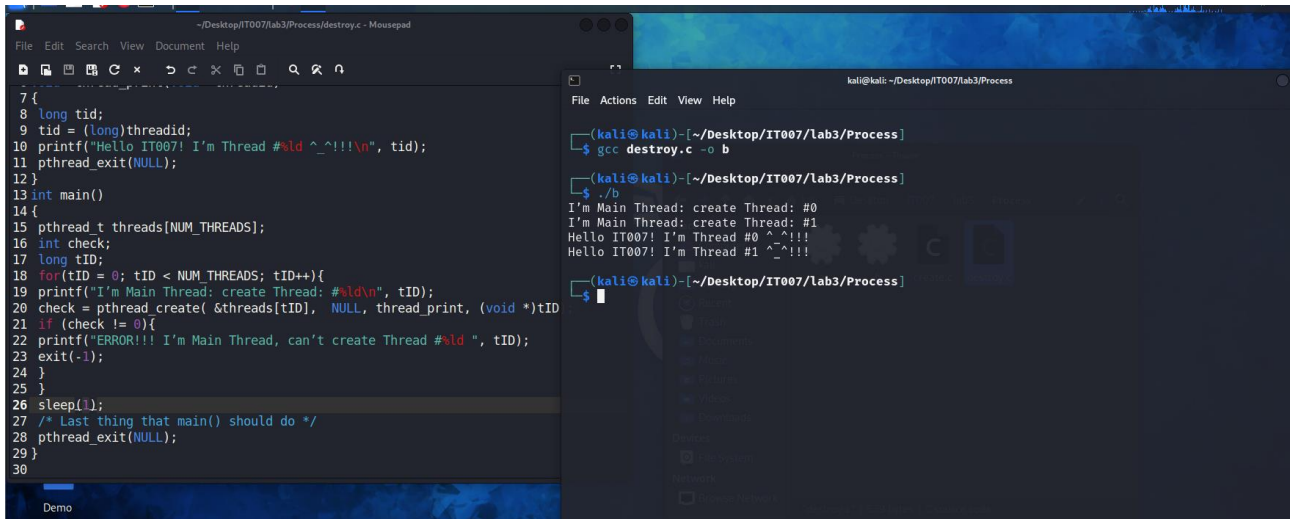


Hủy tiểu trình:

Lỗi:



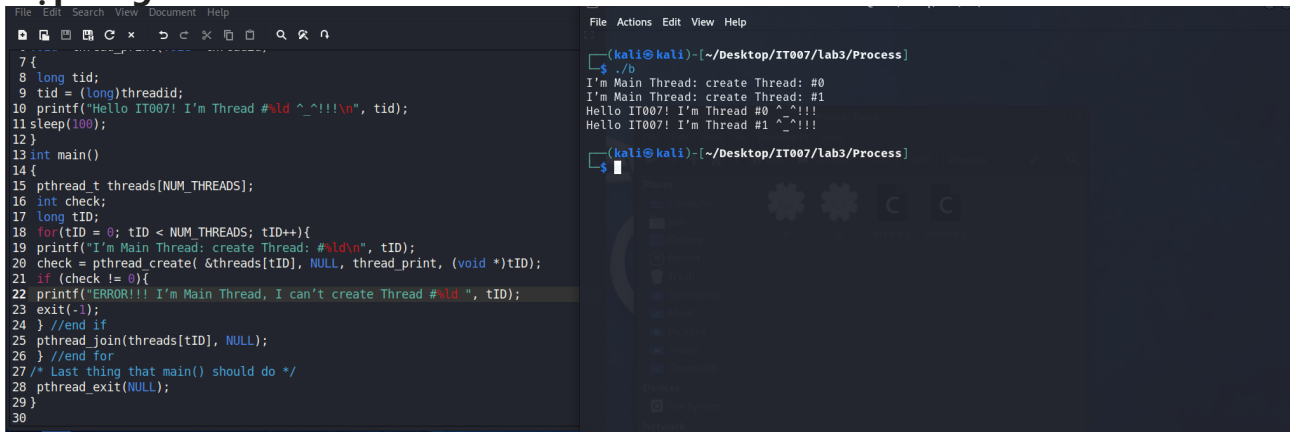
Lab 3: Tiến trình và tiểu trình



```
7 {
8   long tid;
9   tid = (long)threadid;
10  printf("Hello IT007! I'm Thread #%ld ^_^\n", tid);
11  pthread_exit(NULL);
12 }
13 int main()
14 {
15   pthread_t threads[NUM_THREADS];
16   int check;
17   long tID;
18   for(tID = 0; tID < NUM_THREADS; tID++){
19     printf("I'm Main Thread: create Thread: #%ld\n", tID);
20     check = pthread_create(&threads[tID], NULL, thread_print, (void *)tID);
21     if (check != 0){
22       printf("ERROR!!! I'm Main Thread, can't create Thread #%ld ", tID);
23       exit(-1);
24     }
25   }
26   sleep(1);
27   /* Last thing that main() should do */
28   pthread_exit(NULL);
29 }
30
```

```
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ gcc destroy.c -o b
$ ./b
I'm Main Thread: create Thread: #0
I'm Main Thread: create Thread: #1
Hello IT007! I'm Thread #0 ^_^\n
Hello IT007! I'm Thread #1 ^_^\n
$
```

Hợp và gỡ tiến trình:

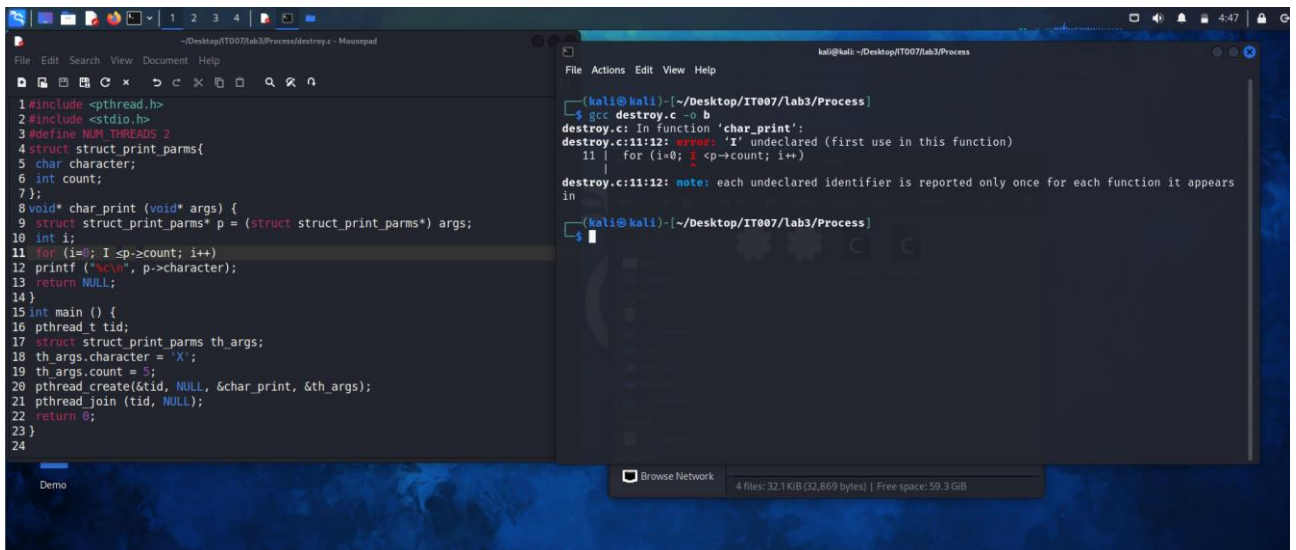


```
7 {
8   long tid;
9   tid = (long)threadid;
10  printf("Hello IT007! I'm Thread #%ld ^_^\n", tid);
11  sleep(100);
12 }
13 int main()
14 {
15   pthread_t threads[NUM_THREADS];
16   int check;
17   long tID;
18   for(tID = 0; tID < NUM_THREADS; tID++){
19     printf("I'm Main Thread: create Thread: #%ld\n", tID);
20     check = pthread_create(&threads[tID], NULL, thread_print, (void *)tID);
21     if (check != 0){
22       printf("ERROR!!! I'm Main Thread, I can't create Thread #%ld ", tID);
23       exit(-1);
24     } //end if
25     pthread_join(threads[tID], NULL);
26   } //end for
27   /* Last thing that main() should do */
28   pthread_exit(NULL);
29 }
30
```

```
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ ./b
I'm Main Thread: create Thread: #0
I'm Main Thread: create Thread: #1
Hello IT007! I'm Thread #0 ^_^\n
Hello IT007! I'm Thread #1 ^_^\n
$
```

Truyền dữ liệu cho tiến trình:

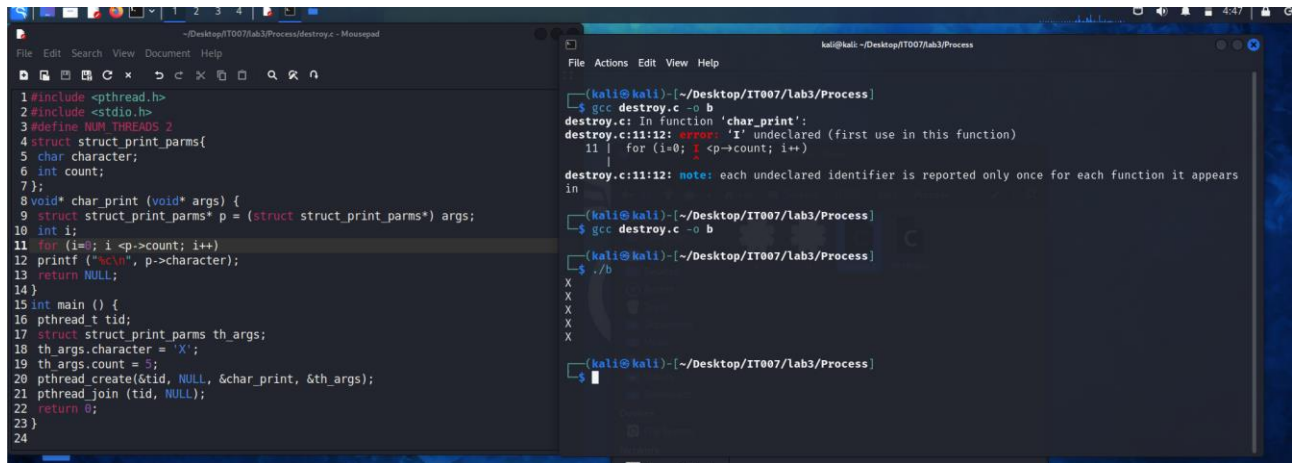
Lỗi:



```
1 #include <pthread.h>
2 #include <stdio.h>
3 #define NUM_THREADS 2
4 struct struct_print_parms{
5   char character;
6   int count;
7 };
8 void* char_print(void* args) {
9   struct struct_print_parms* p = (struct struct_print_parms*) args;
10  int i;
11  for(i=0; i < p->count; i++)
12    printf("%c\n", p->character);
13  return NULL;
14 }
15 int main() {
16   pthread_t tid;
17   struct struct_print_parms th_args;
18   th_args.character = 'X';
19   th_args.count = 5;
20   pthread_create(&tid, NULL, &char_print, &th_args);
21   pthread_join(tid, NULL);
22   return 0;
23 }
24
```

```
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ gcc destroy.c -o b
destroy.c:11:12: error: 'char_print' undeclared (first use in this function)
11 |   for (i=0; i < p->count; i++)
    |           ^
destroy.c:11:12: note: each undeclared identifier is reported only once for each function it appears in
$
```

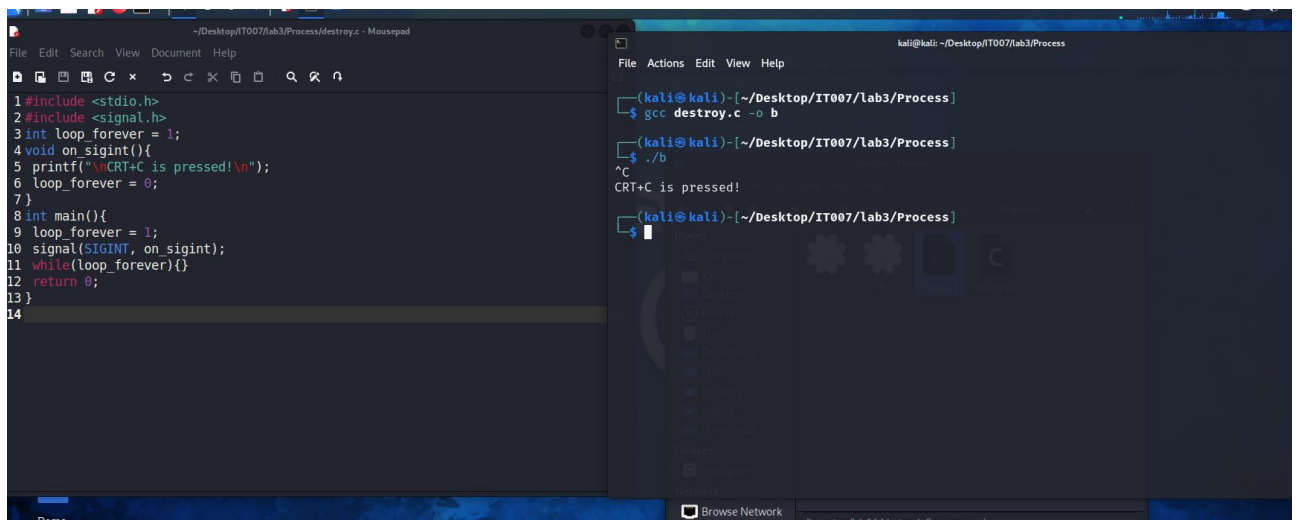
Lab 3: Tiến trình và tiểu trình



```
1 #include <pthread.h>
2 #include <stdio.h>
3 #define NUM_THREADS 2
4 struct struct_print_parms{
5     char character;
6     int count;
7 };
8 void* char_print(void* args){
9     struct struct_print_parms* p = (struct struct_print_parms*) args;
10    int i;
11    for (i=0; i < p->count; i++)
12        printf("%c\n", p->character);
13    return NULL;
14 }
15 int main() {
16     pthread_t tid;
17     struct struct_print_parms th_args;
18     th_args.character = 'X';
19     th_args.count = 5;
20     pthread_create(&tid, NULL, &char_print, &th_args);
21     pthread_join(tid, NULL);
22     return 0;
23 }
24
```

```
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ gcc destroy.c -o b
destroy.c:11:12: error: 'i' undeclared (first use in this function)
11 |     for (i=0; i < p->count; i++)
    |            ^
destroy.c:11:12: note: each undeclared identifier is reported only once for each function it appears in
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ gcc destroy.c -o b
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ ./b
X
X
X
X
X
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$
```

SIGNAL



```
1 #include <stdio.h>
2 #include <signal.h>
3 int loop_forever = 1;
4 void on_sigint(){
5     printf("\nCRT+C is pressed!\n");
6     loop_forever = 0;
7 }
8 int main(){
9     loop_forever = 1;
10    signal(SIGINT, on_sigint);
11    while(loop_forever){}
12    return 0;
13 }
14
```

```
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ gcc destroy.c -o b
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$ ./b
^C
CRT+C is pressed!
(kali@kali)-[~/Desktop/IT007/lab3/Process]
$
```

BÀI TẬP

Câu 1:

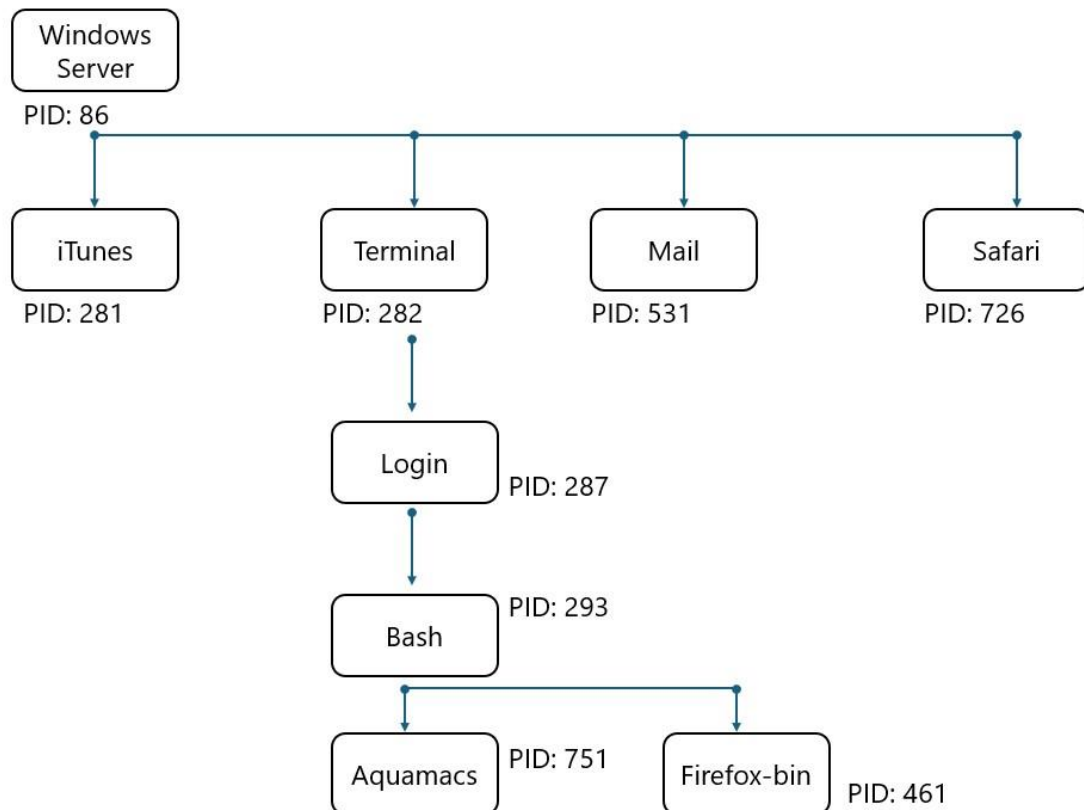
a)

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

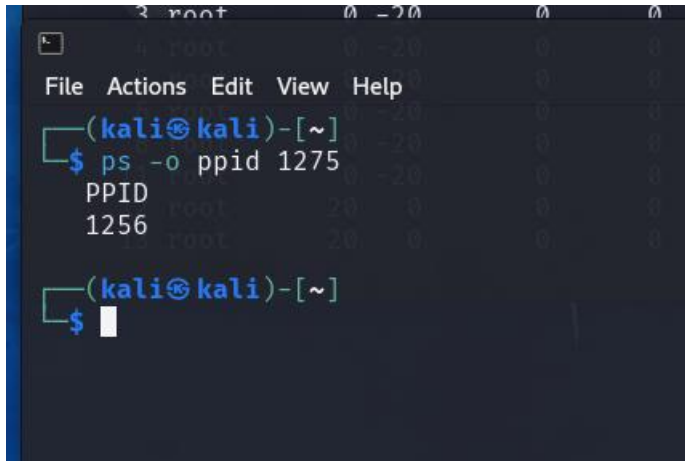
Lời giải:



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

Lời giải:

- Để tìm tiến trình cha của một tiến trình sử dụng lệnh `ps` dựa vào PID; ta có thể dùng lệnh: **`ps -o ppid <PID>`**
- Minh hoạ:



```
(kali㉿kali)-[~]
$ ps -o ppid 1275
PPID
1256
(kali㉿kali)-[~]
$
```

- 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.

Lời giải:

- Cài đặt lệnh `pstree` trong Ubuntu:
 - o **`sudo apt-get update`**
 - o **`sudo apt-get install pstree`**
- Cách sử dụng `pstree` để tìm tiến trình cha dựa vào PID của nó:
 - o **`pstree -p <PID>`**
- Minh hoạ:

```

(kali@kali)-[~]
$ ps -o ppid 1275
PPID
1256

(kali@kali)-[~]
$ pstree -p 1275
panel-15-genmon(1275)---{panel-15-genmon}(1295)
                        |
                        |---{panel-15-genmon}(1296)
                        |
                        |---{panel-15-genmon}(1300)

(kali@kali)-[~]
$
    
```

Câu 2:

Chương trình sau in ra gì?

```
#include<stdio.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);
```

```
}
```


Kết quả thực hiện:

```

ducktab@ducktan: ~/Desktop/IT007/Lab3/Pratice/Bai2
Bai2.c:19:1: warning: incompatible implicit declaration of built-in function 'exit' [-Wbuiltin-declaration-mismatch]
 19 | exit(0);
    | ^~~~~
Bai2.c:19:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
ducktab@ducktan:~/Desktop/IT007/Lab3/Pratice/Bai2$ gcc Bai2.c -o Bai2
Bai2.c: In function 'main':
Bai2.c:12:7: warning: implicit declaration of function 'fork' [-Wimplicit-function-declaration]
 12 | pid = fork();
    | ^~~~~
Bai2.c:15:1: warning: implicit declaration of function 'exit' [-Wimplicit-function-declaration]
 15 | exit(0);
    | ^~~~~
Bai2.c:8:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
  7 | #include<stdio.h>
    | +++ |+#include <stdlib.h>
  8 | #define pid_t int
Bai2.c:15:1: warning: incompatible implicit declaration of built-in function 'exit' [-Wbuiltin-declaration-mismatch]
 15 | exit(0);
    | ^~~~~
Bai2.c:15:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
Bai2.c:17:1: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
 17 | wait(NULL); /*wait until the child terminates */
    | ^~~~~
Bai2.c:20:1: warning: incompatible implicit declaration of built-in function 'exit' [-Wbuiltin-declaration-mismatch]
 20 | exit(0);
    | ^~~~~
Bai2.c:20:1: note: include '<stdlib.h>' or provide a declaration of 'exit'
ducktab@ducktan:~/Desktop/IT007/Lab3/Pratice/Bai2$ gcc Bai2.c -o Bai2
ducktab@ducktan:~/Desktop/IT007/Lab3/Pratice/Bai2$ ./Bai2
I see 17 coconuts!
ducktab@ducktan:~/Desktop/IT007/Lab3/Pratice/Bai2$

```

Giải thích: Khi mới biên dịch, chương trình có lỗi không thể thực thi:

- Chưa khai báo thư viện
- ⇒ Sau khi khai báo thư viện: kết quả: I see 17 coconuts!

Minh chứng:

```

1 #include<stdio.h>
2
3 #include <stdlib.h>
4 #include <unistd.h>
5
6 int main(){
7     pid_t pid;
8     int num_coconuts = 17;
9     pid = fork();
10    if(pid == 0) {
11        num_coconuts = 42;
12        exit(0);
13    } else {
14        wait(NULL);
15    }
16    printf("I see %d coconuts!\n", num_coconuts);
17    exit(0);
18 }
19
14 |
15 |
16 |
17 |
18 |
19 |
20 |
21 |
22 |
23 |
24 |
25 |
26 |
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 |
54 |
55 |
56 |
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 |
101 |
102 |
103 |
104 |
105 |
106 |
107 |
108 |
109 |
110 |
111 |
112 |
113 |
114 |
115 |
116 |
117 |
118 |
119 |
120 |
121 |
122 |
123 |
124 |
125 |
126 |
127 |
128 |
129 |
130 |
131 |
132 |
133 |
134 |
135 |
136 |
137 |
138 |
139 |
140 |
141 |
142 |
143 |
144 |
145 |
146 |
147 |
148 |
149 |
150 |
151 |
152 |
153 |
154 |
155 |
156 |
157 |
158 |
159 |
160 |
161 |
162 |
163 |
164 |
165 |
166 |
167 |
168 |
169 |
170 |
171 |
172 |
173 |
174 |
175 |
176 |
177 |
178 |
179 |
180 |
181 |
182 |
183 |
184 |
185 |
186 |
187 |
188 |
189 |
190 |
191 |
192 |
193 |
194 |
195 |
196 |
197 |
198 |
199 |
200 |
201 |
202 |
203 |
204 |
205 |
206 |
207 |
208 |
209 |
210 |
211 |
212 |
213 |
214 |
215 |
216 |
217 |
218 |
219 |
220 |
221 |
222 |
223 |
224 |
225 |
226 |
227 |
228 |
229 |
230 |
231 |
232 |
233 |
234 |
235 |
236 |
237 |
238 |
239 |
240 |
241 |
242 |
243 |
244 |
245 |
246 |
247 |
248 |
249 |
250 |
251 |
252 |
253 |
254 |
255 |
256 |
257 |
258 |
259 |
260 |
261 |
262 |
263 |
264 |
265 |
266 |
267 |
268 |
269 |
270 |
271 |
272 |
273 |
274 |
275 |
276 |
277 |
278 |
279 |
280 |
281 |
282 |
283 |
284 |
285 |
286 |
287 |
288 |
289 |
290 |
291 |
292 |
293 |
294 |
295 |
296 |
297 |
298 |
299 |
300 |
301 |
302 |
303 |
304 |
305 |
306 |
307 |
308 |
309 |
310 |
311 |
312 |
313 |
314 |
315 |
316 |
317 |
318 |
319 |
320 |
321 |
322 |
323 |
324 |
325 |
326 |
327 |
328 |
329 |
330 |
331 |
332 |
333 |
334 |
335 |
336 |
337 |
338 |
339 |
340 |
341 |
342 |
343 |
344 |
345 |
346 |
347 |
348 |
349 |
350 |
351 |
352 |
353 |
354 |
355 |
356 |
357 |
358 |
359 |
360 |
361 |
362 |
363 |
364 |
365 |
366 |
367 |
368 |
369 |
370 |
371 |
372 |
373 |
374 |
375 |
376 |
377 |
378 |
379 |
380 |
381 |
382 |
383 |
384 |
385 |
386 |
387 |
388 |
389 |
390 |
391 |
392 |
393 |
394 |
395 |
396 |
397 |
398 |
399 |
400 |
401 |
402 |
403 |
404 |
405 |
406 |
407 |
408 |
409 |
410 |
411 |
412 |
413 |
414 |
415 |
416 |
417 |
418 |
419 |
420 |
421 |
422 |
423 |
424 |
425 |
426 |
427 |
428 |
429 |
430 |
431 |
432 |
433 |
434 |
435 |
436 |
437 |
438 |
439 |
440 |
441 |
442 |
443 |
444 |
445 |
446 |
447 |
448 |
449 |
450 |
451 |
452 |
453 |
454 |
455 |
456 |
457 |
458 |
459 |
460 |
461 |
462 |
463 |
464 |
465 |
466 |
467 |
468 |
469 |
470 |
471 |
472 |
473 |
474 |
475 |
476 |
477 |
478 |
479 |
480 |
481 |
482 |
483 |
484 |
485 |
486 |
487 |
488 |
489 |
490 |
491 |
492 |
493 |
494 |
495 |
496 |
497 |
498 |
499 |
500 |
501 |
502 |
503 |
504 |
505 |
506 |
507 |
508 |
509 |
510 |
511 |
512 |
513 |
514 |
515 |
516 |
517 |
518 |
519 |
520 |
521 |
522 |
523 |
524 |
525 |
526 |
527 |
528 |
529 |
530 |
531 |
532 |
533 |
534 |
535 |
536 |
537 |
538 |
539 |
540 |
541 |
542 |
543 |
544 |
545 |
546 |
547 |
548 |
549 |
550 |
551 |
552 |
553 |
554 |
555 |
556 |
557 |
558 |
559 |
560 |
561 |
562 |
563 |
564 |
565 |
566 |
567 |
568 |
569 |
570 |
571 |
572 |
573 |
574 |
575 |
576 |
577 |
578 |
579 |
580 |
581 |
582 |
583 |
584 |
585 |
586 |
587 |
588 |
589 |
590 |
591 |
592 |
593 |
594 |
595 |
596 |
597 |
598 |
599 |
600 |
601 |
602 |
603 |
604 |
605 |
606 |
607 |
608 |
609 |
610 |
611 |
612 |
613 |
614 |
615 |
616 |
617 |
618 |
619 |
620 |
621 |
622 |
623 |
624 |
625 |
626 |
627 |
628 |
629 |
630 |
631 |
632 |
633 |
634 |
635 |
636 |
637 |
638 |
639 |
640 |
641 |
642 |
643 |
644 |
645 |
646 |
647 |
648 |
649 |
650 |
651 |
652 |
653 |
654 |
655 |
656 |
657 |
658 |
659 |
660 |
661 |
662 |
663 |
664 |
665 |
666 |
667 |
668 |
669 |
670 |
671 |
672 |
673 |
674 |
675 |
676 |
677 |
678 |
679 |
680 |
681 |
682 |
683 |
684 |
685 |
686 |
687 |
688 |
689 |
690 |
691 |
692 |
693 |
694 |
695 |
696 |
697 |
698 |
699 |
700 |
701 |
702 |
703 |
704 |
705 |
706 |
707 |
708 |
709 |
710 |
711 |
712 |
713 |
714 |
715 |
716 |
717 |
718 |
719 |
720 |
721 |
722 |
723 |
724 |
725 |
726 |
727 |
728 |
729 |
730 |
731 |
732 |
733 |
734 |
735 |
736 |
737 |
738 |
739 |
740 |
741 |
742 |
743 |
744 |
745 |
746 |
747 |
748 |
749 |
750 |
751 |
752 |
753 |
754 |
755 |
756 |
757 |
758 |
759 |
760 |
761 |
762 |
763 |
764 |
765 |
766 |
767 |
768 |
769 |
770 |
771 |
772 |
773 |
774 |
775 |
776 |
777 |
778 |
779 |
780 |
781 |
782 |
783 |
784 |
785 |
786 |
787 |
788 |
789 |
790 |
791 |
792 |
793 |
794 |
795 |
796 |
797 |
798 |
799 |
800 |
801 |
802 |
803 |
804 |
805 |
806 |
807 |
808 |
809 |
810 |
811 |
812 |
813 |
814 |
815 |
816 |
817 |
818 |
819 |
820 |
821 |
822 |
823 |
824 |
825 |
826 |
827 |
828 |
829 |
830 |
831 |
832 |
833 |
834 |
835 |
836 |
837 |
838 |
839 |
840 |
841 |
842 |
843 |
844 |
845 |
846 |
847 |
848 |
849 |
850 |
851 |
852 |
853 |
854 |
855 |
856 |
857 |
858 |
859 |
860 |
861 |
862 |
863 |
864 |
865 |
866 |
867 |
868 |
869 |
870 |
871 |
872 |
873 |
874 |
875 |
876 |
877 |
878 |
879 |
880 |
881 |
882 |
883 |
884 |
885 |
886 |
887 |
888 |
889 |
890 |
891 |
892 |
893 |
894 |
895 |
896 |
897 |
898 |
899 |
900 |
901 |
902 |
903 |
904 |
905 |
906 |
907 |
908 |
909 |
910 |
911 |
912 |
913 |
914 |
915 |
916 |
917 |
918 |
919 |
920 |
921 |
922 |
923 |
924 |
925 |
926 |
927 |
928 |
929 |
930 |
931 |
932 |
933 |
934 |
935 |
936 |
937 |
938 |
939 |
940 |
941 |
942 |
943 |
944 |
945 |
946 |
947 |
948 |
949 |
950 |
951 |
952 |
953 |
954 |
955 |
956 |
957 |
958 |
959 |
960 |
961 |
962 |
963 |
964 |
965 |
966 |
967 |
968 |
969 |
970 |
971 |
972 |
973 |
974 |
975 |
976 |
977 |
978 |
979 |
980 |
981 |
982 |
983 |
984 |
985 |
986 |
987 |
988 |
989 |
990 |
991 |
992 |
993 |
994 |
995 |
996 |
997 |
998 |
999 |
1000 |

```

Bài 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_*)

Hàm	Chức năng
Pthread_attr_init	Khởi tạo giá trị mặc định cho đối tượng thuộc tính
Pthread_attr_destroy	Xóa bộ nhớ được cấp phát trong quá trình khởi tạo
Pthread_attr_getschedparam	Trả về các tham số lịch trình (Scheduling Parameter) được xác định bởi pthread_attr_setschedparam ()
Pthread_attr_getschedpolicy	Đề xuất scheduling policy của thread
Pthread_attr_getdetachstate	Lấy truy xuất trạng thái khởi tạo của thread, có thể thể là riêng lẻ hoặc kết hợp
Pthread_attr_getinheritsched	Trả về chính sách lịch trình (scheduling policy) được set bởi pthread_attr_setinheritsched ().
Pthread_attr_getscope	Truy xuất phạm vi của thread
Pthread_attr_setdetachstate	Sử dụng lại ID và tài nguyên của thread khi nó bị ngắt mà không phải chờ nếu thread có thuộc tính riêng lẻ.
Pthread_attr_setguardsize	Set kích thước của khu vực an toàn của thread
Pthread_attr_setstackaddr	Set địa chỉ stack của thread
Pthread_attr_setstacksize	Set kích thước stack của thread
Pthread_attr_getguardsize	Lấy kích thước của khu vực an toàn của thread
Pthread_attr_getstackaddr	Trả về địa chỉ stack của thread được set bởi pthread_attr_setstackaddr ()

Pthread_attr_getstacksize

Trả về kích thước stack của thread được set bởi pthread_attr_setstacksize ().

```
// Pthread_attr_init
int pthread_attr_init(pthread_attr_t *tattr);
#include <pthread.h>
pthread_attr_t tattr;
int ret;

/* khởi tạo giá trị mặc định cho thuộc tính */
ret = pthread_attr_init(&tattr);

Pthread_attr_destroy

int pthread_attr_destroy(pthread_attr_t *tattr);
#include <pthread.h>

pthread_attr_t tattr;
int ret;

/* Xóa thuộc tính */
ret = pthread_attr_destroy(&tattr);

Pthread_attr_getschedparam
int pthread_attr_getschedparam(pthread_attr_t *tattr,
    const struct sched_param *param);
#include <pthread.h>

pthread_attr_t attr;
struct sched_param param;
int ret;

/* Lấy tham số lịch trình đang tồn tại trong thread */
ret = pthread_attr_getschedparam (&tattr, &param);

Pthread_attr_getschedpolicy

int pthread_attr_getschedpolicy(pthread_attr_t *tattr, int *policy);
#include <pthread.h>

pthread_attr_t tattr;
int policy;
int ret;

/* Lấy chính sách lịch trình của thread */
ret = pthread_attr_getschedpolicy (&tattr, &policy);

Pthread_attr_getdetachstate
```

```
int pthread_attr_getdetachstate(const pthread_attr_t *tattr,
    int *detachstate;
#include <pthread.h>
```

```
pthread_attr_t tattr;
int detachstate;
int ret;
```

```
/* Lấy trạng thái khởi tạo của thread */
ret = pthread_attr_getdetachstate (&tattr, &detachstate);
```

Pthread_attr_getinheritsched

```
int pthread_attr_getinheritsched(pthread_attr_t *tattr, int *inherit);
#include <pthread.h>
```

```
pthread_attr_t tattr;
int inherit;
int ret;
```

```
/* Lấy chính sách lịch trình và độ ưu tiên của thread được khởi tạo */
ret = pthread_attr_getinheritsched (&tattr, &inherit);
```

Pthread_attr_getscope

```
int pthread_attr_getscope(pthread_attr_t *tattr, int *scope);
#include <pthread.h>
```

```
pthread_attr_t tattr;
int scope;
int ret;
```

```
/* Lấy phạm vi của thread */
ret = pthread_attr_getscope(&tattr, &scope);
```

Pthread_attr_setdetachstate

```
int pthread_attr_setdetachstate(pthread_attr_t *tattr,int detachstate);
#include <pthread.h>
```

```
pthread_attr_t tattr;
int ret;
```

```
/* Set trạng thái riêng lẻ cho thread */
ret = pthread_attr_setdetachstate(&tattr,PTHREAD_CREATE_DETACHED);
```

Pthread_attr_setguardsize

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

```
int pthread_attr_setguardsize(pthread_attr_t *attr, size_t guardsize);
```

Pthread_attr_setstackaddr

```
int pthread_attr_setstackaddr(pthread_attr_t *tattr,void *stackaddr);
#include <pthread.h>
```

```
pthread_attr_t tattr;
void *base;
int ret;
```

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

```
/* Đặt địa chỉ mới cho thread */
ret = pthread_attr_setstackaddr(&tattr, base);
```

Pthread_attr_setstacksize

```
int pthread_attr_setstacksize(pthread_attr_t *tattr, int size);
#include <pthread.h>
```

```
pthread_attr_t tattr;
int size;
int ret;
```

```
size = (PTHREAD_STACK_MIN + 0x4000);
```

```
/* Đặt kích thước mới */
ret = pthread_attr_setstacksize(&tattr, size);
```

Pthread_attr_getguardsize

```
#include <pthread.h>
int pthread_attr_getguardsize(const pthread_attr_t *attr,
size_t *guardsize);
```

Pthread_attr_getstackaddr

```
int pthread_attr_getstackaddr(pthread_attr_t *tattr,void **stackaddr);
#include <pthread.h>
```

```
pthread_attr_t tattr;
void *base;
int ret;
```

```
/* Lấy một địa chỉ mới */
ret = pthread_attr_getstackaddr (&tattr, *base);
```

Pthread_attr_getstacksize

```
int pthread_attr_getstacksize(pthread_attr_t *tattr, size_t *size);
#include <pthread.h>
```

```
pthread_attr_t  tattr;
int size;
int ret;

/* Lấy kích thước stack */
ret = pthread_attr_getstacksize(&tattr, &size);
```

Bài 4:

Note: vim bị lỗi (không thể đóng khi ấn ctrl C – thay thế bằng gedit)

Source code:

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

pid_t vim_pid;

void sig(int signum) {
    kill(vim_pid, SIGINT);
    printf("\nYou are pressed Ctrl + C. Goodbye!\n");
    exit(0);
}

int main() {
    printf("Welcome to IT007, I am 22521303!\n");

    vim_pid = fork();
    if (vim_pid == 0) {
        // Chạy vim trong terminal hiện tại
        execlp("gedit", "gedit", "abcd.txt", NULL);
    } else {
        // Lắng nghe tín hiệu SIGINT
        signal(SIGINT, sig);

        // Đợi khi vim kết thúc
        waitpid(vim_pid, NULL, 0);
    }

    return 0;
}
```

Kết quả thực hiện:

Lab 3: Tiến trình và tiểu trình

