

Department of Electrical Engineering, IIT Delhi
EEL602 Operating Systems: Major Examination
 (Closed book/Closed Notes) Time: 2 hour Maximum Marks: 25

"Thou shalt not covet thy neighbour's answers"

1. **The sole soul of wit: Brevity. Brief answers only, please!**

- (a) Briefly explain 'copy-on-write'.
- (b) What is an atomic operation?
- (c) Describe a simple scheme to avoid a busy wait/spin lock in the implementation of a semaphore. Please mention only the main points.

(1+1+2 marks)

2. **No faults, please!** Consider the page size to be 100 bytes, and the following addresses referenced: 100, 101, 205, 318, 451, 132, 111, 208, 519, 119, 222, 226, 311, 382, 328, 403, 551.

- (a) Write down the page reference string.
- (b) Consider the FIFO page replacement strategy, and the number of frames in memory to be 3. Show the contents of the frames on each item in the page replacement string, and mark each page fault. How many page faults do you get?
- (c) Now, consider the number of frames to be 4. Repeat the above exercise.
- (d) What do you conclude?

(1+6+6+1 marks)

3. **Producing Consumerism!** Consider the following solution to the Producer-Consumer problem that uses no semaphores, mutexes, or any special concurrency control measures. `in`, `out` and `counter` are integer variables, and the variable `counter` is shared between the producer and the consumer processes (along with the shared array `buffer`, of course). All other variables and constants are self-explanatory.

<pre>/* --- Producer process --- */ while(TRUE) { while(counter==BUFFER_SIZE) ; buffer[in] = nextProduced; in = (in+1) % BUFFER_SIZE; counter++; }</pre>	<pre>/* --- Consumer process --- */ while(TRUE) { while(counter==0) ; nextConsumed = buffer[out]; out = (out+1) % BUFFER_SIZE; counter--; }</pre>
--	---

Suppose that `counter` is initially 5, and that the producer and consumer process execute concurrently. Illustrate three cases as to how the value of `counter` can be either 4, 5 or 6, respectively.

<p>Consider <code>counter++</code> to consist of the following assembly language statements:</p> <pre>r1 = counter r1 = r1 + 1 counter = r1</pre>	<p>Consider <code>counter--</code> to consist of the following assembly language statements:</p> <pre>r2 = counter r2 = r2 - 1 counter = r2</pre>
---	---

(2+1+2 marks)

4. **Answer ef-Fort-lessly!** The steep entrance road to the Gwalior fort in Central India allows a width of only one vehicle to pass through. The guards at the start and the end of this entrance road communicate with each other via telephone to allow at most one vehicle to pass through at a time, in either direction. Suppose you are to simulate this using a semaphore/mutex variable, `permission`.
- (a) Write pseudo-code for a process (in an infinite loop!) to be executed by a vehicle in any direction. Use the `wait()` and `signal()` functions.
 - (b) What will you initialise this semaphore/mutex to?
 - (c) What will the Critical section be? *(1+0.5+0.5 marks)*