Candidate Number: 2031B
Paper 2
Question 4

a. FFS:

   P1 waits 0
   P2 waits 5
   P3 waits 6
   P4 waits 9
   Average wait = 20/4 = 5

   SJF:

   P1 waits 0
   P2 waits 5
   P3 waits 6
   P4 waits 9
   Average wait = 20/4 = 5

   SRTF:

   P1 waits 7
   P2 waits 0
   P3 waits 1
   P4 waits 9
   Average wait = 17/4 = 4.25

b. FCFS and SJF each give the fewest number of context switches which is n (including starting the first process in the queue but not including finishing the last process). This is because these methods are not preemptive and so there is only one switch per context. SRTF and RR sometimes switch context in the middle of executing a process, which results in additional context switches. I am assuming that the processes are initially in random order in the queue and are of independent and random lengths.

c.

   i. Even if a process takes a very long time to execute, it is interrupted after a set amount of time (quantum) so that all of the other processes are not kept waiting for too long. Therefore no process has to wait longer than (n-1)q (where n is the number of processes in the queue and q is the quantum) in order to reach its next quantum of allotted time to run

   ii. You could split this critical activity into multiple processes. For example a parent process could fork multiple child processes which each carry out part of the task. This means that although each processes in the queue receives the same quantum of time, more of those processes (and hence more of the quanta) are used to complete this critical task. These processes would even be able to read from the same memory (which would be copy-on-write) and if they needed to communicate with one another, they could write to/read from the disk.

d.

   i. user1 can read file1 and file2.
   user2 can write to file1 and file2.
   user2 and user3 can read file 3.

   ii. rwxr-x--- (0750)
   rwxr-x-- and setuid (4750)