컴퓨터학부 20152385 송민구

1. 소스 코드

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/time.h>

#include <pthread.h>

#include <errno.h>

#include "ssu\_runtime.h"

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

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

int glo\_val1 = 1, glo\_val2 = 2;

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

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

int main(void)

{

pthread\_t tid1, tid2;

gettimeofday(&begin\_t, NULL);

pthread\_create(&tid1, NULL, ssu\_thread1, NULL);

pthread\_create(&tid2, NULL, ssu\_thread2, NULL);

pthread\_join(tid1, NULL);

pthread\_join(tid2, NULL);

pthread\_mutex\_destroy(&lock);

pthread\_cond\_destroy(&cond);

gettimeofday(&end\_t, NULL);

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

exit(0);

}

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

{

sleep(1);

glo\_val1 = 2;

glo\_val2 = 1;

if (glo\_val1 > glo\_val2) {

pthread\_cond\_broadcast(&cond); // broadcast()로 cond\_timedwait()에 신호를 날린다

}

printf("ssu\_thread1 end\n");

return NULL;

}

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

{

struct timespec timeout;

struct timeval now;

pthread\_mutex\_lock(&lock);

gettimeofday(&now, NULL);

timeout.tv\_sec = now.tv\_sec + 5;

timeout.tv\_nsec = now.tv\_usec \* 1000;

if (glo\_val1 <= glo\_val2) {

printf("ssu\_thread2 sleep\n");

if (pthread\_cond\_timedwait(&cond, &lock, &timeout) == ETIMEDOUT) { // 정해진 시간까지 기다리다 반환

printf("timeout\n");

}

else {

printf("glo\_vaa1 = %d, glo\_val2 = %d\n", glo\_val1, glo\_val2);

}

}

pthread\_mutex\_unlock(&lock);

printf("ssu\_thread2 end\n");

return NULL;

}

1. 실행 결과

