Full Name: _____

CS 450 Spring 2009

Midterm Exam

Instructions:

- This exam is closed-book, closed-notes.
- Keep your written answers concise and to-the-point. I reserve the right to deduct points for needless verbiage.
- Write your full name on the front, and make sure that your exam is not missing any sheets.
- Good luck!

Problem 1 (/12) :	
Problem 2 (/6) :	
Problem 3 (/6) :	
Problem 4 (/6) :	
Problem 5 (/6) :	
Problem 6 (/12) :	
TOTAL (/48) :	

Problem 1. (12 points):

Multiple choice. For each of the following multiple choice problems, choose the *single best* answer by circling its corresponding letter.

1. Which of the following criteria most directly influences the decision to switch to an in-core, runnable process?

   (a) the `runin` flag

   (b) the process status (`p_stat`)

   (c) the process priority (`p_pri`)

   (d) the processor priority level

2. Which of the following data structures associated with active processes can be swapped out when the process is not running?

   (a) the kernel stack

   (b) the `proc` struct

   (c) the `callout` array

   (d) the interrupt vector

3. It is frequently the case that v6 code needs to be run *atomically*, i.e., without the possibility of being interrupted by code that will modify crucial data structures concurrently. Which of the following can be used to *begin* an atomic chunk of code?

   (a) `spl0()`

   (b) `rp->p_stat = SWAIT`

   (c) `bis  $340, PS`

   (d) `mov  $1, SSR0`

4. The current user struct can be accessed in the kernel via the pointer `_u`. Which of the following correctly initializes `_u`?

   (a) `_u = 140000`

   (b) `_u = *ka6`

   (c) `mov $USIZE-1\<8|6, _u`

   (d) `UISA->r[7] = ka6[1]`

Problem 2. (6 points):

What does the following line of code in `swtch` accomplish? Why is it necessary?

```
retu(proc[0].p_addr);
```

Problem 3. (6 points):

Explain the function of the following code. When might you expect to find it executed?

```
2:
  bis  $340, PS
  tstb _runrun
  beq  2f
  bic  $340, PS
  jsr  ps,_swtch
  br   2b
2:
```

Problem 4. (6 points):

Is aging a sufficient mechanism for combatting priority inversion? Why or why not?

Problem 5. (6 points):

Consider the design of a multi-level feedback queue (MLFQ) scheduler consisting of a round robin queue with $q = 10$ms and a FCFS queue. 70% of the processor time is allocated to the RR scheduler, and 30% is dedicated to processes on the FCFS queue. Describe separate scenarios that may cause a given process to be moved from the RR to the FCFS queue and then back again.

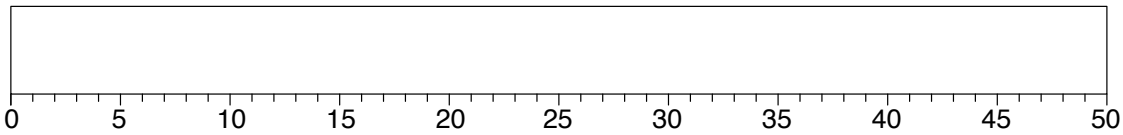
Problem 6. (12 points):

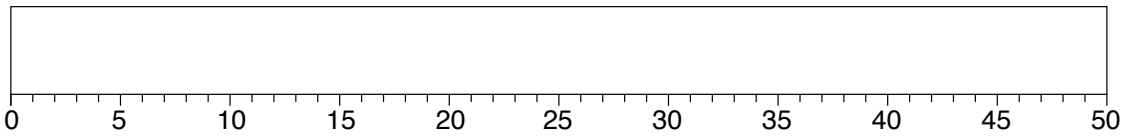
Consider the following two processes:

- P_1, arriving at time $t = 0$, completing after two CPU bursts of 10ms each separated by a single I/O burst of 15ms.
- P_2, arriving at time $t = 2$, completing after three CPU bursts of 5ms each separated by two I/O bursts of 10ms each.

Assume that there is no context switch overhead.

A. Fill in the following Gantt chart template to chart the execution of the two processes using the pre-emptive SJF scheduling algorithm. You should shade in sections where neither process is active. (Multiple templates are provided in case you mess up).





B. What is the total waiting time for each of the two processes? (Recall that waiting time does not include I/O overhead.)

C. What is the CPU utilization over the execution of the two processes – i.e., what is the fraction of time during which the CPU is not idle?

D. Could the CPU utilization be improved using a different scheduling algorithm? Justify your answer.