

# Homework #4

[ECE30021] Operating Systems

# Mission



## ■ Solve problem 1, 2, and 3

- Solve the problems on Ubuntu Linux (VirtualBox or UTM) using vim and gcc.
- Problem 1: discussion within study group and using AI are allowed
- Problems 2 and 3: for individual students (discussion and use of Internet/AI are prohibited)

## ■ Submission

- Submit a .tgz file containing hw4\_1.c, hw4\_2.c and hw4\_3.c on LMS.  
“tar cvfz hw4\_<student\_id>.tgz hw4\_1.c hw4\_2.c hw4\_3.c”  
(Do not copy&past but type the above command manually!)
- After compression, please check the .tgz file by decompressing in an empty directory.  
“tar xvfz hw4\_<student\_id>.tgz”

## ■ Due date: PM 11:00:00, May 17<sup>th</sup>

# Honor Code Guidelines

## ■ “과제”

- 과제는 교과과정의 내용을 소화하여 실질적인 활용 능력을 갖추기 위한 교육활동이다. 학생은 모든 과제를 정직하고 성실하게 수행함으로써 과제에 의도된 지식과 기술을 얻기 위해 최선을 다해야 한다.
- 제출된 과제물은 성적 평가에 반영되므로 공식적으로 허용되지 않은 자료나 도움을 획득, 활용, 요구, 제공하는 것을 포함하여 평가의 공정성에 영향을 미치는 모든 형태의 부정행위는 단호히 거부해야 한다.
- 수업 내용, 공지된 지식 및 정보, 또는 과제의 요구를 이해하기 위하여 동료의 도움을 받는 것은 부정행위에 포함되지 않는다. 그러나, 과제를 해결하기 위한 모든 과정은 반드시 스스로의 힘으로 수행해야 한다.
- 담당교수가 명시적으로 허락한 경우를 제외하고 다른 사람, 또는 AI가 작성하였거나 인터넷 등에서 획득한 과제물, 또는 프로그램 코드의 일부, 또는 전체를 이용하는 것은 부정행위에 해당한다.
- 자신의 과제물을 타인에게 보여주거나 빌려주는 것은 공정한 평가를 방해하고, 해당 학생의 학업 성취를 저해하는 부정행위에 해당한다.
- 팀 과제가 아닌 경우 두 명 이상이 함께 과제를 수행하여 이를 개별적으로 제출하는 것은 부정행위에 해당한다.
- 스스로 많은 노력을 한 후에도 버그나 문제점을 파악하지 못하여 동료의 도움을 받는 경우도 단순한 문법적 오류에 그쳐야 한다. 과제가 요구하는 design, logic, algorithm의 작성에 있어서 담당교수, TA, tutor 이외에 다른 사람의 도움을 받는 것은 부정행위에 해당한다.
- 서로 다른 학생이 제출한 제출물간 유사도가 통상적으로 발생할 수 있는 정도를 크게 넘어서는 경우, 또는 자신이 제출한 과제물에 대하여 구체적인 설명을 하지 못하는 경우에는 부정행위로 의심받거나 판정될 수 있다.

# Problem 0

---



- Search the Internet for the following functions and learn how to use them to guarantee mutual exclusion.
  - `int pthread_mutex_lock(pthread_mutex_t *mutex);`
  - `int pthread_mutex_trylock(pthread_mutex_t *mutex);`
  - `int pthread_mutex_unlock(pthread_mutex_t *mutex);`
- Read the following functions in `Console.c` and learn the synchronization techniques used for them
  - `LockDisplay()`, `UnlockDisplay()`
  - `ExclusivePrintf()`, `PrintXY()`, `LogMessageF()`
- You may discuss with your teammates or use AI.

# Problem 0

- Read the following code carefully and learn the techniques used for it.

```
int SplitArgs(char *input, char *args[], int max_arg)
{
    int len = strlen(input);
    int no_arg = 0;

    for(int i = 0; i < len; i++){
        while(isspace(input[i]))
            i++;

        if(input[i] == 0)
            break;

        args[no_arg++] = &input[i];

        for(; i < len && !isspace(input[i]); i++){
#ifdef BONUS_PROBLEM
            if(input[i] == 'W'){
                strcpy(&input[i], &input[i + 1]); // delete W
                // not decrease i to treat the next character
                // as a normal character
                len--;
            } else if(input[i] == 'W' || input[i] == 'W'){
                char quote = input[i]; // opening quote
                strcpy(&input[i], &input[i + 1]); // delete it

                i--;
                len--;

                while(i < len && input[i] != quote)
                    i++;

                if(input[i] == quote){
                    strcpy(&input[i], &input[i + 1]); // delete quote
                    i--;
                    len--;
                }
            }
#endif // BONUS_PROBLEM
        }

        input[i] = 0;

        if(no_arg >= max_arg - 1) // -1 is for NULL
            break;
    }

    args[no_arg] = NULL;

    return no_arg;
}
```

# Problem 1



- **Mission: peer review of the midterm problems as a group**
  - Review the solutions of other members and share comments to each other.
  - Each member update her/his solutions individually applying the comments.
    - Name the updated solution to Problem 2 as hw4\_1\_<student\_ID>.c
    - Mark the modified parts by comments as the next page.
    - If you believe your previous solution was perfect, submit it again and put a comment “My previous solution was perfect, so I didn't modify it!” in the first line.
  - In hw4\_1\_<student\_ID>.c, write the advices you gave to your group mates as comments. (Hangul comments are okay.)
    - /\* Examples of comments to other members
      - Comments to other\_member\_1:
        - Line ##: This line is buggy. Fix in this way...
        - Line ##: This line can be rewritten more efficiently like ...
        - ...
      - Comments to other\_member\_2:
        - Line ##: Initialize the variable i
        - Line ##: Use increase the size of aaaa\_array

\*/

# Problem 1

---



- Example of updated solution.

```
int main()
{
    int i = 0;

    // previous code
    /*
    <old code>

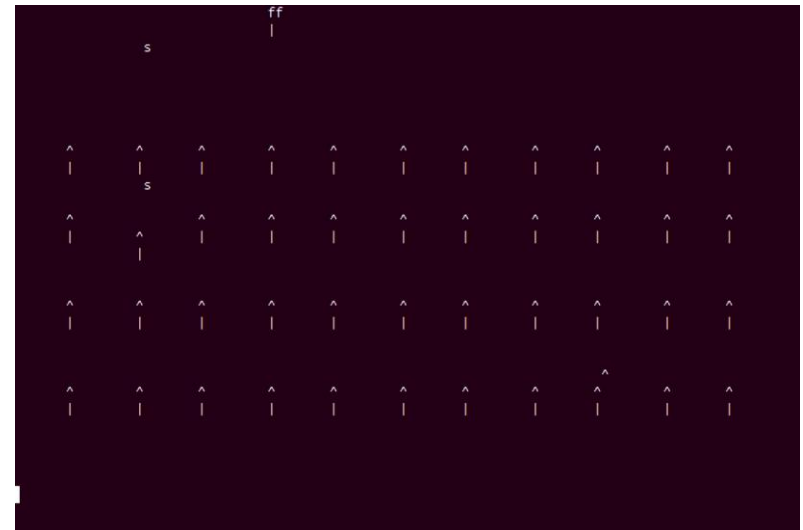
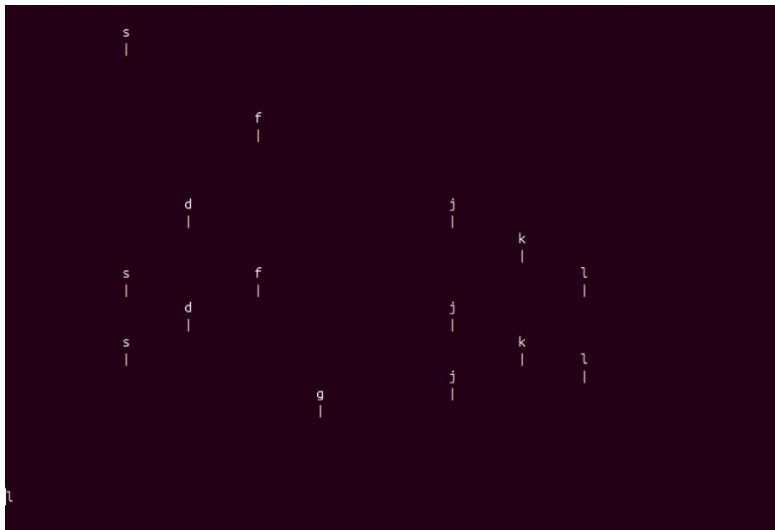
    */

    // updated code
    <new code>

    return 0;
}
```

# Problem 2

- Write a multi-threaded program that shoots missiles whenever the user presses a key
  - See hw4\_2\_demo.mp4.
  - Complete hw4\_2\_problem.c following instructions.
  - If the user types a character in "asdfghjkl;", shoot a missile at the corresponding horizontal coordinate, as the left figure.
  - If the user presses space, shoot all missiles, as the right figure.
  - If the user presses ESC, display a good bye message and terminate.





# Problem 2

---



## ■ Enabling and disabling synchronization

- Defining a macro SYNCHRONIZE enables synchronization
  - Put “#define SYNCHRONIZE” in the source file
  - Or compile with “-D SYNCHRONIZE” option

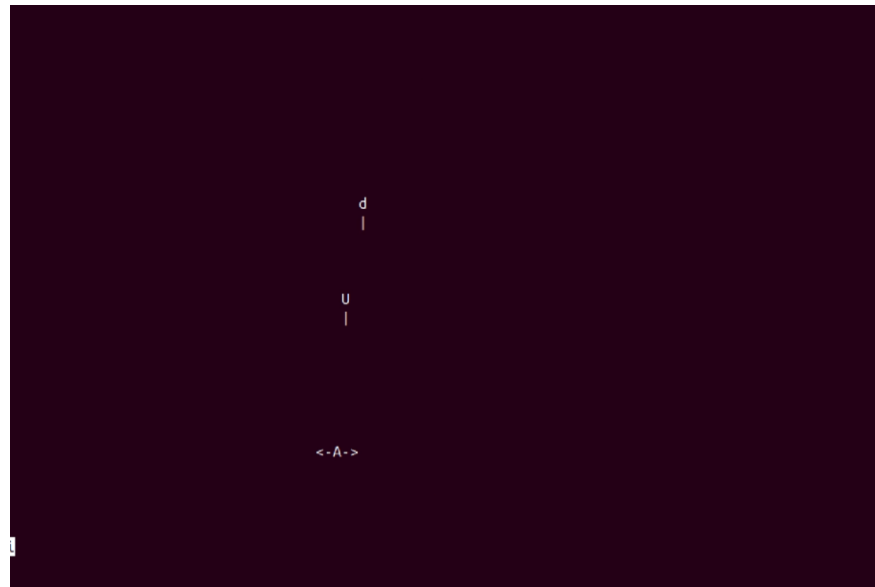
## ■ Compile commands

- Without synchronization option
  - gcc hw4\_2.c Console.c -o hw4\_2\_no\_sync
  - Press multiple keys rapidly and observe the problem.
- With synchronization option
  - gcc hw4\_2.c Console.c -D SYNCHRONIZE -o hw4\_2\_sync
  - Press multiple keys rapidly and see if the problem was resolved.

# Problem 3



- Write a multi-threaded program that moves a fighter
  - See hw4\_3\_demo.mp4
  - Complete hw4\_3\_problem.c following instructions.
  - Movement: j: left, l: right, i: up, k: bottom
    - Other keys stop the fighter
  - Missile launch: space
  - Terminate: ESC



# Problem 3

---



## ■ Enabling and disabling synchronization

- Defining a macro SYNCHRONIZE enables synchronization
  - Put “#define SYNCHRONIZE” in the source file
  - Or compile with “-D SYNCHRONIZE” option

## ■ Compile commands

- Without synchronization option
  - gcc hw4\_3.c Console.c -o hw4\_3\_no\_sync
  - Press multiple keys rapidly and observe the problem.
- With synchronization option
  - gcc hw4\_3.c Console.c -D SYNCHRONIZE -o hw4\_3\_sync
  - Press multiple keys rapidly and see if the problem was resolved.