ใบงาน 6

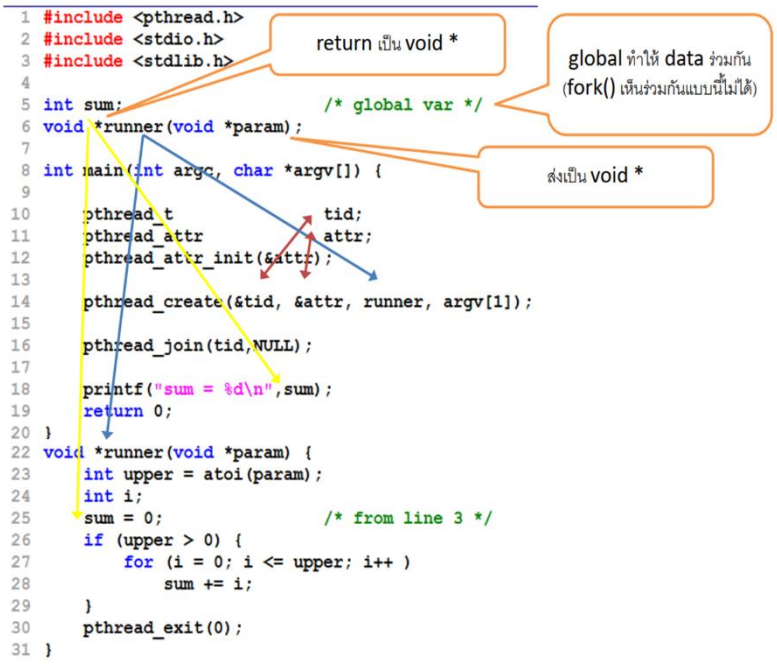
<u>วัตถุประสงค์</u> เพื่อทดลองใช้ **pthread**

ในการเรียกใช้ POSIX Thread

A. include <pthread.h>

B. ประกาศ tid และ attr โดย attr เป็นโครงสร้างข้อมูลที่ thread tid จะใช้ (แต่เราใช้ค่า default ด้วยการให้ pthread_attr_init(&attr); จัดการ

เครื่องหมาย &หน้าตัวแปร ในภาษาซี หมายถึงส่งเป็น address ของตัวแปรนั้น




```
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 }
```

C. ทำการสร้าง thread ด้วยคำสั่ง pthread_create(); ในที่นี้ พารามิเตอร์ตัวที่ 3 คือ function ที่เป็นโค้ดที่ tid จะไปทำงาน พารามิเตอร์ตัวสุดท้ายคือ พารามิเตอร์ที่ส่งเป็นพารามิเตอร์ไปให้ function ที่จะไปทำงาน ความเป็น void * หมายถึงว่ารับมาเป็น address ของข้อมูลโดยไม่สนใจ type เมื่อจะนำมาใช้ จึงต้องแปลงให้เป็น type ที่โปรแกรมจะนำไปใช้ได้ที่บรรทัดที่ 23

D. เรียก pthread_join(); เพื่อให้แม่รอ thread ลูกที่สร้างขึ้นทำงานเสร็จก่อน

การ compile โปรแกรมที่เรียกใช้ pthread ให้ใช้
option –pthread ตอนสั่ง (เปลี่ยนเป็น -l pthread หรือ -lpthread ถ้า gcc คุณไม่รู้จัก -pthread)

หมายเหตุ

เรียกตัวแปร sum บรรทัดที่ 5 ซึ่งประกาศนอก main ว่า global variable หมายถึงว่าทุก function เห็น sum นี้เดียวกัน (พึงระวังอย่าประกาศ sum เป็น local ซ้ำเพราะโปรแกรมจะใช้ local หากมีให้อ้างถึง) พฤติกรรมนี้บางเอกสารกล่าวว่าเหมือน static attribute ของ java class แต่ java เรียกตัวแปรที่มี scope แบบนี้ว่า class variable ไม่เรียก global variable

บรรทัดที่ 6 เรียกว่าการทำ prototype กล่าวคือเป็นการบอกโครงสร้างของ runner หากจะไม่มีบรรทัด 6 ถือว่า compiler ไม่รู้จัก runner บรรทัดที่ 14 เพราะยังไม่ได้สร้าง การประกาศก่อนเขียนจริง ทำให้ลำดับของโค้ดส่วน main อยู่ต้นไฟล์ (หมายถึงว่าสามารถเขียนทั้ง function (บรรทัดที่ 22 – 31) ก่อน main

คำสั่ง

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)

กำหนดส่ง (TBA)