

# CS 3113 Intro to Operating Systems

Name and ID \_Kyumin Lee 113591341\_

## Homework #1

**Due date: 09/19/2023 at 11:59 pm**

**Instructions:**

**1) To complete assignment one, you need to read Chapters 1, and 2 of the textbook.**

**2) HW must be submitted on a typed pdf or word document.**

**You must do your work on your own.**

Q1. Give the definition of Operating System. (5 points)

Answer:

Operating system in my definition is the very bridge in between hardware and application programs, handing common functions, such as communicating with I/O devices, controlling and allocating resources.

Q2. What are the three main purposes of an operating system? (10 points)

Answer:

- 1: Providing a convenient and efficient environment for programmers to build applications without handling common functions to communication with hardware and for users to execute application programs.
- 2: Managing I/O devices and supervising user programs to prevent improper use and errors.
- 3: Fair and efficiently allocating different resources to solve problems.

Q3. What is an interrupt? (5 points)

Answer:

To simply put it, it's a signal that the hardware sends to the CPU.

Q4. Describe the actions taken by a kernel to context-switching between processes. (10 points)

Answer:

This is a Chapter 3 question, but

Interrupt triggered -> kernel saves the current process state -> kernel selects the next process  
-> kernel loads the next process state -> resumes execution -> kernel updates process  
accounting -> kernel returns from interrupt.

Q5. What are three advantages and one disadvantage of multiprocessor systems? (15 points)

Answer:

Advantages

- 1: increased throughput.
- 2: increased processing power.
- 3: increased fault tolerance.

Disadvantage

Increased cost and complexity.

Q6. What are two sets of operating system services that provide functions helpful to the user and ensure the efficient operation of the operating system itself, respectively? Briefly describe those two sets. (15 points)

Answer:

Services helpful to the user:

1: User Interface

UI lets users to interact with the computer in many ways.

Terminal, GUI, and Multitouch interface are some examples.

2: I/O Operations

OS hides the details of I/O hardware.

Services ensure the efficient operation of the OS:

1: Resource allocation

When many different users or jobs running at the same time, resources get allocated to each of them.

2: Accounting

Keeps the track of each user's usage of computer resources.

Q7. Programming assignment: You will write a C program (60-100 lines) that satisfies the following criteria. Please compile, run your program on one of gpe machines, and attach a screen shot of the execution result (40 points).

- REMINDER: If your code cannot be compiled, you will get 0 point.
- This assignment covers the basic C coding techniques. (Hint: please do not forget to include corresponding header files.)
- **[Arrays and pointers]** Declare two integer arrays with the same size using pointers and dynamic memory allocation. The size should be determined by a user's input.
- **[Pointer]** Print the address of the first array.
- **[Address space]** Try the operator *sizeof* and print size of the first array pointer. Compare with the above one and think about the reason.
- **[Loop and array]** Use the array size provided by the user and assign an integer value to the array's element. Again, each integer value must be provided by the user. Repeat the procedure to fill both arrays we declared.
- **[Declaring a function]** Declare a function with two integer parameters, which calculates and returns the multiple of the two numbers.
- **[Calculation, condition statements, and file write]** Using the function we declared, multiply i-th element of the first array by i-th element of the second array. For each result, check if the multiplication result is an even or odd number using condition statements. Declare a file pointer and write the multiplication results in the new file named "hw1\_output.txt". Once the file write is done, do not forget to close the file.
- **[File read]** Open "hw1\_output.txt" file in your program, read, and print the content of the file. Again, please close the file.

Example output:

```
Start program.
Input array size:3
Address of arr1 after allocation: 0x55894ddfd80
Size of pointer arr1: 8
Input content of arr1:
Enter value of arr1[0]:1
Enter value of arr1[1]:2
Enter value of arr1[2]:3
Input content of arr2:
Enter value of arr2[0]:3
Enter value of arr2[1]:3
Enter value of arr2[2]:3
Multiplication start.
Multiplication done.
Read file.
arr[0] * arr2[0] = 3

        3 is an odd number.

arr[1] * arr2[1] = 6

        6 is an even number.

arr[2] * arr2[2] = 9

        9 is an odd number.

End program.
```

Answer:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
/*
```

```
[Declaring a function]
```

```
Declare a function with two integer parameters,  
which calculates and returns the multiple of the two numbers.
```

```
*/
```

```
int multiply(int num1, int num2);
```

```
int main() {
```

```
    int userArrSize;
```

```
    printf("Start program.\n");
```

```
    printf("Input array size: ");
```

```
    scanf("%d", &userArrSize);
```

```
    printf("\n");
```

```

/*
[Arrays and pointers]
Declare two integer arrays with the same size
using pointers and dynamic memory allocation.
The size should be determined by a user's input.
*/
    int *arr1 = (int *)malloc(userArrSize * sizeof(int));
    if (arr1 == NULL) {
        printf("Memory allocation failed for arr1.\n");
        return 1;
    }

    int *arr2 = (int *)malloc(userArrSize * sizeof(int));
    if (arr2 == NULL) {
        printf("Memory allocation failed for arr2.\n");
        return 1;
    }

/*
[Pointer]
Print the address of the first array.
*/
    printf("Address of arr1 after allocation: %p", (void *)arr1);

/*
[Address space]
Try the operator sizeof and print size of the first array pointer.
Compare with the above one and think about the reason.
*/
    printf("Size of pointer arr1:%zu", sizeof(arr1));

/*
[Loop and array]
Use the array size provided by the user and
assign an integer value to the array's element.
Again, each integer value must be provided by the user.
Repeat the procedure to fill both arrays we declared.
*/
    printf("Input content of arr1:\n");
    for(int i = 0; i < userArrSize; i++) {
        printf("Enter value of arr1[%d]: ", i);
        scanf("%d", &arr1[i]);
        printf("\n");
    }
    printf("Input content of arr2:\n");

```

```

for(int i = 0; i < userArrSize; i++) {
    printf("Enter value of arr2[%d]: ", i);
    scanf("%d", &arr2[i]);
    printf("\n");
}

printf("Multiplication start.\n");

printf("Multiplication done.\n");

/*
[Calculation, condition statements, and file write]
Using the function we declared,
multiply i-th element of the first array by i-th element of the second array.
For each result, check if the multiplication result is an even or odd number
using condition statements.
Declare a file pointer and write the multiplication results in the new file
named "hw1_output.txt".
Once the file write is done,
do not forget to close the file.
*/
FILE *file = fopen("hw1_output.txt", "w");
if (file == NULL) {
    printf("Error opening the file.\n");
    free(arr1);
    free(arr2);
    return 1;
}

for (int i = 0; i < userArrSize; i++) {
    int result = multiply(arr1[i], arr2[i]);
    const char *evenOdd = (result % 2 == 0) ? "even" : "odd";
    fprintf(file, "arr[%d] * arr2[%d] = %d\n\n", i, i, result);
    fprintf(file, "\t\t%d is an %s numer\n\n", result, evenOdd);
}

fclose(file);

/*
[File read]
Open "hw1_output.txt" file in your program,
read, and print the content of the file. Again,
please close the file.
*/

```

```
printf("Read file.\n");

file = fopen("hw1_output.txt", "r");
if (file == NULL) {
    printf("Error opening the file.\n");
    return 1;
}

char character;
while ((character = fgetc(file)) != EOF) {
    putchar(character);
}

fclose(file);

free(arr1);
free(arr2);

return 0;

}

int multiply(int num1, int num2) {
    return num1 * num2;
}
```

```
[lee0436@gpel8:~/homework1$ nano homework1.c
[lee0436@gpel8:~/homework1$ gcc homework1.c -o homework1
[lee0436@gpel8:~/homework1$ ./homework1
Start program.
Input array size: 3

Address of arr1 after allocation: 0x55b530a16a80
Size of pointer arr1:8
Input content of arr1:
[Enter value of arr1[0]: 1

[Enter value of arr1[1]: 2

[Enter value of arr1[2]: 3

Input content of arr2:
[Enter value of arr2[0]: 3

[Enter value of arr2[1]: 3

[Enter value of arr2[2]: 3

Multiplication start.
Multiplication done.
Read file.
arr[0] * arr2[0] = 3

           3 is an odd numer

arr[1] * arr2[1] = 6

           6 is an even numer

arr[2] * arr2[2] = 9

           9 is an odd numer

lee0436@gpel8:~/homework1$
```