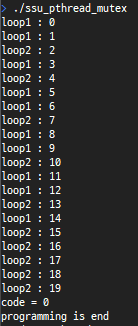
컴퓨터학부 20162448 김병준

1. 결과



1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <pthread.h>  void \*ssu\_loop1(void \*arg);  void \*ssu\_loop2(void \*arg);  pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;  int shared\_value;  int main(void)  {  pthread\_t tid1, tid2;  int status;  shared\_value = 0;  // 새 스레드 생성  if (pthread\_create(&tid1, NULL, ssu\_loop1, NULL) != 0)  {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  sleep(1);  // 새 스레드 생성  if (pthread\_create(&tid2, NULL, ssu\_loop2, NULL) != 0)  {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  // 스레드 1 종료 대기  if (pthread\_join(tid1, (void \*)&status) != 0)  {  fprintf(stderr, "pthread\_join error\n");  exit(1);  }  // 스레드 2 종료 대기  if (pthread\_join(tid2, (void \*)&status) != 0)  {  fprintf(stderr, "pthread\_join error\n");  exit(1);  }  // mutex 변수 해제  status = pthread\_mutex\_destroy(&mutex);  printf("code = %d\n", status);  printf("programming is end\n");  exit(0);  }  void \*ssu\_loop1(void \*arg)  {  int i;  for (i=0;i<10;i++)  {  // mutex 변수 잠금  pthread\_mutex\_lock(&mutex);  printf("loop1 : %d\n", shared\_value);  // 변수 증가  shared\_value++;  // 10에 도달시 종료  if (i == 10)  return NULL;  // mutex 변수 해제  pthread\_mutex\_unlock(&mutex);  sleep(1);  }  return NULL;  }  void \*ssu\_loop2(void \*arg)  {  int i;  for (i=0;i<10;i++)  {  // mutex 변수 잠금  pthread\_mutex\_lock(&mutex);  printf("loop2 : %d\n", shared\_value);  // 변수 증가  shared\_value++;  // mutex 변수 해제  pthread\_mutex\_unlock(&mutex);  sleep(2);  }  return NULL;  } |