COMP 2432 Operating Systems
Mid-term
Date: 18 March 2023

This mid-term test constitutes **20%** of the grade. Answer all questions. The time allowed is **90 minutes**.

1. I/O and Operating Systems (15 points).
Draw a diagram to differentiate between *synchronous* I/O and *asynchronous* I/O. ***Name*** the different types of *system programs*.

2. CPU Scheduling (25 points).
Draw *Gantt charts* for the following set of processes using different scheduling algorithms, (*a*) **SRT**, and (*b*) **RR** with quantum 3. ***Compute*** the *waiting time* and *turnaround time* for each process by filling in the table. If you find these to be too hard, you may try the easier non-preemptive version, i.e. (*a*) with **SJF**, and (*b*) with **FCFS**, but you can only receive *half of the score* if you select the easier **SJF** or **FCFS**. (20 points)

Process	*Burst Time*	*Arrival Time*	*Priority*	(*a*)		(*b*)	
				Waiting time	Turnaround	Waiting time	Turnaround
P_1	9	0	3				
P_2	7	1	2				
P_3	3	3	4				
P_4	7	5	1				
P_5	2	8	5				

(*c*) Consider the following set of processes being scheduled based on an *unknown non-preemptive* algorithm *X* and there is a context switching overhead of 1 time unit. ***Determine*** the *moment of time* when all the processes complete their execution. Working steps are optional. (5 points)

Process	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}
Burst time	9	7	3	7	2	9	7	3	7	2	9	7	3	7	2	9
Arrival time	0	1	3	5	8	10	11	13	15	18	20	21	23	25	28	30
Priority	3	2	4	1	5	3	2	4	1	5	3	2	4	1	5	3

3. Process Management (15 points).
Draw a *state-transition diagram* for the five states of a process as it is inside the system. ***Distinguish*** between a *zombie process* and an *orphan process*. In Linux, orphan processes are *adopted* by a special process. ***Who*** is that special process?

4. Memory Management (20 points).
Consider a memory of total size of 500K managed using IBM MVS ***MVT*** approach. After a sequence of memory requests, there is a list of 3 holes of size 70, 100, and 60K. You are given 6 requests: 50, 49, 29, 45, 30, and 22K, arriving in that order. ***Indicate*** how the requests are satisfied with (*a*) **first-fit**, (*b*) **best-fit**, and (*c*) **worst-fit** algorithm. ***Determine*** the *utilization* for ***any one*** of the three algorithms. (*d*) ***Show*** how you could satisfy all requests if you see all of them in advance. (15 points)

(*e*) Consider a computer system with 8KB main memory and frame size of 256 bytes, with a logical address length of 12 bits. The logical address 010011010010 is generated by the CPU for a user process, with Page Table Base Register value of 0101010101010. Given the page table content below, ***determine*** the *physical address* in the main memory for the CPU generated logical address 010011010010. Working steps are optional. (5 points)

Page	Frame number
0	10001
1	00010
2	10110
3	00111
4	10011
5	00100

5. Programming with Processes (10 points).
Assuming that other statements are correctly written (e.g., definition of variables) to convert the following program fragment into a full executable program, ***how many*** *processes* are created altogether? ***List*** all the possible output sequences that could have been generated from this program.

```
pipe(fd); strcpy(msg,"hello");
p = fork();
if (p > 0) strcpy(msg,"bye"); else strcpy(msg,"welcome");
if (p == 0) { len = write(fd[1],msg,strlen(msg)); p = fork(); }
if (p == 0) { len = read(fd[0],msg,80); msg[len] = 0; }
printf("%s\n",msg);
if (p > 0) fork();
```

6. Shell Programming (15 points).

We have developed a bash shell script to generate the transcripts for students in our lab exercise. Consider the situation that all the subject files were erased by someone accidentally executing "**`rm *`**" on Linux. It is now necessary to generate the *grade distribution* for some subjects.

Write a *bash script program* called **`grade.sh`** to recover the *grade distribution* of a specific subject in an offering. The subject and the year of offering are given as arguments to the program. The following shows some sample transcript files.

trans1223.txt	**trans1234.txt**	**trans1236.txt**	**trans1238.txt**
Transcript 1223 bob **COMP1011 2021 Sem 2 F** **COMP1011 2022 Sem 1 B** **COMP2411 2022 Sem 1 C+** **GPA for 2 subjects 2.65**	**Transcript 1234 john** **COMP1011 2021 Sem 2 B** **COMP2411 2022 Sem 1 B-** **GPA for 2 subjects 2.85**	**Transcript 1236 peter** **COMP1011 2021 Sem 2 A** **COMP2411 2022 Sem 1 A** **GPA for 2 subjects 4.00**	**Transcript 1238 alice** **COMP1011 2021 Sem 2 C+** **COMP2411 2022 Sem 1 A** **GPA for 2 subjects 3.15**

Sample program execution based on the transcript files and outputs are shown.

grade.sh COMP1011 2022	**grade.sh COMP1011 2021**	**grade.sh COMP2411 2022**	**grade.sh COMP2411 2021**
COMP1011 in 2022 **B: 1**	**COMP1011 in 2021** **A: 1** **B: 1** **C+: 1** **F: 1**	**COMP2411 in 2022** **A: 2** **B-: 1** **C+: 1**	**COMP2411 in 2021** **No student in subject**

You could assume that all subjects taken by a student bear a grade (i.e. subjects currently taking are not shown) and that all subjects are offered only at most once per year. You can assume that there is no error in the input transcript files.