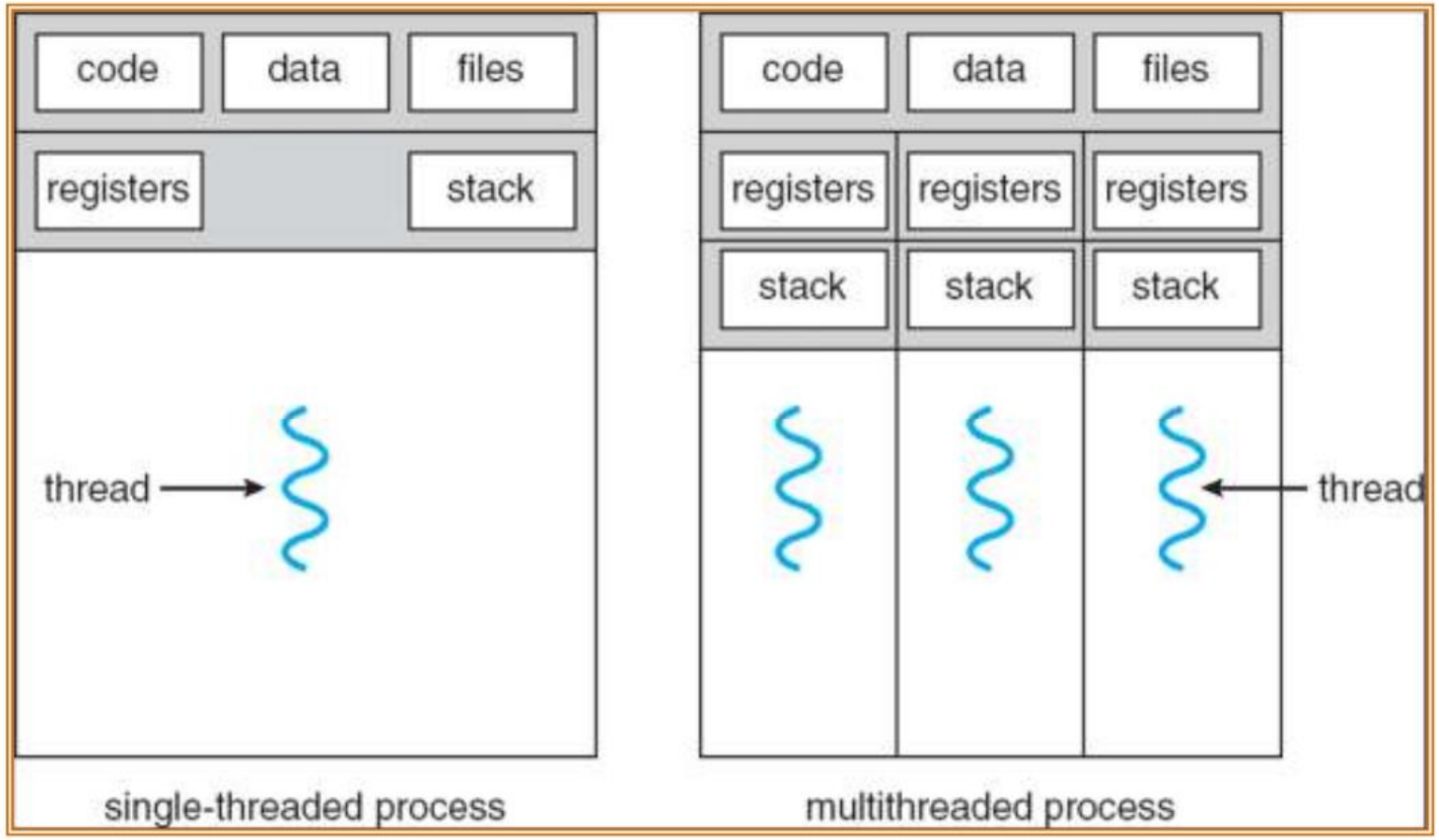
pthread

Single and Multithreaded Processes



Benefits of Thread

- Responsiveness – allow a program to continue running even if part of it is blocked (e.g. In a browser, multiple threads are working simultaneously to increase responsiveness)
- Resource Sharing – threads share the memory and resources of the process
  - Easier to perform communication
- Economy – it is more economical to create and context-switch threads. (Speed)
  - In Solaris, a process is thirty times slower to create and five times slower to context-switch
- Scheduling a thread is easy

- Scalability – utilizing multiprocessor architecture
- Modular Program Structure – easier to design a software

Pthreads

- A POSIX standard (IEEE 1003.1c) API for thread creation and synchronization

```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>

int sum;                /* global variable */
void *runner(void *param);
int main(int argc, char *argv[]) {
    pthread_t tid;
    pthread_attr_t attr;
    /*
    if (argc != 2) {
        fprintf(stderr, "usage: pthread <integer value>\n");
        exit(1);
    }
    if (atoi(argv[1]) < 0) {
        fprintf(stderr, "number must be >=0\n");
        exit(2);
    }
    */
    pthread_attr_init(&attr);                       /* create the thread */
    pthread_create(&tid, &attr, runner, argv[1]);   /* thread call function runner */

    printf("I am mother thread, I will wait for my child thread\n");
    pthread_join(tid, NULL);
    printf("from my child sum = %d\n", sum);
}

/* the thread will begin control in this function */
void *runner (void *param) {
    int upper = atoi(param);    //upper is local variable
    sum += upper;               //sum is shared among the thread(s)
    printf("I am child thread, sum value here is %d\n",sum);
    pthread_exit(0);
}
```

```
1  #include <pthread.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4
5  int sum;                          /* global var */
6  void *runner(void *param);
7
8  int main(int argc, char *argv[]) {
9
10     pthread_t                     tid;
11     pthread_attr                  attr;
12     pthread_attr_init(&attr);
13
14     pthread_create(&tid, &attr, runner, argv[1]);
15
16     pthread_join(tid,NULL);
17
18     printf("sum = %d\n",sum);
19     return 0;
20 }
22 void *runner(void *param) {
23     int upper = atoi(param);
24     int i;
25     sum = 0;                      /* from line 3 */
26     if (upper > 0) {
27         for (i = 0; i <= upper; i++ )
28             sum += i;
29     }
30     pthread_exit(0);
31 }
```

return เป็น void *

global ทำให้ data ร่วมกัน (fork() เห็นร่วมกันแบบนี้ไม่ได้)

ส่งเป็น void *

gcc -o q6 lab6_pthread.c –pthread

```
suntana@DESKTOP-IQOCR48:~/lab6$ ls -l
total 4
-rwxrwxr-x 1 suntana suntana 891 Jan 18 16:47 lab6_pthread.c
suntana@DESKTOP-IQOCR48:~/lab6$ gcc -o q6 lab6_pthread.c
/usr/bin/ld: /tmp/cc2b5Z1a.o: in function `main':
lab6_pthread.c:(.text+0x4f): undefined reference to `pthread_create'
/usr/bin/ld: lab6_pthread.c:(.text+0x6c): undefined reference to `pthread_join'
collect2: error: ld returned 1 exit status
suntana@DESKTOP-IQOCR48:~/lab6$ gcc -o q6 lab6_pthread.c -pthread
suntana@DESKTOP-IQOCR48:~/lab6$ ./q6
I am mother thread, I will wait for my child thread
Segmentation fault (core dumped)
suntana@DESKTOP-IQOCR48:~/lab6$ ./q6 56
I am mother thread, I will wait for my child thread
I am child thread, sum value here is 32546
from my child sum = 32546
suntana@DESKTOP-IQOCR48:~/lab6$
```

123

`-pthread` tells the compiler to link in the pthread library as well as configure the compilation for threads.

For example, the following shows the macros that get defined when the `-pthread` option gets used on the GCC package installed on my Ubuntu machine:

```
$ gcc -pthread -E -dM test.c > dm.pthread.txt
$ gcc          -E -dM test.c > dm.nopthread.txt
$ diff dm.pthread.txt dm.nopthread.txt
152d151
< #define _REENTRANT 1
208d206
< #define __USE_REENTRANT 1
```

Using the `-lpthread` option only causes the pthread library to be linked - the pre-defined macros don't get defined.

Bottom line: you should use the `-pthread` option.

https://stackoverflow.com/questions/23250863/difference-between-pthread-and-lpthread-while-compiling

1. เขียนโปรแกรม **xxx_Lab6q1.c** ตาม requirement ต่อไปนี้

2.1 รับเลขจำนวนเต็มบวกจากผู้ใช้หนึ่งจำนวน ซึ่งเป็น parameter ตอนเรียกใช้ จากนั้น

2.2 สร้าง child Thread โดย child thread คำนวณ csum จาก 1 ถึง 2 เท่าของเลขดังกล่าว

2.3 parent thread คำนวณผลบวก msum จาก 1 ถึงเลขจำนวนนั้น

2.4 ให้ parent แสดงค่า ผลต่างของ csum กับ msum

2.5 หากไม่ join คำตอบที่ได้มีกี่แบบ อะไรบ้าง (โดยมากจะพบ 3 แบบ)

(หมายเหตุ) การ compile โปรแกรมที่เรียกใช้ pthread ให้ใช้

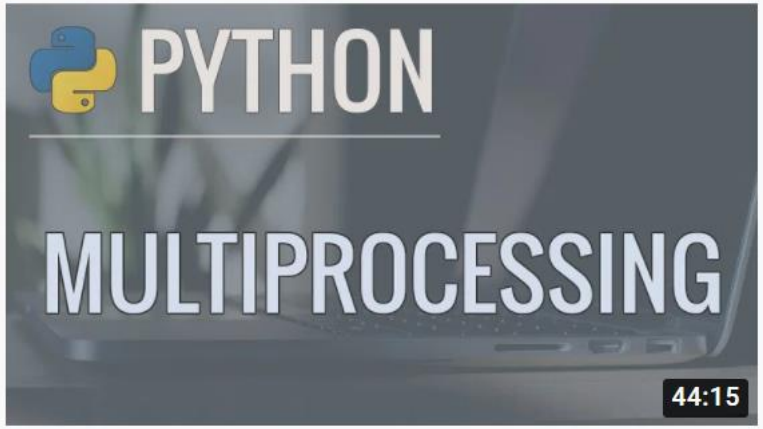
option –pthread ตอนสั่ง (เปลี่ยนเป็น -l pthread หรือ -lpthread ถ้า gcc คุณไม่รู้จัก -pthread)

Python Threading Tutorial: Run Code Concurrently Using the Threading Module

274K views • 1 year ago

Corey Schafer

In this video, we will be learning how to use threads in **Python**. This video is sponsored by Brilliant. Go to https://brilliant.org/cms to ...



Python Multiprocessing Tutorial: Run Code in Parallel Using the Multiprocessing Module

250K views • 1 year ago

Corey Schafer

In this video, we will be learning how to use multiprocessing in **Python**. This video is sponsored by Brilliant.