CSE 3033 – Operating Systems (Fall 2020)
Midterm Exam (100 Minutes)

HONOR CODE:
"On my honor, I have neither given nor received any unauthorized and/or inappropriate assistance for this exam. The work done on this exam is totally my own. I understand that by the school code, violation of these principles will lead to a zero grade and is subject to harsh discipline issues."

Notes:

- **Provide handwritten solutions on blank A4 sheets.**
- **Read, accept and write the HONOR CODE inside the frame above to the first solution sheet, write the today's date and sign the sheet.**
- **Exam solutions without the HONOR CODE and a valid signature will not be accepted.**
- **Write your student ID, first name and surname on top of each solution page.**
- **During the exam, you may use textbooks, course slides and lecture notes.**
- **During the exam, Internet access except during the exam download and submission periods is NOT allowed.**
- **Show all your work.**
- **If you make any assumptions in the solutions, please state them clearly.**
- **Scan all the solution pages to a single PDF file named as "FirstName_LastName.pdf" and upload the pdf file to UES (ues.marmara.edu.tr) before the exam finish time.**
- **Unreadable exams will not be graded and get zero grade. Please check your pdf file before submission for readability.**
- **This exam consists of 5 questions in 6 pages. Duration of the exam is 100 minutes. You also have additional 20 minutes for uploading the solutions file.**
- **In case you could not upload your solutions file to UES (if UES does not work), you should send your solutions file to haydar.ozer@marmara.edu.tr before the exam deadline.**

Problem 1. (30 points)

Note that each <u>totally irrelevant</u> answer will get -2 points!

a-) (10 pts) What are the two fundamental hardware mechanisms required for a preemptive multitasking system? Explain why they are necessary.

b-) (10 pts) What is the primary difference between Von Neumann architecture (stored program computer) and ENIAC? What is the advantage and the disadvantage of this difference?

c-) (10 pts) Which parts (segments) of the address space of a process are not shared among threads of that process? Why is it necessary to use these segments different for every thread?

Problem 2. (30 points)

Consider the workload given below:

Process	**Burst Time**	**Arrival Time**
P_1	7 ms	0
P_2	4 ms	1
P_3	1 ms	3
P_4	2 ms	5

a. (10 points) Draw their **execution sequence** on a Gantt chart using **Preemptive Shortest Job First (Preemptive SJF)** scheduling policy considering the process arrival times. Show all your work.

b. (10 points) Draw their **execution sequence** on a Gantt chart using **Round-Robin (RR)** scheduling policy with a time quantum of 2 milliseconds considering the process arrival times. Draw the contents of the ready queue after each quantum. Show all your work.

c. (10 points) Can we implement and use the Preemptive Shortest Job First (Preemptive SJF) algorithm in a multitasking operating system? Explain?

Problem 3. (15 points)

Which segments of the process address space are used by the following program throughout its execution? Briefly state what the corresponding segments contain for the following code (do not give general answers where possible, try to be specific based on the following code).

```c
#include <stdio.h>
#include <stdlib.h>
int z1 = 1;
int *z2;
int z3;

int f1(int);
int f2(int);

int main(void) {
  int z1 = 3;
  z2 = (int*)
malloc(z1*sizeof(int));
  z2[z1-1] = f2(z1);
  printf("%d",z2[z1-1]);
  return 0; }
```

```c
int f1(int x) {
  if (x == 0) {
    return 1; }
  z2 = (int *) calloc (x,sizeof(int));
  printf("%d",z2[0]);
  return f2(x+1); }

int f2(int x) {
  int i, y = 0;
  for(i = 0; i < 2; i++) {
    static int h = 2;
    h *= 2;
    y += h; }
  y = f1(x-2);
  return y; }
```

Problem 4. (12 points) (Lab - Shell Script)

a) **(2 points)** What is the functionality of the command cp `../*.c ../*.h ./files`.

b) **(3 points)** Write a command to give “**`rw-r-x--x`**” permissions to all the files whose name ends with ~ under a sub directory of current working directory named sub_dir **using wildcards and relative pathnames**.

c) (**2 points)** Write a command that prints all arguments given to a shell script.

d) **(2 points)** Write a command that prints the manual of the **`find`** command into a file named find.txt
Hint: You need to use **`man`** command.

e) **(3 points)** Write a command to count the contents of current working directory.
Hint: “wc” command counts the character with –c, the lines with –l, and the words with –w option.

Problem 5. (13 points) (Lab - Process)

Assume that the following program is free from *run-time* and *compile time* errors (i.e., all functions execute successfully).

```
int main() {
      int counter = 0;
      pid_t pid;
      if( !(pid = fork()) ) {
            while((counter < 2) && (pid = fork()) ){
                  counter++;
                  printf("%d \n", counter);
            }
            if (counter > 0) {
                  execlp("who", "who", NULL); //Assume prints USER1
            }
      }
      if(pid) {
            waitpid(pid, NULL, 0);
            counter = counter * 2;
            printf("%d \n", counter);
      }
      printf("%d \n", counter);
}
```

a) **(9 points)** Draw the relationship diagram between the processes. Give a process id to each process. Show how the variables change for each process.

b) **(4 points)** Trace the program segment and write a possible output. Justify your answer using diagram above. **(You will not get any grade from this part of the question if part (a) above is empty!)**