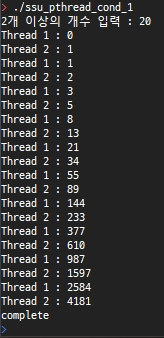
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <pthread.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;  // 새 스레드 생성  if (pthread\_create(&tid1, NULL, ssu\_thread1, NULL) != 0)  {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  // 새 스레드 생성  if (pthread\_create(&tid2, NULL, ssu\_thread2, NULL) != 0)  {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  while (1)  {  printf("2개 이상의 개수 입력 : ");  scanf("%d", &input);  if (input>=2)  {  pthread\_cond\_signal(&cond1);  break;  }  }  // thread1이 끝날때 까지 대기  pthread\_join(tid1, (void\*)&status);  // thread2가 끝날때 까지 대기  pthread\_join(tid2, (void\*)&status);  printf("complete \n");  exit(0);  }  void \*ssu\_thread1(void \*arg)  {  while (1)  {  // 시그널이 도착할때 까지 대기  pthread\_mutex\_lock(&mutex1);  if (input <2)  pthread\_cond\_wait(&cond1, &mutex1);  // 카운트가 input만큼 도달하면 thread2를 시작하고 이 스레드를 종료한다.  if (input == count)  {  pthread\_cond\_signal(&cond2);  break;  }  // 피보나치 시작  if(count == 0)  {  t2++;  count++;  printf("Thread 1 : %d\n", t1);  }  // 짝수번 피보나치  else if (count %2 == 0)  {  t1+=t2;  count++;  printf("Thread 1 : %d\n", t1);  }  pthread\_cond\_signal(&cond2);  pthread\_cond\_wait(&cond1, &mutex1);  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);  // 카운트가 input만큼 도달하면 thread1을 시작하고 이 스레드 종료  if (input==count)  {  pthread\_cond\_signal(&cond1);  break;  }  // 두번쌔 피보나치 시작  /\*second term of fibonacci\*/  if(count==1)  {  count++;  printf("Thread 2 : %d\n", t2);  }  // 홀수번 피보나치  else if (count%2 ==1)  {  t2+=t1;  count++;  printf("Thread 2 : %d\n", t2);  }  pthread\_cond\_signal(&cond1);  pthread\_cond\_wait(&cond2, &mutex2);  pthread\_mutex\_unlock(&mutex2);  }  return NULL;  } |