Name: _____

UB ID Number: | | | | | | | | |

Question:	1	2	3	4	5	6	7	8	Total
Points:	10	5	5	5	5	5	5	20	50
Score:									

CSE421 Midterm Exam

09 Mar 2012

This midterm exam consists of three types of questions:

1. **10 multiple choice** questions worth 1 point each. These are drawn directly from lecture slides and intended to be easy.
2. **6 short answer** questions worth 5 points each. You can answer as many as you want, but we will give you credit for your best four answers for a total of up to 20 points. You should be able to answer the short answer questions in four or five sentences.
3. **2 long answer** questions worth 20 points each. **Please answer only one long answer question.** If you answer both, we will only grade one. Your answer to the long answer should span a page or two.

Please answer each question as **clearly** and **succinctly** as possible. Feel free to draw pictures or diagrams if they help you to do so. **No aids of any kind are permitted.**

The point value assigned to each question is intended to suggest how to allocate your time. So you should work on a 5 point question for roughly 5 minutes.

Please fill out your name and UB ID number above. Also write your UB ID number at the bottom of each page of the exam in case the pages become separated.

There are **three** scratch pages at the end of the exam if you need them. If you use them, please clearly indicate which question you are answering.

I have neither given nor received help on this exam.

Sign and Date: _____

Multiple Choice

1. (10 points) Answer all **ten** of the following questions. Each is worth **one** point.

   (a) In the story that GWA (Geoff) began class with on Monday, March 4th, why was the Harvard student concerned about his grade?
   ◯ He never attended class. ◯ He never arrived at class on time. ◯ He usually fell asleep in class. ◯ He was using drugs.

   (b) All of the following are inter-process (IPC) communication mechanisms *except*
   ◯ shared files. ◯ exit codes. ◯ pipes. ◯ non-uniform memory.

   (c) New processes are created by calling
   ◯ `fork()`. ◯ `exec()`. ◯ `create()`. ◯ `new()`.

   (d) What does `exec()` do to the process file table?
   ◯ Copies it. ◯ Opens `STDIN`, `STDOUT` and `STDERR`. ◯ Nothing. ◯ Resets it.

   (e) What MIPS instruction is used by the kernel to return to userspace after handling an exception?
   ◯ `rfe` ◯ `lw` ◯ `syscall` ◯ `addiu`

   (f) Which of the following is *not* a requirement for deadlock?
   ◯ Multiple independent resource requests. ◯ A linear dependency graph.
   ◯ Protected access to shared resources. ◯ No resource preemption.

   (g) Which of the following is *not* an example of an operating system policy?
   ◯ Deciding which thread to run. ◯ Giving preference to interactive tasks.
   ◯ Using timer interrupts to stop a running thread. ◯ Choosing a thread to run at random.

   (h) When using our computers, normal users are generally *not* actively aware of
   ◯ responsiveness. ◯ continuity. ◯ resource allocation. ◯ interactivity.

   (i) Con Kolivas is
   ◯ the maintainer of the Linux scheduling subsystem. ◯ a Turing award winner. ◯ opposed to the use of profanity. ◯ an Australian anaesthetist and Linux hacker.

   (j) Address translation allows the kernel to implement what abstraction?
   ◯ Address spaces. ◯ Files. ◯ Threads. ◯ Processes.

 UB ID: | | | | | | | | |

Short Answer

Choose **4 of the following 6** questions to answer. You may choose to answer additional questions, in which case you will receive credit for your best four answers.

Virtual Page Number	Physical Page Number	Permissions
98	119	Read
2	120	Read

Table 1: **TLB.**

Virtual Page Number	Location	Address	Permissions
98	Memory	119	Read, Write
0	Memory	12	Read, Write
2	Memory	120	Read
3	Disk	2	Read, Write

Table 2: **Process Page Table.** Each entry corresponds to a PTE.

2. (5 points) Using both the TLB (Table 1) and page table (Table 2) for the running process above, describe what would happen if each of the following pseudo-instructions were executed. In particular, make sure to identify any TLB or page faults. Assume this (fairly weird) machine uses 1000 byte pages.

   Note: to make things easier on everyone the question uses base-10 arithmetic.

   1. load 2378
   2. store 98733
   3. load 36788
   4. load 143
   5. store 3700

 UB ID: | | | | | | | | |

3. (5 points) Identify and describe three serious problems with the code snippet below. (You may want to use the line numbers to help identify the problems.) Assume `sharedStateLock` and `sharedStateCV` have been properly initialized.

```
1
2  struct lock * sharedStateLock;
3  struct cv * sharedStateCV;
4  bool getGoing = false;
5  int sharedState = 0;
6
7  void
8  fubar(int doubleRainbow)
9  {
10   // Might already have the lock!
11   if (!lock_do_i_hold(sharedStateLock)) {
12     lock_acquire(sharedStateLock);
13   }
14
15   // Wait to get going! Grab the lock to protect the shared state.
16   while (!getGoing) {
17     ;
18   }
19   lock_release(sharedStateLock);
20
21   // Reset shared state before we make our changes.
22   sharedState = 0;
23
24   lock_acquire(sharedStateLock);
25   sharedState = doubleRainbow;
26   lock_release(sharedStateLock);
27
28   // Let everyone know about the double rainbow!
29   cv_signal(sharedStateCV, sharedStateLock);
30   return;
31 }
```

 UB ID: | | | | | | | | |

Extra space for Problem #3

 UB ID: | | | | | | | | |

4. (5 points) Give an example of (1) a hardware interrupt, (2) a software interrupt, and (3) an exception. (Three examples total.) Briefly describe what happens when an interrupt is triggered.

UB ID: | | | | | | | | |

5. (5 points) First, from the perspective of the operating system, what is the difference between interactive and non-interactive threads? Or, put another way, describe how the operating system might try to guess whether a thread is interactive or not. Second, describe how either multi-level feedback queues (MLFQ) or the Rotating Staircase Deadline (RSDL) scheduler prioritize interactive threads.

UB ID: | | | | | | | | |

6. (5 points) Operating systems require special privileges to multiplex memory. Below, describe:

   - **what** special privileges are required,
   - **how** they are used,
   - and **why** they are needed.

UB ID: | | | | | | | | |

7. (5 points) Describe the tradeoffs surrounding memory page size. What happens when pages become very small? What happens when pages become very large?

UB ID: | | | | | | | | |

Long Answer

Choose 1 of the following 2 questions to answer. **Please do not answer both questions.** If you do, we will only read one.

If you need additional space, continue and clearly label your answer on the back of this or other exam sheets.

8. (20 points) Choose one of the following two questions to answer:

   1. **User- v. kernel-level multithreading.** In class we focused our discussion of the thread abstraction on kernel-level threads. However, a popular alternative is to implement threads in userspace libraries. We refer to these threads as user-level threads.

      First, explain what a userspace library would need to do to implement the thread abstraction. How are threads created? Where is thread state stored? How do you perform a context switch? Can you implement preemption and, if so, how? What support from the kernel, if any, is needed to accomplish these things?

      Second, discuss the tradeoffs between implementing threads in userspace and in the kernel. What's potentially better about user-level threads? What's potentially better about kernel-level threads? Give one example of an application that you argue would perform better using user threads and one application that you argue would perform better with kernel threads.

---

   2. **System design principles.** We have discussed a number of general systems design principles throughout the semester. As an example, when motivating on-demand paging we introduced the idea that procrastination might be effective if it allows the kernel to avoid doing things that it will never have had to do, such as loading an unused code page into a process address space.

      List three *other* design principles that we have discussed this semester. Explain each design principle clearly and illustrate each principle with an operating systems example drawn from class. In addition, for each principle construct a new example of its applicability not drawn from class. Your examples do not necessarily have to be drawn from the world of operating systems or even the world of computers, but those are good places to start.

 UB ID: | | | | | | | | |

Scratch. **Please indicate what question you are answering.**

 UB ID: | | | | | | | | |

Scratch. **Please indicate what question you are answering.**

 UB ID: | | | | | | | | |

Scratch. **Please indicate what question you are answering.**

 UB ID: | | | | | | | | |

Scratch. **Please indicate what question you are answering.**

 UB ID: | | | | | | | | |