Indian Institute of Engineering Science and Technology, Shibpur

Five year Dual Degree (B.Tech-M.Tech) 6^{th} Semester Final Examination 2019
Operating Systems (CS-601)

FULL MARKS: 70 TIME: 3 hrs.

Answer Question No. 1 and any four (4) from the rest. Students are advised to go through the questions carefully before answering. Clearly mention your assumptions, if any, while answering. Credits will be given to precise answer.

1. Answer any 7 questions. Justify your answer whenever necessary.

(a) Explain the use of set user id-bit.

(b) What is an incore inode?

(c) What is a Zombie process?

(d) Exaplain copy-on-write?

(e) Does inode of a file contain its name?

(f) What is thrashing?

(g) What is belady's anomaly?

(h) Explain the use of TLB.

(i) What is PTBR? [7× 2]

2. (a) Explain bakery algorithm and prove that it satisfies all three requirements (mutual Exclusion, progress, bounded waiting).

(b) Explain how is it possible for two processes to get the same token number in bakery algorithm.

(c) Suppose in the ongoing IPL T20 tournament few seats (say n) have been put up for sell online. You can book or cancel tickets online. Since number of people will try to book or cancel tickets at the same time, the booking and cancellation need to be synchronized. Suggest a solution to this requirement.

[5+4+5]

3. (a) Answer the following questions in the context of deadlock avoidance.

(i) Define safe sequence of processes.

(ii) Why is it sufficient to have one safe sequence for the state of a system to be safe?

(iii) Propose an algorithm to determine whether granting a request made by a process will leave the system in safe state or not. Calculate the running time of your proposed algorithm (Assume there are n processes and m resource types).

(b) Consider the following snapshot of a system and answer the following questions using the bankers algorithm:

Process	Allocation ABCD	Max ABCD	Available ABCD
P_0	0012	0012	1520
P_1	1000	1750	
P_2	1354	2356	
P_3	0632	0652	
P_4	0014	0656	

(i) What is the content of the matrix Need?

(ii) Is the system in a safe state?

(iii) If a request from process P_1 arrives for $(0, 4, 2, 0)$, can the request be granted immediately?

[(2+2+3)+(2+2+3)]

4. (a) Explain the difference between preemptive and nonpreemptive scheduling.

(b) Consider the following set of processes, with the length of the CPU burst given in milliseconds:

Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2) and calculate the average waiting time for each of the above scheduling algorithm.

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

(c) Consider a preemptive priority scheduling algorithm based on dynamically changing priorities. Larger priority numbers imply higher priority. When a process is waiting for the CPU (in the ready queue, but not running), its priority changes at a rate α. When it is running, its priority changes at a rate β. All processes are given a priority of 0 when they enter the ready queue. The parameters α and β can be set to give many different scheduling algorithms.

(i) What is the algorithm that results from $\beta > \alpha > 0$?

(ii) What is the algorithm that results from $\beta < \alpha < 0$? [3+6+5]

5. (a) Why does a process on a system with paging cannot access memory that it does not own? How could the operating system allow access to other memory?

(b) Why are segmentation and paging sometimes combined into one scheme?

(c) Consider a paging system with the page table stored in memory.

(i) If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?

(ii) If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds, if the entry is present.)

(d) Why Inverted Page Table is sometime used? Illustrate the method of using it. [3+2+5+4]

6. (a) Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p, and n distinct page numbers occur in it. Answer these questions for any page-replacement algorithms:

(i) What is a lower bound on the number of page faults?

(ii) What is an upper bound on the number of page faults?

(b) How LRU page replacement algorithm is implemented?

(c) Consider the following page reference string:
$7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.$
Assuming demand paging with three frames, how many page faults would occur for the following replacement algorithms?

(i) LRU replacement

(ii) FIFO replacement

(iii) Optimal replacement

(d) Discuss second chance page replacement algorithm. What would be the number of page faults in this algorithm for the above page reference string? [2+2+6+4]

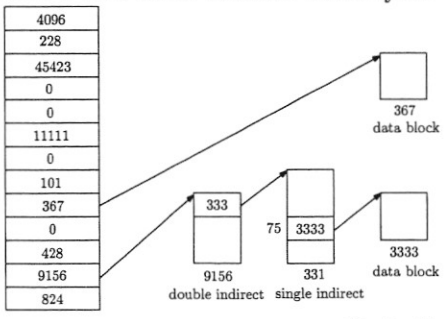
7. (a) Briefly explain how files are represented internally in a typical unix system.

(b) What do you mean by hard linked file and soft linked file? Explain their advantages and disadvantages.

(c) Consider the following block layout of a file and assume that a disk block contains 1024 bytes.

(i) If a process wants to access byte offset 9000 of the file, how does the kernel calculate which byte of which block actually contains the asked byte offset?

(ii) Similarly what will be the block number and byte number if the process wants to access byte offset 350, 000 of the file?



[2+6+6]