컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <pthread.h>

#include "ssu\_runtime.h"

void \*ssu\_thread1(void \*arg);

void \*ssu\_thread2(void \*arg);

pthread\_mutex\_t mutex1 = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_mutex\_t mutex2 = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t cond1 = PTHREAD\_COND\_INITIALIZER;

pthread\_cond\_t cond2 = PTHREAD\_COND\_INITIALIZER;

int count = 0;

int input = 0;

int t1 = 0, t2 = 0;

int main(void)

{

pthread\_t tid1, tid2;

int status;

gettimeofday(&begin\_t, NULL);

if (pthread\_create(&tid1, NULL, ssu\_thread1, NULL) != 0) {

fprintf(stderr, "pthread\_create error\n");

exit(1);

}

if (pthread\_create(&tid2, NULL, ssu\_thread2, NULL) != 0) {

fprintf(stderr, "pthread\_create error\n");

exit(1);

}

while (1) {

printf("2개 이상의 수 입력 : ");

scanf("%d", &input);

if (input >= 2) {

pthread\_cond\_signal(&cond1);

break;

}

}

pthread\_join(tid1, (void \*)&status);

pthread\_join(tid2, (void \*)&status);

printf("complete\n");

gettimeofday(&end\_t, NULL);

ssu\_runtime(&begin\_t, &end\_t);

exit(0);

}

void \*ssu\_thread1(void \*arg)

{

while (1) {

pthread\_mutex\_lock(&mutex1);

if (input < 2) {

pthread\_cond\_wait(&cond1, &mutex1); // cond1시그널을 기다림

}

if (input == count) {

pthread\_cond\_signal(&cond2); // cond2에 시그널을 보냄

break;

}

if (count == 0) {

t2++;

count++;

printf("Thread 1 : %d\n", t1);

}

else if (count % 2 == 0) {

t1 += t2;

count++;

printf("Thread 1 : %d\n", t1);

}

pthread\_cond\_signal(&cond2); // cond2에 시그널을 보냄

pthread\_cond\_wait(&cond1, &mutex1); // cond1시그널을 기다림

pthread\_mutex\_unlock(&mutex1);

}

return NULL;

}

void \*ssu\_thread2(void \*arg)

{

while (1) {

pthread\_mutex\_lock(&mutex2);

if (input < 2) {

pthread\_cond\_wait(&cond2, &mutex2); // cond2 시그널을 기다림

}

if (input == count) {

pthread\_cond\_signal(&cond1); // cond1 시그널을 보냄

break;

}

if (count == 1) {

count++;

printf("Thread 2 : %d\n", t2);

}

else if (count % 2 == 1) {

t2 += t1;

count++;

printf("Thread 2 : %d\n", t2);

}

pthread\_cond\_signal(&cond1); // cond1 시그널을 보냄

pthread\_cond\_wait(&cond2, &mutex2); // cond2 시그널을 기다림

pthread\_mutex\_unlock(&mutex2);

}

return NULL;

}

1. 실행 결과

