Birla Institute of Technology & Science Pilani, Hyderabad Campus

Second Semester 2019-2020

CS F372: Operating Systems

Mid Semester Examination (Regular)

Time: 90 minutes **Max Marks: 105** **Date: 07/03/2020**

Instructions:

1. Answer all questions. All parts of the same question should be answered together.
2. There are total 3 pages in the question paper. Answer the questions according to the marks allotted.
3. Calculators are allowed.

Q1. Write short notes on the following **(5+5+5+5)**

(a) Micro kernel
(b) Layered kernel
(c) Zombie process
(d) Orphan process

Q2. Consider two computer systems X and Y having different architectures. These systems are having two different operating systems OS-X and OS-Y running on them, respectively. Both operating systems are POSIX compliant. Does the source code of a user application which is written to run on computer system X must always be rewritten to run on computer system Y. Answer using Yes or No and give the justification accordingly. **(2.5+2.5)**

Q3. Consider the following three processes that arrive in a system at the specified times, along with the duration of their CPU bursts. **(10+10)**

Process P1 arrives at time t=0, and has a CPU burst of 10 time units. P2 arrives at t=2, and has a CPU burst of 2 units. P3 arrives at t=3, and has a CPU burst of 3 units. Assume that the processes execute only once for the duration of their CPU burst, and terminate immediately. **Calculate the time of completion of the three processes under each of the following scheduling policies. For each policy, you must state the completion time of all three processes, P1, P2, and P3.** Assume there are no other processes in the scheduler's queue.

For the pre-emptive policy, assume that a running process can be immediately pre-empted as soon as the new process arrives (if the policy should decide to pre-empt).

1. First Come First Serve
2. Round robin (pre-emptive) with a time slice of 5 units per process

Q4. Consider the following program and find out how many processes are created excluding the initial parent process? **(5)**

```
#include<stdio.h>
#include<unistd.h>
int main(){
fork();
fork();
return 0;
}
```

Q5. Explain the difference between preemptive and nonpreemptive scheduling? **(5)**

Q6. Many CPU-scheduling algorithms are parameterized. For example, the RR algorithm requires a parameter to indicate the time slice. Multilevel Queue scheduling algorithm requires queue to move processes between queues, and so on.

These algorithms are thus really sets of algorithms (for example, the set of RR algorithms for all time slices, and so on). **One set of algorithms may include another (for example, the FCFS algorithm is the RR algorithm with an infinite time quantum).** Similarly, what (if any exists) relation holds between the following pairs of algorithm sets? **(5+5+5)**

a. Priority and SJF

b. Priority and FCFS

c. RR and SJF

Q7: What resources are used when a thread is created? How do they differ from those used when a process is created? **(5)**

Q8: Consider a parent process P that has forked a child process C in the program below.

```
int a = 5;
int fd = open(...) //opening a file
int ret = fork();
if(ret >0) {
close(fd);
 a = 6;
 ...
 }
else if(ret==0) {
printf("a=%d\n", a);
read(fd, something);
}
```

After the new process is forked, suppose that the parent process is scheduled first, before the child process. Once the parent resumes after fork, it closes the file descriptor and changes the value of a variable as shown above. Assume that the child process is scheduled for the first time only after the parent completes these two changes.

What is the value of **the variable a** as printed in the child process, when it is scheduled? Explain. **(10)**

Q9: Distinguish between PCS and SCS scheduling? **(5)**

Q10: Explain in brief the following scheduling criteria? **(2.5+2.5)**

(a) Turnaround Time
(b) Response Time

Q11: Starvation is a condition where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.

Consider the following scheduling algorithms and find which among these lead to the starvation? **(2.5+2.5+2.5+2.5)**

a. First-come, first-served

b. Shortest job first

c. Round robin

d. Priority