MP2 Design

sanjitk3

Implementation and Design

1. First, MP1 code was used to create the proc_fs interface for the MP2 module.
2. This is followed updating the write callback to implement 3 cases - Register, Yield and Deregister. Only the register and deregister sections are used at first - where a task structure is used that is wrapped around the typical task_struct of linux. This is used to store items as a linked list of tasks with different states.
3. The dispatching thread is then written that is woken up at every context switch or preemption, different deadline cases are handled here and in the end it goes back to sleep after context switch.
4. The wake up caller function at every period wakes up the process and puts it in the ready state and calls the dispatch thread.
5. The yield portion of proc_write is now written to update the deadline, call dispatch thread and put calling process to sleep when called.
6. userapp is written to register, do jobs and call yield and deregister.

- Admission control is not written. Deallocation is done and exited.