

## BÀI THỰC HÀNH LAB 03

Họ và tên: **Nguyễn Đỗ Quang**  
MSSV: **20520720**

### Phần 3.4

## 1. example\_thread\_creation.c

[illegible]

## 2. example\_thread\_selfexit.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5
6 #define NUM_THREADS 2
7 void *thread_print(void *threadid)
8 {
9     long tid;
10    tid = (long)threadid;
11    printf("Hello IT007! I'm Thread #%ld ^_!!!\n", tid);
12    pthread_exit(NULL);
13 }
14 int main()
15 {
16     pthread_t threads[NUM_THREADS];
17     int check;
18     long tid;
19     for (tid = 0; tid < NUM_THREADS; tid++)
20     {
21         printf("I'm Main Thread: create Thread: #%ld\n", tid);
22         check = pthread_create(
23             &threads[tid],
24             NULL,
25             thread_print,
26             (void *)tid);
27         if (check != 0)
28         {
29             printf("ERROR!!! I'm Main Thread, can't create Thread #%ld", tid);
30             exit(-1);
31         }
32     }
33     sleep(100);
34     /* Last thing that main() should do */
35     pthread_exit(NULL);
36 }
```

### 3. example\_thread\_mainexit.c

```
403 > C example_thread_mainexit.c > NUM_THREADS

1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #define NUM_THREADS 2
6 void *thread_print(void *threadid)
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     {
20         printf("I'm Main Thread: create Thread: #%ld\n", tID);
21         check = pthread_create(
22             &threads[tID],
23             NULL,
24             thread_print,
25             (void *)tID);
26         if (check != 0)
27         {
28             printf("ERROR!!! I'm Main Thread, I can't create Thread%ld ",
29                 tID);
30             exit(-1);
31         }
32     }
33     /* Last thing that main() should do */
34     pthread_exit(NULL);
35 }
```

```
quang@quang-FX503VM: ~/Code/IT007/TH03
File Edit View Search Terminal Help
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_thread_mainexit
I'm Main Thread: create Thread: #0
I'm Main Thread: create Thread: #1
Hello IT007! I'm Thread #1 ^_?!!!
Hello IT007! I'm Thread #0 ^_?!!!

quang@quang-FX503VM: ~
File Edit View Search Terminal Help
top - 09:43:34 up 1:24, 1 user, load average: 0.59, 0.74, 0.79
Tasks: 305 total, 1 running, 302 sleeping, 0 stopped, 2 zombie
%Cpu(s): 10.0 us, 0.5 sy, 0.0 ni, 82.1 id, 0.0 wa, 0.0 hi, 1.3 si, 0.0 st
MiB Mem : 7913.3 total, 201.7 free, 3795.6 used, 3916.0 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used, 3600.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 3451 quang    20   0  54.5g 308268 127604 S   17.6   3.8   2:41.15 code
 1379 quang    20   0 4233996 251244 116664 S    8.6   3.1   2:07.20 cinnamon
  927 root       20   0   24.4g 171132  96268 S    7.6   2.1   1:41.87 Xorg
 4138 quang    20   0 1477556 184392 137064 S    5.6   2.3   3:04.38 wpsoffi+
 3427 quang    20   0   32.4g 195412  96964 S    5.0   2.4   1:03.06 code
 9593 quang    20   0 1024832 49744  35480 S    5.0   0.6   0:00.54 gnome-s+
  945 root       20   0      0      0      0 S    3.0   0.0   0:51.01 nv.queue
 3391 quang    20   0   36.6g 160868 116880 S    3.0   2.0   0:28.89 code
 4249 quang    20   0 1195480 438520 260276 S    2.7   5.4   3:25.24 wps
  943 root      -51   0      0      0      0 S    1.7   0.0   0:43.26 irq/133+
 9629 quang    20   0  1129.1g 96876  78356 S    1.0   1.2   0:00.08 chrome
 3347 quang    20   0 469832  43408  32240 S    0.7   0.5   8:24.15 gnome-t+
 9591 quang    20   0 15212  4224  3464 R    0.7   0.1   0:00.17 top
    1 root       20   0 168220  11484  8444 S    0.3   0.1   0:01.03 systemd
    30 root       20   0      0      0      0 S    0.3   0.0   0:00.06 ksoftirq
   306 root       20   0 24552  7568  4052 S    0.3   0.1   0:01.01 systemd+
   733 root       20   0 814704 20912 17248 S    0.3   0.3   0:05.03 Network+
```

#### 4. example\_thread\_join.c

```
1 #include <pthread.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #define NUM_THREADS 2
6 void *thread_print(void *threadid)
7 {
8     long tid;
9     tid = (long)threadid;
10    printf("Hello IT007! I'm Thread #%ld ^_?!!!\n", tid);
11    sleep(10);
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     {
20         printf("I'm Main Thread: create Thread: #%ld\n", tID);
21         check = pthread_create(
22             &threads[tID],
23             NULL,
24             thread_print,
25             (void *)tID);
26         if (check != 0)
27         {
28             printf("ERROR!!! I'm Main Thread, I can't create Thread %ld ",
29                 tID);
30             exit(-1);
31         }
32         pthread_join(threads[tID], NULL);
33     }
34     /* Last thing that main() should do */
35     pthread_exit(NULL);
36 }
```

```
quang@quang-FX503VM: ~/Code/IT007/TH03
File Edit View Search Terminal Help
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ gcc -o example_thread_join example_thread_join.c -lpthread
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_thread_join
I'm Main Thread: create Thread: #0
Hello IT007! I'm Thread #0 ^_?!!!
I'm Main Thread: create Thread: #1
Hello IT007! I'm Thread #1 ^_?!!!
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ |
```

#### 5. example\_thread\_structure

```
403 > C example_thread_structure.c > main()

1 #include <pthread.h>
2 #include <stdio.h>
3 #define NUM_THREADS 2
4 struct struct_print_parms
5 {
6     char character;
7     int count;
8 };
9 void *char_print(void *args)
10 {
11     struct struct_print_parms *p = (struct struct_print_parms *)args;
12     int i;
13     for (i = 0; i < p->count; i++)
14         printf("%c\n", p->character);
15     return NULL;
16 }
17 int main()
18 {
19     pthread_t tid;
20     struct struct_print_parms th_args;
21     th_args.character = 'X';
22     th_args.count = 5;
23     pthread_create(&tid, NULL, &char_print, &th_args);
24     pthread_join(tid, NULL);
25     return 0;
26 }
```

```
quang@quang-FX503VM: ~/Code/IT007/TH03
File Edit View Search Terminal Help
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ gcc -o example_thread_structure example_thread_structure.c -lpthread
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_thread_structure
X
X
X
X
X
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ |
```

## 6. example signal

```
1 #include <stdio.h>
2 #include <signal.h>
3 int loop_forever = 1;
4 void on_sigint()
5 {
6     printf("\nCRT+C is pressed!\n");
7     loop_forever = 0;
8 }
9 int main()
10 {
11     loop_forever = 1;
12     signal(SIGINT, on_sigint);
13     while (loop_forever);
14     return 0;
15 }
```

```
quang@quang-FX503VM: ~/Code/IT007/TH03
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ gcc -o example_thread_structure example_thread_structure.c -lpthread
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_thread_structure.c
bash: ./example_thread_structure.c: Permission denied
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_thread_structure
X
X
X
X
X
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ touch example_signal.c
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ gcc -o example_signal example_signal.c -lpthread
(base) quang@quang-FX503VM:~/Code/IT007/TH03$ ./example_signal
^C
CRT+C is pressed!
(base) quang@quang-FX503VM:~/Code/IT007/TH03$
```

### Phần 3.5

#### Câu 1:

a. Về cây quan hệ parent-child trong đó

**PPID:** ID của tiến trình cha

**PID:** ID của tiến trình con

```
(PPID = 1)          (PPID = 86)          (PPID = 282)    (PPID = 287)    (PPID = 293)
WindowServer(86)  -----> iTunes(281)
                  -----> Terminal(282) -----> login(287) -----> bash(293)-----> firefox-bin(461)
                  -----> Safari(531)
                  -----> Mail(726)
                                     |
                                     -----> Aquamacs(751)
```

b. Cách dùng lệnh ps để tìm tiến trình cha:

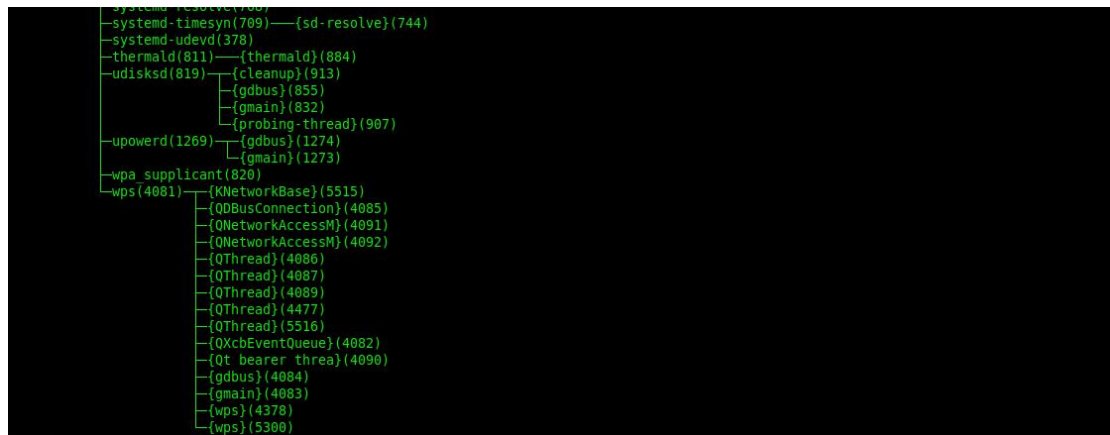
Chúng ta sử dụng lệnh ps -f [PID của tiến trình con] để hiển thị PPID của tiến trình cha.

Sử dụng tiếp lệnh ps -f [PPID của tiến trình cha] để hiển thị tiến trình PPID của tiến trình cha tiếp theo...

```
2181 pts/0    Ss      0:00 bash
(base) quang@quang-FX503VM:~$ ps -f 2181
UID          PID    PPID  C STIME TTY          STAT       TIME CMD
quang        2181    2174  0  06:59 pts/0    Ss          0:00 bash
(base) quang@quang-FX503VM:~$ ps -f 2174
UID          PID    PPID  C STIME TTY          STAT       TIME CMD
quang        2174     982  0  06:59 ?        Ssl         0:00 /usr/libexec/gnome-termin
(base) quang@quang-FX503VM:~$ ps -f 982
UID          PID    PPID  C STIME TTY          STAT       TIME CMD
quang        982      1  0  06:57 ?        Ss          0:00 /lib/systemd/systemd --us
(base) quang@quang-FX503VM:~$
```

c. lệnh pstree:

Cây tiến trình như sau:



Giả sử như tìm tiến trình thermald (PID = 884)  
 pstree -p -s [PID của tiến trình con]  
 pstree -p -s 884

```
(base) quang@quang-FX503VM:~$ pstree -p 884
{thermald}(884)
(base) quang@quang-FX503VM:~$ pstree -p -s 884
systemd(1)---thermald(811)---{thermald}(884)
(base) quang@quang-FX503VM:~$ |
```

## Câu 2:

Chương trình sẽ in ra “I see 17 coconuts” vì tiến trình con được lệnh kết thúc (exit(0)) trước khi thực hiện câu lệnh printf, nên chỉ có câu lệnh printf của tiến trình cha được thực thi. Mà tiến trình cha và tiến trình con có tài nguyên bộ nhớ riêng nên biến num\_coconuts ở tiến trình cha vẫn là 17.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<pthread.h>
4 #include<unistd.h>
5 #include<sys/wait.h>
6
7 int main()
8 {
9     pid_t pid;
10    int num_coconuts = 17;
11    pid = fork();
12    if (pid == 0)
13    {
14        num_coconuts = 42;
15        exit(0);
16    }
17    else
18    {
19        wait(NULL); /*wait until the child terminates */
20        printf("I see %d coconuts!\n", num_coconuts);
21        exit(0);
22        // printf("Hello world!");
23    }
24 }
```

## Câu 3:

Trình bày về POSIX thread:

Trong khi tạo ra một thread thì ta đều có thể truyền vào các thuộc tính (pthread\_attr\_t)

Để khởi tạo một thuộc tính với giá trị mặc định

int pthread\_attr\_init(pthread\_attr\_t \*tattr);

```
16 pthread_attr_t attr;
17 int s;
18
19 /* initialize an attribute to the default value */
20 s= pthread_attr_init(&attr);
```

Hủy thuộc tính

int pthread\_attr\_destroy(pthread\_attr\_t \*tattr);

cài đặt thuộc tính chờ:

int pthread\_attr\_setdetachstate(pthread\_attr\_t \*tattr,int detachstate);

tham số detachstate ta có thể đặt:

PTHREAD\_CREATE\_DETACHED: thread ID cũng như các tài nguyên khác đều được tái sử dụng khi mà thread bị hủy

PTHREAD\_CREATE\_JOINABLE: thread sẽ đợi, dùng tham số này khi ta thực thi pthread\_join() trên thread

```
/* initialize an attribute to the default value */
s= pthread_attr_init(&attr);
// s = pthread_attr_destroy(&attr);
// if(s!=0)
//     // errExitEN(s,"pthread_attr_init");

s= pthread_attr_setdetachstate(&attr,PTHREAD_CREATE_JOINABLE);
pthread_attr_getdetachstate(&attr,&detachstate);
printf("%d",detachstate);
s= pthread_attr_setdetachstate(&attr,PTHREAD_CREATE_DETACHED);
pthread_attr_getdetachstate(&attr,&detachstate);
printf("%d",detachstate);
// if(s!=0)
//     // errExitEN(s,"pthread_attr_setdetachstate");
// s= pthread_attr_getschedparam()
```

0: đặt thành công PTHREAD\_CREATE\_DETACHED

1: đặt thành công PTHREAD\_CREATE\_JOINABLE

#### thuộc tính phạm vi:

int pthread\_attr\_setscope(pthread\_attr\_t \*tattr,int scope);

```
/* bound thread */
s= pthread_attr_setscope(&attr, PTHREAD_SCOPE_SYSTEM);

/* unbound thread */
s= pthread_attr_setscope(&attr, PTHREAD_SCOPE_PROCESS);
```

thread có giới hạn PTHREAD\_SCOPE\_SYSTEM

thread không giới hạn PTHREAD\_SCOPE\_PROCESS

#### Thiết lập thuộc tính guardsize:

int pthread\_attr\_setguardsize(pthread\_attr\_t \*attr, size\_t guardsize);

tham số guardsize cấp phát thêm bộ nhớ trước khi ngăn xếp đầy, giúp bảo vệ stack pointer khỏi việc bị overflow

#### Thiết lập phương thức lập lịch cho tiến trình:

int pthread\_attr\_setschedpolicy(pthread\_attr\_t \*tattr, int policy);

tham số policy bao gồm

SCHED\_FIFO (giải thuật first-in-first-out)

SCHED\_RR (giải thuật round-robin)

SCHED\_OTHER (giải thuật do người dùng định nghĩa)

```
/* set the scheduling policy to SCHED RR */
s= pthread_attr_setschedpolicy(&attr, SCHED_RR);
```

sử dụng giải thuật Round-Robin

#### Thiết lập thuộc tính kích thước cho ngăn xếp

int pthread\_attr\_setstacksize(pthread\_attr\_t \*tattr, int size);

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

/* setting a new size */
s = pthread_attr_setstacksize(&attr, size);
```

Thiết lập kích thước mới cho ngăn xếp

Ngoài các hàm thiết lập (set) ta còn có các hàm get tương tự cho mỗi hàm set, các hàm get được dùng để lấy giá trị thuộc tính, cũng như trả về 0 nếu các thiết lập(set) thành công.

#### **Câu 4: Viết chương trình thực hiện các công việc sau:**

a In ra dòng chữ “Welcome to IT007, I am 20520720”

b Mở tệp abcd.txt bằng vim editor

d Khi người dùng nhấn CTRL + C thì in ra dòng chữ “CTRL-C was pressed!Goodbye!”

```
TH03 > C:\Ba14-a.c > open_vim(void *args)
You, 2 minutes ago | 1 author (You)
1 #include <stdio.h>
2 #include <pthread.h>
3 #include <signal.h>
4 #include <stdlib.h>
5 #include <unistd.h>
6
7 long id_thread;
8 int loop = 2;
9 void on_sigint(){
10     if(loop == 1){
11         loop--;
12         printf("\nCTRL+C was pressed!Goodbye!\n");
13     }
14 }
15 void on_sigint2(){
16     if(loop == 2){
17         loop--;
18         system("killall vim");
19     }
20 }
21 void *open_vim(void *args){
22     system("vim abcd.txt");
23     signal(SIGINT, on_sigint2);
24     while(loop=1);
25 }
26 int main(){
27     // Print Welcome message
28     char *mssv = "20520720";
29     printf("Welcome to IT007, I am %s\n",mssv);
30     // Open vi editor
31     pthread_create(&id_thread,NULL,&open_vim,(void*)id_thread);
32     pthread_join(id_thread,NULL);
33     signal(SIGINT, on_sigint);
34     while(loop>0);
35     printf("\n\n");
36     return 0;
37 }
```