

**[506489] 시스템프로그래밍****실습 #11 문제 및 보고서**

<b>이름</b>	박영주
<b>학번</b>	20175105
<b>소속 학과/대학</b>	빅데이터
<b>분반</b>	01 (담당교수: 김태운)

### <주의사항>

- 개별 과제 입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)

- 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.

- 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.

- SmartLEAD 제출 데드라인:

- 월요일 분반: 다음 실습 시간 전날까지(일요일 까지)

- 수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)

- 데드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점(5 일 경과 시 0 점)

- 주말/휴일/학교행사 등 모든 날짜 카운트 함

- 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함

- 예외 없음

- SmartLEAD 에 아래의 파일을 제출해 주세요

- 보고서(PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)

- 보고서 파일명에 이름과 학번을 입력해 주세요.

- 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)

### <개요>

이번 과제는 스레드를 생성/관리하고 동기화를 구현하는 내용입니다.

## &lt;실습 과제&gt;

**[Q 0] 요약 [배점: 10]**

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 thread 와 mutex 및 deadlock 의 동작 흐름을 알게 되었습니다.

또한, 교착 상태를 해결하기 위한 방법이 무엇이 있는 찾아보는 계기가 되었습니다.

**[Q 1] 스레드 [18 점]**

강의자료 p.29 [Pthread: 예제]를 참고하여 2 개의 스레드를 생성하는 프로그램을 작성하세요. 각 스레드는 생성할 때 struct StudentInfo 구조체를 전달 받으며, 구조체에는 name, major, univ 정보가 각각의 서로 다른 char 배열로 정의되어 있다. 스레드 1 과 2 에 전달하는 구조체는 각각 stu1, stu2 이며, 아래와 같이 정보가 저장되어 있다.

- 'stu1' = {"Brad Pitt", "Computer Engineering", "Holly University"}
- 'stu2' = {"Angelina Jollie", "Computer Science", "Wood University"}

'start\_thread' 함수는 for-loop에서 name, major, univ 정보를 하나의 라인으로 출력하며, for-loop 은 0 부터 9 까지 총 10 번을 반복한다 (즉, 동일한 정보를 10 번 반복 출력하고 종료). 지금까지 언급된 부분을 제외하고는 p.29 의 코드를 그대로 사용해야 함 (특히 sleep 구문).

[Task 1] (printf 구문 위 아래로 sleep 을 수행하는) 프로그램을 구동하고, 터미널 화면을 캡처해서 본 문서에 첨부하세요.

[Task 2] 소스코드에서 sleep 함수를 모두 주석처리 하고 프로그램을 구동하세요. 터미널 화면을 캡처해서 본 문서에 첨부하세요.

소스코드도 제출해야 합니다.

답변:

**[Task 1]**

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q01$ ./pthread
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
yeongju@vm-ubuntu20:~/sp2021-2/w11/q01$
```

**[Task 2]**

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q01$ ./pthread
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Angelina Jollie, major: Computer Science, univ: Wood University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
name: Brad Pitt, major: Computer Engineering, univ: Holly University
yeongju@vm-ubuntu20:~/sp2021-2/w11/q01$
```

**[Q 2] 스레드 + 현금인출 [18 점]**

강의자료 p.12 에 제시된 withdraw 함수를 사용할 수 있도록 프로그램을 작성하세요. 계좌를 나타내는 struct account 는 자유롭게 구현하면 됩니다. 단, 멤버 변수로 int saving 과 int balance 를 가지고 있어야 합니다. 여기서 saving 은 계좌에 있는 실제 잔액이고, balance 는 통장에 찍힌 잔액 숫자 입니다. 참고로, saving 과 balance 는 다를 수 있습니다. ‘withdraw’ 함수의 구현을 보면 (2)번에서 통장에 찍힌 숫자 balance 를 업데이트 한 후, (3)번에서 실제로 돈을 인출하며 saving 의 값을 업데이트 합니다. 예를 들어, ‘balance’ 업데이트 후 비정상 종료하면 ‘saving’의 값과 ‘balance’의 값이 달라지게 됩니다. (disburse\_money 함수는 돈을 실제로 인출하며, saving 값을 업데이트 합니다.)

현금 인출 과정을 테스트 하기 위해 main 함수를 만드세요. 2 개 또는 그 이상의 스레드를 만들고, 같은 계좌에서 돈을 인출하는 프로그램을 구성하세요. 뮤텝스를 사용하지 않아서 잘못된 결과를 유발하는 경우를 만드세요. 오류가 발생한 상황을 확인할 수 있도록, 터미널에 메시지를 출력하세요. (즉, 터미널에 출력된 결과를 통해서, 오류가 발생했다는 것을 이해할 수 있어야 합니다. ‘disburse\_money’ 함수에서 오류를 출력하도록 구현하면 됩니다.) 어떤 상황에서 오류가(=즉, 원치 않는 결과가) 발생하는지, 그리고 왜 발생했는지를 설명하세요.

<참고>

- 주어진 withdraw 함수를 약간 변형해서 사용해도 됩니다 (sleep 함수 추가?)
- account, amount 이외에 추가로 인자를 전달해도 됩니다

터미널 화면을 캡처해서 본 문서에 첨부하세요. 그리고 설명을 추가하세요(어떤 상황에서 어떤 문제가 발생하는지를 설명). 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q02$ ./withdraw
[user2] saving: 500, balance: 200, amount: 300
[user1] saving: 200, balance: 100, amount: 400
[user2] 인 출 성 공
[user1] 인 출 성 공
yeongju@vm-ubuntu20:~/sp2021-2/w11/q02$
```

**【설명】** 두 thread 가 동시에 접근하여 공유자원(saving)에 대해 동기화가 안 이루어져 두 thread 가 인출된 경우의 오류이다.

### [Q 3] 스레드 + 현금인출 + 뮤텝스 [18 점]

강의자료 p.32 [Pthread: 뮤텝스 (mutex) 예제]를 참고해서, [Q 2] 에서 발생한 경쟁조건 문제를 해결하세요. 터미널 화면을 캡처해서 본 문서에 첨부하세요 (터미널 출력

결과를 통해서 오류상황이 발생하지 않았다는 것을 알 수 있어야 합니다). 어떻게 [Q2]에서 발생한 문제를 해결했는지 설명하세요. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q03$ ./withdrawMutex
[user2] saving: 500, balance: 200, amount: 300
[user2] 인출 성공
[user1] saving: 200, balance: 200, amount: 400
[user1] 잔액 부족
yeongju@vm-ubuntu20:~/sp2021-2/w11/q03$
```

**[설명]** mutex 를 사용하여 하나의 thread 가 임계영역에 들어가면 lock 을 걸어 또 다른 thread 가 처음에 들어간 thread 가 임계영역에서 나와 unlock 이 될 때까지 lock 걸린 부분에서 대기한다. 이후 unlock 이 되면 대기하던 thread 가 임계영역에 진입한다.

#### [Q 4] 데드락 [18 점]

최소 2 개 또는 그 이상의 스레드를 만들고, 데드락 (Deadlock)이 발생하도록 프로그램을 작성하시오. 터미널 출력 화면을 캡처 후 첨부하시오. 데드락이 왜 발생하는지 설명하시오. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q04$ ./deadlock
[Thing 2] World
[Thing 1] Hello
█
```

**[설명]** 두 thread 가 각각 mutex1, mutex2 사용하고 있는 상태에서 mutex1 을 사용하고 있는 thread 는 mutex2 의 unlock 를 기다리고 있고, mutex2 를 사용하고 있는 thread 는 반대로 mutex1 의 unlock 를 기다리고 있는 상황에서 교착상태가 발생했다.

**[Q 5] 데드락 + 뮤텍스 [18 점]**

위의 [Q 4] 에서 작성한 프로그램에 뮤텍스를 추가하여 데드락을 방지하세요. 터미널 출력 화면을 캡처 후 첨부하시오. 데드락이 어떻게 해소 되었는지 설명하시오. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w11/q05$ ./deadlockMutex
[Thing 2] World
[Thing 1] start sleep...
[Thing 2] Hello
[Thing 1] end sleep!!!
[Thing 1] World
[Thing 1] Hello
yeongju@vm-ubuntu20:~/sp2021-2/w11/q05$
```

**[설명]** 하나의 thread1 가 사용하고 있는 mutex1 을 잠시 unlock 시키고 추가한 mutex 를 사용하는 동안 또 다른 thread2 가 mutex1 을 사용하고 나간 후, thread1 은 추가한 mutex 를 다 사용하고 다시 기존에 mutex1 을 lock 하여 마저 사용한다.

끝! 수고하셨습니다 ☺