Department of Computer Science and Engineering, Indian Institute of Technology Palakkad

CS3110: Operating Systems Lab

5th September, 2019 **Viva prelims** **2:30pm - 3:00pm**

1. **(Stage 6)** Suppose you have put several breakpoints in your SPL and expl codes and when your program is run in debug mode, the debugger shows the mode as KERNEL and the page number as 8. How do you know the breakpoint in which SPL program is being executed now?

2. **(Stage 9)** What is the need for a Kernel Stack?

3. **(Stage 12)** Suggest modifications to the OS code so that no process is scheduled more than 10 times.

4. **(Stage 12)** Suppose we do the following modification in the timer interrupt routine, in stage 12.

   - Place 'backup' instruction as the first instruction in the timer interrupt routine, instead of the place where it is placed now.
   - Place the 'restore' instruction as the instruction just before the ireturn instruction for processes not in CREATED state.

   Suppose init process is the first process to be scheduled and just before the timer interrupt occured for the first time, the SP value was 5000. When timer interrupt routine runs for the first time, while executing the backup instruction, the register values will not be saved to the user stack of init. Where will it try to store the values instead? Why?

5. **(Stage 13)** We want to modify assignment 1 of stage 13 such that process with pid 2 is scheduled as the first user process to run after the machine starts. What changes are required (above assignment 1) in boot module and os startup code to implement this?

6. **(Stage 14)** When the control of execution is to be going out of any Module (like Boot Module or Scheduler module), usually we use a return statement and not an ireturn statement. Why? Point out one exception to this rule.

7. **(Stage 14)** Suppose you have implemented upto stage 14 and you have the standard idle program that just loops infinitely and an init program that prints numbers 1 to 100. From the time the machine first enters user mode and till the time the system halts, what are the different process states the init and idle processes are going through?

8. **(Stage 15)** Modules are for bringing in modularity to system design. Suppose you have implemented upto stage 15. Point out from which all SPL programmes the scheduler module is being invoked?

9. **(Stage 16)** Suppose you have implemented upto stage 16. You decided to change your idle program so that it prints numbers 1 to 50 and your init program is to read two numbers and print their GCD. There are no other processes. There is a possibility that after the machine starts running, the OS gets stuck in the scheduler module. How?

[END]