운영체제 - 멀티 쓰레드 프로그래밍 과제

* **코드**

//

// 이름 : 쓰레드 공유변수 문맥교환

// 날짜 : 2019. 11. 14.

// 설명 : 다수의 쓰레드가 공통의 변수를 공유할 때

// 임계영역에서의 문맥교환의 문제점

//

#include <stdio.h>

#include <pthread.h>

#include <unistd.h>

// 공유 변수

**static** **volatile** **int** counter = 0;

**void** \*thread\_one(**void** \*arg){

// 출력 및 카운트

**for**(**int** i = 0; i<1000000; i++){

counter += 1;

printf("[%s] 100 usec\n", (**char** \*)arg);

usleep(100);

}

// 끝나면 출력

printf("[%s] : done\n", (**char** \*) arg);

printf("counter : %d\n", counter);

**return** **NULL**;

}

**void** \*thread\_two(**void** \*arg){

// 출력 및 카운트

**for**(**int** i = 0; i<1000000; i++){

counter += 1;

printf("[%s] 50 usec\n", (**char** \*)arg);

usleep(50);

}

// 끝나면 출력

printf("[%s] : done\n", (**char** \*) arg);

printf("counter : %d\n", counter);

**return** **NULL**;

}

**void** \*thread\_three(**void** \*arg){

// 출력 및 카운트

**for**(**int** i = 0; i<1000000; i++){

counter += 1;

printf("[%s] 30 usec\n", (**char** \*)arg);

usleep(30);

}

// 끝나면 출력

printf("[%s] : done\n", (**char** \*) arg);

printf("counter : %d\n", counter);

**return** **NULL**;

}

**void** \*thread\_four(**void** \*arg){

// 출력 및 카운트

**for**(**int** i = 0; i<1000000; i++){

counter += 1;

printf("[%s] 30 usec\n", (**char** \*)arg);

usleep(30);

}

// 끝나면 출력

printf("[%s] : done\n", (**char** \*) arg);

printf("counter : %d\n", counter);

**return** **NULL**;

}

**void** \*thread\_five(**void** \*arg){

// 출력 및 카운트

**for**(**int** i = 0; i<1000000; i++){

counter += 1;

printf("[%s] 10 usec\n", (**char** \*)arg);

usleep(10);

}

// 끝나면 출력

printf("[%s] : done\n", (**char** \*) arg);

printf("counter : %d\n", counter);

**return** **NULL**;

}

**int** main(**int** argc, **const** **char** \*argv[]){

pthread\_t t1, t2, t3, t4, t5;

printf("main : begin (counter = %d)\n", counter);

// 쓰레드 생성

pthread\_create(&t1, **NULL**, thread\_one, "A");

pthread\_create(&t2, **NULL**, thread\_two, "B");

pthread\_create(&t3, **NULL**, thread\_three, "C");

pthread\_create(&t4, **NULL**, thread\_four, "D");

pthread\_create(&t5, **NULL**, thread\_five, "E");

// 쓰레드 대기

pthread\_join(t1,**NULL**);

pthread\_join(t2,**NULL**);

pthread\_join(t3,**NULL**);

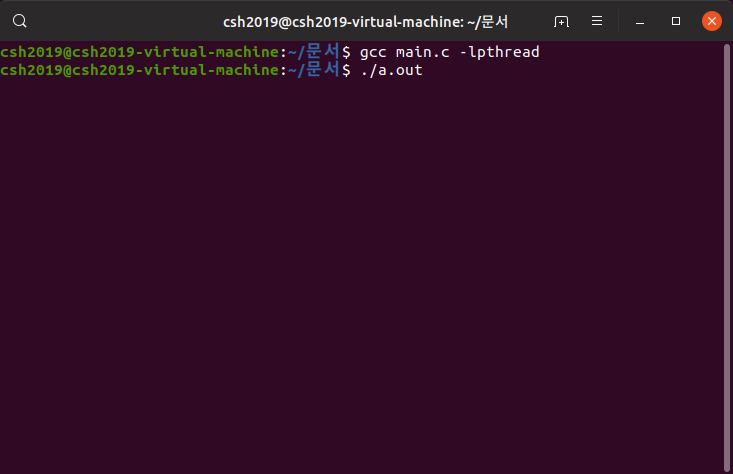
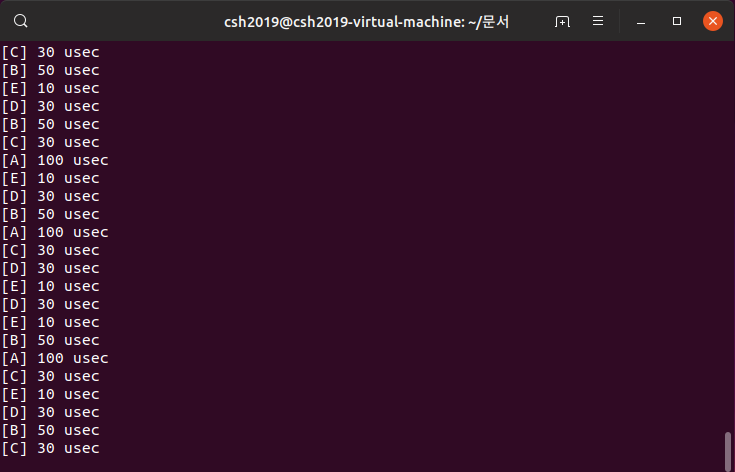
pthread\_join(t4,**NULL**);

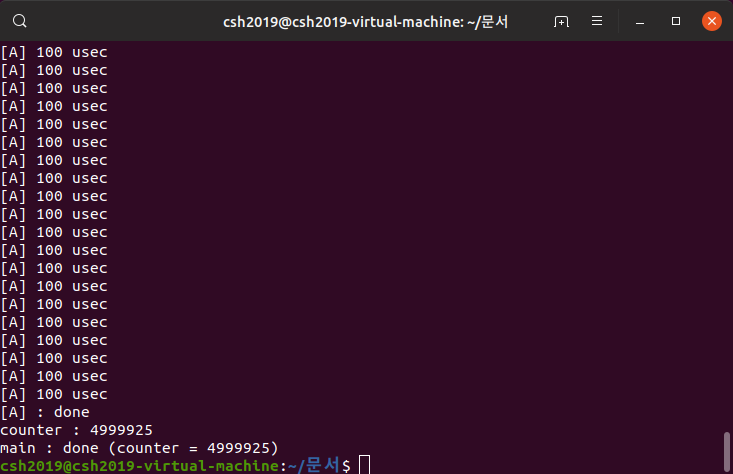
pthread\_join(t5,**NULL**);

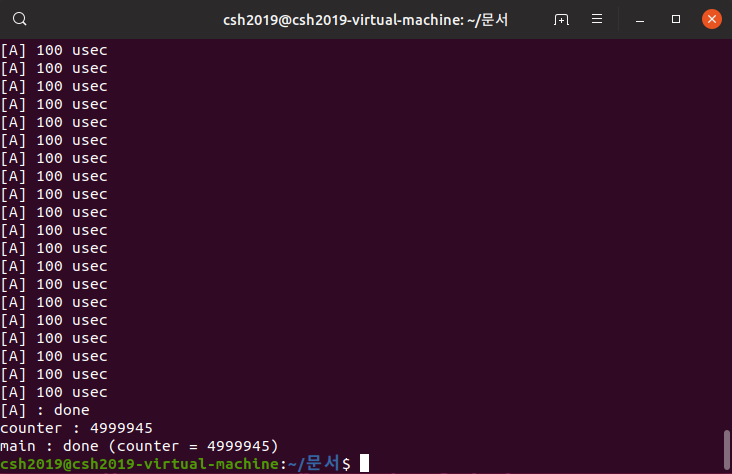
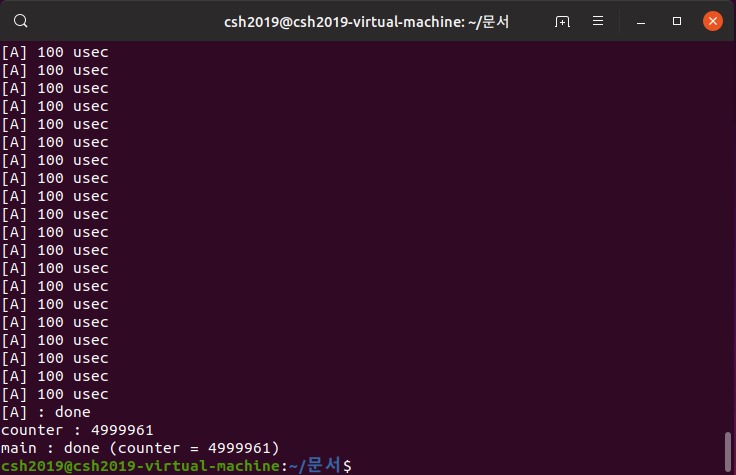
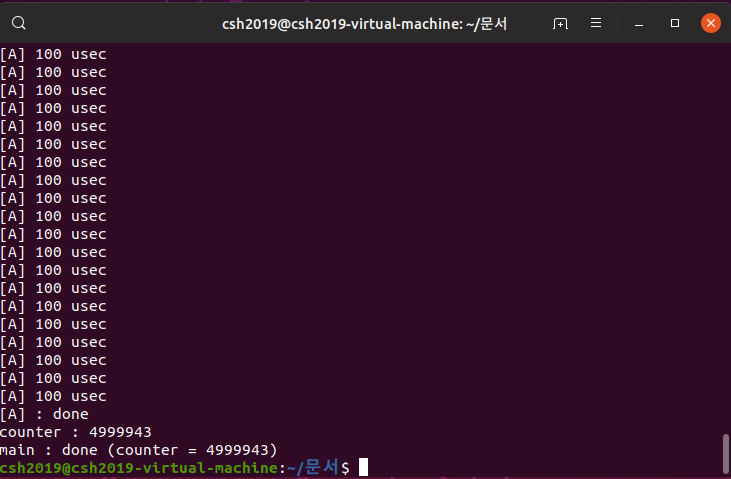
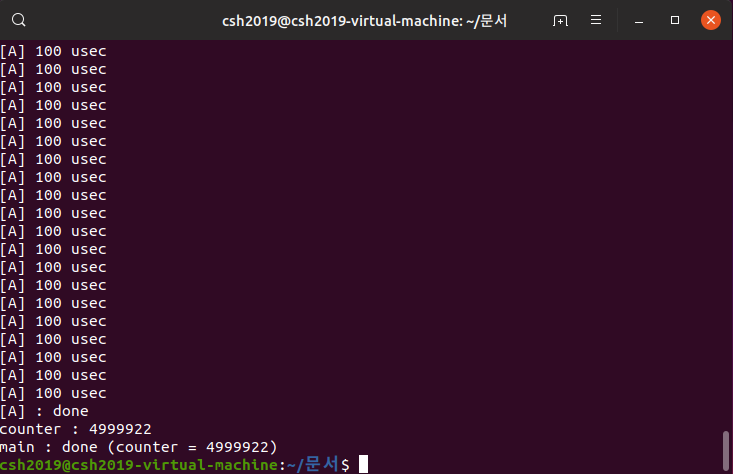
printf("main : done (counter = %d)\n", counter);

**return** 0;

}

* **첫번째 실행결과**



* **두번째 실행결과**
* **세번째 실행결과**
* **네번째 실행결과**
* **다섯번째 실행결과**