EE 4770 **Homework 4** **Due: 24 April 1998**

Problem 1: The table below describes tasks on an otherwise empty processor. Show the CPU activity and the task states from time zero until tasks E and F finish or until task B first runs, whichever is later. Assume that the scheduler is called at $t = 0$.

The system uses a two-level scheduler. The first round uses round-robin scheduling with class sequence $\{\alpha, \beta, \gamma, \beta\}$, class α will be chosen at time zero. (Note that β appears twice in the class sequence so class β tasks get twice as many opportunities to run.) Class α tasks are chosen using a priority scheme, class β are chosen using deadline scheduling, and γ are chosen using first come, first served. The quantum is 50 ms and the system is not task preemptive.

Task A computes for 15 ms then will wait (go into the wait state) for the clock to indicate the next multiple of 100 ms, at which time it will become ready again, it will repeat the process each time it runs. For example, if it finishes running at 237 ms it will go into a wait state until 300 ms.

Task Name	Round 1 Class	Round 2 Info	Arrival Time	Run Time	Activity
A	α	Prio 2	0 ms	∞	Repeats: 15 ms, wait until multp of 100 ms.
B	α	Prio 1	0 ms	200 ms	
C	β	235 ms deadline	215 ms	10 ms	
D	β	265 ms deadline	205 ms	30 ms	
E	γ	FCFS	5 ms	70 ms	
F	γ	FCFS	10 ms	90 ms	

Problem 2: The scheduling used above does not guarantee that deadline-scheduled tasks will run as soon as they arrive. Modify the scheduling so that tasks C and D will not have to wait for any of the others, while maintaining the round-robin relationship between tasks in classes α and γ, and the priority and FCFS used within those classes.

Problem 3: Compute the worst-case latency and response time for the one-shot events listed in the table below.

Event Name	Strong Prior.	Weak Prior.	Handler Run Time
A	2	2	20 μs
B	2	1	30 μs
C	1	3	40 μs
D	1	2	50 μs
E	1	1	100 μs