#include <pthread.h>

#include <stdio.h>

#include <stdlib.h>

int sum;

int T\_NUM = 5; defind number of threads = 5

void \*runner(void \*param); สร้างฟั่งค์ชั่น prototype เพื่อบอกว่ามี ฟั่งค์ชั่น runner อยู่ในนี้

struct two{ สร้างตัวแปรโดยใช้ struct

char\* tname;

int lower;

int upper;

};

int main(int argc,char \*argv[]) {

pthread\_t tid1; the thread identifier 1-5

pthread\_t tid2;

pthread\_t tid3;

pthread\_t tid4;

pthread\_t tid5;

pthread\_attr\_t attr; set of thread attributes

ต่อไปคือโค้ดที่ใช้สำหรับตรวจ command line

if (argc != 2) { ถ้า argc ไม่เท่ากับ2 แสดงผลลัพธ์ด้านล่าง

fprintf(stderr," usage: semaj.o <interger value > \n");

return -1;

}

if (atoi(argv[1]) < 0){ ค่าต้องมากกว่าหรือเท่ากับ 0

fprintf(stderr," %d must be >= 0 \n", atoi(argv[1]));

return -1;

}

/\*for (int i = 0 ; i < T\_NUM; i++) {

struct two \*l[i] = (struct two \*)malloc(sizeof(struct two)); ลูป ทำ malloc

} \*/

struct two \*l1 = (struct two \*)malloc(sizeof(struct two)); ทำการ molloc เพื่อจองพื้นที่ในการทำงาน

struct two \*l2 = (struct two \*)malloc(sizeof(struct two));

struct two \*l3 = (struct two \*)malloc(sizeof(struct two));

struct two \*l4 = (struct two \*)malloc(sizeof(struct two));

struct two \*l5 = (struct two \*)malloc(sizeof(struct two));

int max = atoi(argv[1]); สร้างตัวแปรmax จากเลขที่พิมพ์

sum = 0 ;

((struct two \*)l1)-> tname = "Thread #1 lower"; ตัวควบคุมในการบวกเลข 0 ถึง max

((struct two \*)l1)-> lower = 0; /\*Start form Zero\*/

((struct two \*)l1)-> upper = ((struct two \*)l1)->lower +(max/T\_NUM);

((struct two \*)l2)-> tname = "Thread #2 middle lower";

((struct two \*)l2)-> lower = ((struct two \*)l1)->upper +1;

((struct two \*)l2)-> upper = ((struct two \*)l2)->lower +(max/T\_NUM);

((struct two \*)l3)-> tname = "Thread #3 middle";

((struct two \*)l3)-> lower = ((struct two \*)l2)->upper +1;

((struct two \*)l3)-> upper = ((struct two \*)l3)->lower +(max/T\_NUM);

((struct two \*)l4)-> tname = "Thread #4 middle upper";

((struct two \*)l4)-> lower = ((struct two \*)l3)->upper +1;

((struct two \*)l4)-> upper = ((struct two \*)l4)->lower +(max/T\_NUM);

((struct two \*)l5)-> tname = "Thread #5 upper";

((struct two \*)l5)-> lower = ((struct two \*)l4)->upper +1;

((struct two \*)l5)-> upper = max;

/\*for(int i = 0; i < T\_NUM; i++) {

pthread\_create(&tid[i],&attr,runner,(void \*) l[i]); ลูป ทำ create join

pthread\_join(tid[i], NULL);

}

printf("sum = %d \n",sum);\*/

pthread\_attr\_init(&attr); // atr = pthread\_attr\_init(); กำหนดค่าให้กลับตัวแปล attr

pthread\_create(&tid1,&attr,runner,(void \*) l1); สร้างเทรด บอก id ฟั่งค์ชันที่ใช้ เลข

pthread\_create(&tid2,&attr,runner,(void \*) l2);

pthread\_create(&tid3,&attr,runner,(void \*) l3);

pthread\_create(&tid4,&attr,runner,(void \*) l4);

pthread\_create(&tid5,&attr,runner,(void \*) l5);

pthread\_join(tid1, NULL); เป็นการให้ thread main รอ thread ที่สร้าง

pthread\_join(tid2, NULL); ไปทำฟั่งชั่นให้เรียบร้อยก่อน ค่อยไปทำบรรทัดต่อไป

pthread\_join(tid3, NULL);

pthread\_join(tid4, NULL);

pthread\_join(tid5, NULL);

printf("sum = %d \n",sum); ปลิ้นผลลัพธ์ ในการบวก

}

void \*runner (void \*param){ สร้างคลาส runner ใช่ในการ บวกเลขเลื่อยๆไปถึง max

int upper = ((struct two \*)param)->upper;

int lower = ((struct two \*)param)->lower;

char \*name = ((struct two \*)param)->tname;

for(int i = lower; i <= upper; i++)

printf("Working on %s and get %d\n",name , sum += i);

pthread\_exit(0);

}