Department of Electrical Engineering, IIT Delhi

EEL602 Operating Systems: Major Examination

(Closed book/Closed Notes) Time: 2 hours   Maximum Marks: 25

"Thou shalt not covet thy neighbour's answers"

1. **A hierarchy of irritating questions**

   (a) **No faults, please!** Consider the FIFO page replacement algorithm, and the following page reference string: 1 2 3 4 1 2 5 1 2 3 4 5. Assume that main memory has 3 frames. Suppose one is asked to show the contents of the frames on each item in the page reference string. What is wrong with the following solution (Apart from the fact that the page faults have not been marked, and that the page replacement string has not been mentioned at the top)?

1	2	3	4	1	2	5	1	2	3	4	5
1	2	3	4	1	2	5	5	5	3	4	4
	1	2	3	4	1	2	2	2	5	3	3
		1	2	3	4	1	1	1	2	5	5

   (b) **Filling too many pages will cause a page fault!** Consider a system with a 32-bit address bus, 512 MB main memory for paging (not for the OS, and the page table itself), and a page size of 4 KB. What will be the size of the page table, in case a single level page table is needed, and it is to be all stored in the memory? Now, consider a two-level page table, where this page table itself is to be paged. For the paging of the page table, each page is to have 4 K entries, and the main memory is to store only 4 of these. Show a diagram describing all entities, and their sizes. *(2+4 marks)*

2. **Paging for your attention!**

   (a) What is demand paging?

   (b) **No segmentation fault!** How does pure segmentation need more storage/memory than pure paging?

   (c) Why is the kernel not paged, by definition?

   (d) **Inverted Page Tables** Why should page sharing be problematic for an inverted page table representation?

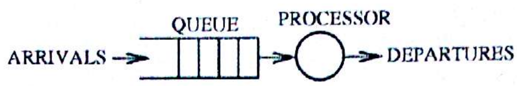
   (e) What are the two main reasons why kernel memory allocation is different from user-mode memory allocation?

   (f) What are the three possibilities/steps in slab allocation under Linux?

   (g) **Off-beat question** What is thrashing? *(1+1+1+2+2+3+1 marks)*

P. T. O.

3. **ICQ: I See Queue?** Consider a M/M/1 modeling of a Computer System - Poisson Zeroth Order Markovian Arrival Process, Exponential Service Time Distribution, and a Uniprocessor System. The processor handles jobs in a First-In-First-Out (FIFO) manner. Assume the arrival rate and processor service rate to be λ and μ jobs per unit time, respectively. Let



$P_n(t)$ denote the Arrival Process: probability of n jobs arriving during time t. Let $S(t)$ denote the Service Time Distribution: the probability that the service is not completed by time interval t. The Zeroth-Order Markovian assumption implies that there is no dependence of a job arrival on the arrival of any other job, and that at most one job can arrive in an infinitesimal interval of time δt. Similarly, at most one service can take place in time δt. Let $p_n(t)$ denote the probability of the system having n jobs at time t. Assume that the system can also have an infinite number of jobs.

(a) **Caustix Stochastix: The instructor's hand-waving** In the derivation for the arrival process being Poisson, the instructor first proved the case for $n = 0$. Convincing enough. Then, he set out to prove the same for $n = 1$, ostensibly taking the most difficult route possible, using one of the toughest ways to solve a differential equation i.e., using the homogeneous solution and working the general case. The instructor mentioned that there was some (intentional) hand-waving in his derivation at one specific place. What was it?

(b) **Encore: Independence from the course? Some Waiting time** Consider the derivation of the service time distribution. One of the core assumptions of the entire derivation was the independence assumption. For an interval of time $t+\delta t$, the independence assumption means that the probability of no service being completed in $t + \delta t$ is the product of the probability of no service being completed in t, and the probability of service not being completed in time δt. This should have troubled you. After all, given more time, should the probability of completion not increase? (Remember our worst examination fears: not enough time, and suddenly being faced with the prospect of some extra time). Enough said. Give an example of a job where increasing the time period does not increase the probability of completion.

(c) **Finally, being job-free** From *First Principles (ab initio)* find the steady state probability of the system being job-free. *(2+2+4 marks)*

$e^{-\mu t}$

Prob (service time $\geq t$) = $e^{-\mu t}$

$1 - [1 - \mu t \cdots]$

$[\mu t]$