1. 소스코드

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <pthread.h>  #define VALUE\_DONE 10  #define VALUE\_STOP1 3  #define VALUE\_STOP2 6  //뮤텍스 변수 초기화  pthread\_mutex\_t lock = PTHREAD\_MUTEX\_INITIALIZER;  //조건 변수 초기화  pthread\_cond\_t cond = PTHREAD\_COND\_INITIALIZER;  void \*ssu\_thread1(void \*arg);  void \*ssu\_thread2(void \*arg);  //공용 변수  int glo\_val = 0;  int main(void) {  pthread\_t tid1, tid2;  //스레드 생성  pthread\_create(&tid1, NULL, &ssu\_thread1, NULL);  pthread\_create(&tid2, NULL, &ssu\_thread2, NULL);  //스레드가 종료될 때까지 대기  pthread\_join(tid1, NULL);  pthread\_join(tid2, NULL);  printf("final value: %d\n", glo\_val);  exit(0);  }  void \*ssu\_thread1(void \*arg) {  while(1) {  //lock  pthread\_mutex\_lock(&lock);  //cond를 받을 때까지 대기  pthread\_cond\_wait(&cond, &lock);  glo\_val++;  printf("global value ssu\_thread1: %d\n", glo\_val);  //unlock  pthread\_mutex\_unlock(&lock);  if(glo\_val >= VALUE\_DONE)  return NULL;  }  }  void \*ssu\_thread2(void \*arg) {  while(1) {  //lock  pthread\_mutex\_lock(&lock);  if(glo\_val < VALUE\_STOP1 || glo\_val > VALUE\_STOP2)  pthread\_cond\_signal(&cond);//cond를 기다리고 있는 스레드 중 하나를 재시작  else {  glo\_val++;  printf("global value ssu\_thread2: %d\n", glo\_val);  }  //unlock  pthread\_mutex\_unlock(&lock);  if(glo\_val >= VALUE\_DONE)  return NULL;  }  } |

2. 실행결과

