1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <unistd.h>  #include <pthread.h>  void \*ssu\_loop1(void \*arg);  void \*ssu\_loop2(void \*arg);  //뮤텍스 변수 초기화  pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;  //공유 변수  int shared\_value;  int main(void) {  pthread\_t tid1, tid2;  int status;  shared\_value = 0;  //스레드 생성  if(pthread\_create(&tid1, NULL, ssu\_loop1, NULL) != 0) {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  sleep(1);  if(pthread\_create(&tid2, NULL, ssu\_loop2, NULL) != 0) {  fprintf(stderr, "pthread\_create error\n");  exit(1);  }  //스레드가 끝날 때까지 대기  if(pthread\_join(tid1, (void \*)&status) != 0) {  fprintf(stderr, "pthread\_join error\n");  exit(1);  }  if(pthread\_join(tid2, (void \*)&status) != 0) {  fprintf(stderr, "pthread\_join error 2\n");  exit(1);  }  //뮤텍스 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++) {  //lock  pthread\_mutex\_lock(&mutex);  printf("loop1 : %d\n", shared\_value);  shared\_value++;  if(i == 10)  return NULL;    //unlock  pthread\_mutex\_unlock(&mutex);  sleep(1);  }  return NULL;  }  void \*ssu\_loop2(void \*arg) {  int i;  for(i = 0; i < 10; i++) {  //lock  pthread\_mutex\_lock(&mutex);  printf("loop2 : %d\n", shared\_value);  shared\_value++;  //unlock  pthread\_mutex\_unlock(&mutex);  sleep(2);  }  return NULL;  } |

2. 실행결과

