Kariharan Thilagakumar 100922048

Loic Touzard 101057279

SYSC 4001

Assignment 2

Part I

a) Because the kernel is using Round Robin the schelduler is in preemptive mode and each process gets equal processor time (quantum) before behing interrupted by the timer (hardware). As the question says, the next process election is related to their priority.
The running process may abandon the RUNNING state if :

- It's quantum time is over, and it gets interrupted by the timer. The process will go back to READY state and a new process is elected. Note that this new process can be the same one if it has the higher priority.
- The RUNNING process performs IO request or a waiting system call, it will leave the CPU and go to the WAITING state until the requested task has been completed.
- The RUNNING process finished its task, it simply release the CPU going to TERMINATED state.
- In the case that there is a middle-term schelduler, managing the process mix of the system, the RUNNING process can be swap out. That makes it leave the cycle system and eventually go back to the cycle beeing swap in by the middle-term schelduler, and go to the READY state.
- A new process is created, in preemptive mode, the RUNNING process will leave this state to go back to READY state. If there is a process (maybe the new one) that hase a higher priority it will takes the RUNNING state instead of the old one.

b) In user level threads the thread management kernel is not aware of the existence of threads. The thread library contains code for creating and destroying threads, for passing messages data between threads, for scheduling thread execution and for saving and restoring thread contexts, in the case of the Kernel Level threads there is no thread management code in the application area. Kernel threads are supported directly by the operating system.
However kernel and user level thread are connected together in different possible configuration (depends on the user level thread library) that can be one-to-one, many-to-one, many-to-many relashionship.

c) Check Excel Doc (.ods)

d) Check Excel Doc (.ods)

e) (i) FCFS does not discriminate in favour of long processes. It is a fair algorithm.
(ii) RR does not discriminate in favour of long processes. It is a fair algorithm.
(iii) Multilevel feedback queues will prioritize long processes to the lowest priority.

f) (i) Job No. 1 will go to the 205K memory partition
Job No. 2 will go to the 180K memory partition
Job No. 3 has no free partitions to go to
Job No. 4 will go to the 125K memory partition

(ii) Job No. 1 will go to the 125K memory partition which is the closest size to it
Job No. 2 will go to the 150K memory partition which is the closest size to it
Job No. 3 will go to the 205K memory partition which is the closest size to it
Job No. 4 will go to the 91K memory partition which is the closest size to it

(iii) Job No. 1 will go to the 205K which is the largest partition
Job No. 2 will go to the 180K which is the next largest partition
Job No. 3 has no free partitions to go to
Job No. 4 will go to the 150K which is the largest partition