

## Round Robin Scheduling using OSP2

The purpose of this programming assignment is to implement Round Robin CPU Scheduling using OSP2 environment and Java. Round Robin Scheduling involves a thread to execute for a length of time that is constant which is called Time Quantum. This is used to preempt a thread and placed it back to the read queue. This method can be used to allow short threads, corresponding to interactive commands to get through computing quicker and improve its response time. The template classes ThreadCB.java and TimerInterruptHandler.java is given. The Data Structure that is used is a Generic link list given given by OSP2. The round robin logic involve Context Switching which involves passing control of CPU to one thread to another by preempting the thread and dispatching it. The methods in ThreadCB implemented to perform Round Robin scheduling are do\_create(Task cb task), do\_kill(), do\_resume(), do\_suspend() and do\_dispatch(). In Do\_create(TaskCB task), I used the conditional if statement to test if the task is null then call dispatch(). Then another if branch to test if task exceeds the maximum number of threads per task. Then Dispatch() is called and return null. Otherwise, instantiate a new thread and set its task to its priority and status to ready. If the added thread is new thread fails call dispatch and return null. If new thread is appended to the Queue, dispatch is called then returns the thread. Public void do kill() kills the specified thread. Status of thread is assigned to get Status. If the status of thread is ready remove it from the queue. If the statues of thread is running make the CPU to get the task and current thread and set MMU of PTBR to null and set the current thread to null. I test this in try catch block and throws a catch NullPointerException e. Then the thread is removed from the task to set the status to kill the thread. A for loop is implemented to manage table size of the pendingIO and device modules. Next statement gives up resources then calls dispatch. If the task and thread count is 0 kill the thread. In public void do\_suspend(Event event), current status is assigned to get status if status of thread is running, manage PTBR and get the task to current thread and set it to this. If so set PTBR to null and get the task to set its thread null. Otherwise status is waiting set the status and get status and increment 1. If the Queue is contained remove it from the Queue. Then add thread to event then call dispatch(). Next is public void do\_resume() if status of thread is waiting, print of a message to resume and return. If status is ready then append thread to queue and call dispatch. Do\_dispatch() is the method where Round Robin logic is implemented . First thread is instantiated by ThreadCB and is assigned to null. Try and catch block is used to assign the MMU of PTBR and get is task and current thread. The catch block throws a null pointer exception. If

thread is null get its task and set current thread to null then get its task and get thread to null and PTBR to null of MMU. Set its status to ready and append the thread to the Queue.

If Queue is empty, set PTBR of MMU to null and return FAILURE. Otherwise set the thread to the Queue then remove it from the head of the Queue and set PTBR of MMU to the page table then set status of thread to running. Next the quantum is set. And return Success.

When running the program and simulating in OSP2, it is mostly successful. However every now and then there will be a page fault warning.