Extras

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Mid-Spring Semester 2017-18

Date of Examination: 19/02/2018 **Session: AN** **Duration: 2 hours** **Full Marks: 80**

Subject No.: CS30002 **Subject: Operating Systems**

Department: Computer Science and Engineering

Instructions: Answer all the questions

1. Briefly answer the following. **[8 × 2 = 16]**
   a) Which of FCFS, SJF and Round-Robin scheduling algorithms can lead to starvation?
   b) List two events that may take a process to the ready state.
   c) How many times does the following C program print "**Hello**"?

```
main() {
for (int i=0; i<8; i++) {
      if (i%2 == 0) break;
      fork();
}
printf ("Hello\n");
}
```

   d) Given that we can create user-level code to control access to critical sections (e.g. Peterson's algorithm), why is it important for an operating system to provide synchronization facilities such as semaphores in the kernel?
   e) How can the following pair of scheduling criteria conflict in certain settings: *I/O device utilization* and *CPU utilization*?
   f) When can a process enter the Zombie state? How can it come out of it?
   g) List two hardware supports that the OS designer require from the hardware designer of the computer system.
   h) Can an unnamed pipe in Unix created using the `pipe()` system call be used to communicate between any two processes in the system?

2. State with clear justifications whether the following statements are true or false. *No marks will be awarded if the justification is not correct.* **[8 × 2 = 16]**
   a) Operation of a time sharing system is identical to operation of a traditional multiprogramming system executing the same programs if the time slice exceeds the CPU burst of every program.
   b) The instruction to change the processor from user mode to supervisor mode is a privileged instruction.
   c) For multithreaded programming, we need to create shared memory segments through which the threads can access common data.
   d) Pre-emptive CPU scheduling algorithms can result in shorter average waiting time as compared to non pre-emptive scheduling algorithms.

e) The `signal(pid,sig)` function call can be used by a user-level process to send a designated signal to some other process.

f) Switching among threads in the same process is more efficient than switching among processes.

g) Local variables defined within a function cannot be used as shared data between a parent process and a child process, but can be used as a shared data between two execution threads of a process.

h) Binary semaphores cannot be used by more than two processes. For more than two processes, we need counting semaphores.

3. Answer the following. **[4 + 4 + 4 = 12]**

a) Consider a program consisting of 3 concurrent processes P0, P1 and P2 as shown below, and 3 binary semaphores that are initialized as: S0 = 1, S1 = 0; S2 = 0. How many times will process P0 print '0'?

Process P0	Process P1	Process P2

`while (true) {`
`  wait (S0);`
`  print '0';`
`  signal (S1);`
`  signal (S2);`
`}`
	`wait (S1);`
	`signal (S0);`
		`wait (S2);`
		`signal (S0);`

b) Consider the following solution for the Producer-Consumer problem that uses two binary semaphores **n** and **s**, initialized to 0 and 1 respectively. Comment on the correctness of the solution.

```
void producer()
{
  while (true) {
    produce();
    wait (s);
    addToBuffer();
    signal (s);
    signal (n);
  }
}
```

```
void consumer()
{
  while (true) {
    wait (s);
    wait (n);
    removeFromBuffer();
    signal (s);
    consume();
  }
}
```

c) Consider two processes P1 and P2 accessing the shared variables X and Y protected by two binary semaphores SX and SY respectively, both initialized to 1. Complete the entry and exit sections of the following codes such that the processes can update the shared variables correctly without deadlock.

```
P1: while (true) do {
      <entry section>
      X = X + 10;
      Y = Y - 20;
      <exit section>
}
```

```
P2: while (true) do {
      <entry section>
      Y = Y + 20;
      X = X - 10;
      <exit section>
}
```

4. Answer the following. **[5 + 5 + 5 + 5 = 20]**

a) Consider three processes, all arriving at time zero, with total execution times of 10, 20 and 30 units, respectively. Each process spends the first 20% of execution time doing I/O (reads the input data), the next 70% of time doing computation, and the last 10% of time doing I/O again (prints the results). The operating system uses a ***shortest remaining compute time first*** scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or

when the running process finishes its compute burst. Assume that *all I/O operations can be overlapped as much as possible.* For what percentage of time does the CPU remain idle?

b) The shortest job first (SJF) algorithm minimizes the average waiting time. Prove this for a set of **n** processes, which arrive at the same time with CPU burst times $t_1 \leq t_2 \leq \ldots \leq t_n$, ignoring further arrivals.

c) Consider a system with 100 processes in the Ready Queue. If round robin scheduling algorithm is used with a scheduling overhead of 5 microseconds, what must be the time quantum to ensure that each process is guaranteed to get its turn at the CPU after 1 second?

d) Consider the following CPU processes with arrival times (in milliseconds) and length of CPU bursts (in milliseconds) except for process P4 as given below:

Process	**P1**	**P2**	**P3**	**P4**
Arrival Time	0	1	3	4
CPU Burst Time	5	1	3	**Z**

If the average waiting time across all processes is 2 milliseconds and pre-emptive shortest remaining time first scheduling algorithm is used to schedule the processes, what will be the value of **Z**?

5. Answer the following. **[4 + 6 + 6 = 16]**

a) Define a binary semaphore. Suggest an implementation of binary semaphore that eliminates busy waiting in the entry section of the code.

b) Consider an extension to the MIPS instruction set where a register-memory swap instruction is added. For example, to exchange the contents of register R5 and the word stored in memory location 5000, we use:

```
SWAP  R5,0(5000)
```

Provide a solution to the critical section problem for arbitrary number of processes using this instruction.

c) What is the readers/writers problem? Using pseudo-code, provide a solution for the same using semaphores.