

SAMPLE PROBLEM LIST FOR SWE-234 LAB EXAM

SCOPE: PROCESS SCHEDULING

Problem#1 Write a program to find the average waiting time (AWT) and average turnaround time (ATT) for the following process scheduling algorithms. [Input the number of processes and their details of process bursts time and arrival time from the user.]

- ✓ SJF (With/Without Preemption)
- ✓ Priority (With/Without Preemption)
- ✓ Round Robin (Fixed/Varying Time Quanta)

Problem#2 You have to try for the following variants also: SRF, PRR, EDF

- ✓ Shortest Remaining First (**SRF**) : The preemptive version of SJF where the selection of any process from the ready queue depends on the minimum (additional) CPU burst to complete the task.
- ✓ Progressive Round-Robin (**PRR**) : A variation of the RR scheduler where the time-quantum is increased by 10% if a process does not complete within its entire time-quantum. When the time time-quantum increases by 100% and any process blocks before using its entire time quantum, the time quantum will be reduced to its default value again.
- ✓ Earliest-deadline-first (**EDF**) : A variant of Priority scheduling where a deadline is given for each process and this deadline is treated as the priority of that process - the earlier the deadline, the higher the priority; the later the deadline, the lower the priority.

SCOPE: DEADLOCKS

Problem#3 Write a program to implement Banker's Algorithm. Program must take a process request and compute if the request can be allocated I.e., compute state of the systems is safe or not, and grant the request if there exists any safe-sequence. You must consider the initial snapshot which may include the NUMBER_OF_PROCESSES, NUMBER_OF_RESOURCES, MAX_NEED, ALLOCATED_RESOURCES, etc.,

Problem#4 You have to try to implement the safety procedure using “Resource-Allocation-Graph” also.

SCOPE: **MEMORY MANAGEMENT**

Problem#5 Page replacement is basic to demand paging. It completes the separation between logical memory and physical memory. There are many different page-replacement algorithms and different benchmarks to evaluate them. You should implement LRU, Optimal, Second-Chance page replacement scheme and find their performance in terms of the page-fault. Code should read the number of free frames and a reference string.

Problem#6 You have to try for the following variants also: SRF, PRR, EDF

- ✓ Least Frequently Used (**LFU**) : This page-replacement algorithm requires that the page with the smallest count be replaced.
- ✓ Most Frequently Used (**MFU**) : This page-replacement algorithm is based on the argument that the page with the smallest count was probably just brought in and has yet to be used.

In the LAB, the task list may be LIKE or PARTIAL of these problem(s)