CS-450 & CS-550: *Operating Systems,*
Sections 1 & 2
Fall 2008: Quiz # 2
ANSWERS

INSTRUCTIONS: (1) Closed book, closed notes (but open mind), ***NO*** calculators allowed.
(2) For each question, **circle** the identifying letter next to the choice corresponding to your answer, or fill in the blank, as appropriate.*
(3) You will not get credit for your grade unless you sign the Honor Code declaration on the back of this page(this is a JMU requirement).
(4) You must also print your name legibly ***on the back of this page sheet***, so that I know who you are, and also write the **last four digits only** of your JMU ID number in the indicated location on the back of the page.
(5) For any questions requiring calculation, you must show all your work. If you perform your calculation on sheet(s) of paper not part of the exam, then you must write your name **legibly** on all such sheets and hand them in together with your exam.

1. Indicate which of the activities listed below fall under the rubric of: **Long-Term Scheduling.**
   Answer: **a** (by definition of the term "Long-Term Scheduling) *1 pt*
   - **a.** admission of a process to the system.
   - **b.** dispatch of a process to the CPU.
   - **c.** dispatch of a Kernel-Level Thread to the CPU.
   - **d.** swapping of a process into or out of main memory.
   - **e.** all of the above.
   - **f.** more than one, but **not** all, of the above.
   - **g.** none of the above.

2. Indicate which of the activities listed below fall under the rubric of: **Short-Term Scheduling.**
   Answer: **f** (since ***both*** choice **b** and choice **c** are applicable, but **not** choices **a** or **c.** *1 pt*
   - **a.** admission of a process to the system.
   - **b.** dispatch of a process to the CPU.
   - **c.** dispatch of a Kernel-Level Thread to the CPU.
   - **d.** swapping of a process into or out of main memory.
   - **e.** all of the above.
   - **f.** more than one, but **not** all, of the above.
   - **g.** none of the above.

3. Indicate those process states during which the entire content of the Process Control Block is up-to-date:
   Answer: **e** (since choices **b** and **c** are ***both*** applicable, but **not** choice **a**) *2 pts*
   - **a.** Running
   - **b.** Waiting
   - **c.** Ready
   - **d.** all of the above
   - **e.** more than one, but **not** all of the above
   - **f.** none of the above

4. Indicate what ***Linux*** command will cause the absolute pathname of your Current Working Directory to be output to a file called "current" that is situated in the current working directory.
   1 pt

   Answer: `pwd > current`

Last four digits of your JMU Number: ___ ___ ___ ___

This work complies with the JMU Honor Code: ______________________________

YourSignature

Please **print** your name **legibly:** ______________________________

5. Indicate what ***Linux*** command will cause your Current Working Directory to be reset to your Home Directory. **Note:** To get credit for a correct answer, you must **not** have to type out either the absolute or the relative pathname of your Home Directory.

1 pt

Answer: `cd ~` or just `cd`

6. Several processes enter the Ready Queue in the order shown. Each I/O takes 20 msec, and each process completes after executing 1,001 CPU bursts and 1,000 I/Os. Context switching takes less than 100 μsec, and can therefore be ignored. Assume that the CPU Burst Time specified for each process is identical from one CPU burst to the next: For **First-Come-First-Served** scheduling, determine the result of the **fourth** scheduling decision made by the process scheduler. Indicate **which** process will be selected, and at **what time** will it get the processor. Write your answers in the space provided.

4 pts

Time of Arrival	Process ID	CPU Burst Time (msec)
0.0 msec	P1	8
1.0 msec	P2	12
2.0 msec	P3	27
3.0 msec	P4	4
4.0 msec	P5	29

Comment: At t = 0, the Ready Queue contains only one process, which certainly makes the first scheduling decision a simple one. **P1** gets the processor, and holds it until its CPU burst finishes at 8 msec. when it issues a blocking I/O call and goes onto the Wait Queue for 20 msec, that is, until t = 28 msec. **P2** then gets the processor for 12 msec, *i.e.*, until t = 20 msec, when it blocks until 40 msec. At t = 20 msec, **P3** gets the processor, and keeps it until t = 47 msec. By 47 msec, both **P1** and **P2** have completed their I/Os and are back on the Ready Queue. However, out of the four processes on the Ready Queue at this point, **P4** was first in, and therefore it gets the CPU.

The following set of tables is provided to aid you in figuring out the answer. However, you are not provided to use these tables. **PID** stands for "Process Identifier", and **RBT** stands for "Remaining [CPU] Burst Time"

Queue Content. t = 0		
Element	**PID**	**RBT**
First:	**P1**	**8**
Second:		
Third:		
Fourth:		
Fifth:		

Queue Content, t = 8		
Element	**PID**	**RBT**
First:	**P2**	**12**
Second:	**P3**	**27**
Third:	**P4**	**4**
Fourth:	**P5**	**29**
Fifth:		

Queue Content, t = 20		
Element	**PID**	**RBT**
First:	**P3**	**27**
Second:	**P4**	**4**
Third:	**P5**	**29**
Fourth:		
Fifth:		

Queue Content, t = 47		
Element	**PID**	**RBT**
First:	**P4**	**4**
Second:	**P5**	**29**
Third:	**P1**	**8**
Fourth:	**P2**	**2**
Fifth:		

Queue Content		
Element	**PID**	**RBT**
First:		
Second:		
Third:		
Fourth:		
Fifth:		

Queue Content		
Element	**PID**	**RBT**
First:		
Second:		
Third:		
Fourth:		
Fifth:		

Queue Content		
Element	**PID**	**RBT**
First:		
Second:		
Third:		
Fourth:		
Fifth:		

Queue Content		
Element	**PID**	**RBT**
First:		
Second:		
Third:		
Fourth:		
Fifth:		

Answer to Question 6:	**PROCESS:**	**P4**	**START Time:**	**47 msec**