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 Alternate Midterm Exam

08 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 very 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) On Monday, March 4th, (GWA) Geoff began class with a story about which famous Saturday Night Live comedian?
   ◯ Tina Fey ◯ Al Franken ◯ Sarah Silverman ◯ Dan Akroyd

   (b) Unlike threads, address spaces and files, the process abstraction is not tied to
   ◯ a hardware component. ◯ another abstraction. ◯ a specific policy. ◯ user behavior.

   (c) What does the following system call do?

   ```
   waitpid(54);
   ```

   ◯ Fail. ◯ Return the exit code of process 45. ◯ Block until process 45 exits. ◯ There is not enough information to tell.

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

   (e) What MIPS instruction do processes use to request kernel attention?
   ◯ `rfe` ◯ `lw` ◯ `syscall` ◯ `addiu`

   (f) True or false: the following code ensures that variable `foo` is protected? (Assume `foo_lock` exists and was properly initialized.)

   ```
   lock_acquire(foo_lock);
   // modify foo
   lock_release(foo_lock);
   ```

   ◯ True. ◯ False.

   (g) Which of the following is *not* an example of an operating system mechanism?
   ◯ A context switch. ◯ Using timer interrupts to stop a running thread. ◯ Maintaining the running, ready and waiting queues. ◯ Choosing a thread to run at random.

   (h) Normal users are actively aware of computer
   ◯ resource allocation. ◯ responsiveness. ◯ throughput. ◯ power consumption.

   (i) Who is Ingo Molnar?
   ◯ The maintainer of the Linux scheduling subsystem. ◯ The author of the Rotating Staircase Deadline Scheduler (RSDL). ◯ An Australian who tried to change the Linux scheduler. ◯ A Turing award winner.

   (j) A virtual address might point to all of the following *except*
   ◯ physical memory. ◯ a disk block. ◯ a port on a hardware device. ◯ a register on the CPU.

 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.

2. (5 points) Describe the process of performing a context switch.

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

3. (5 points) Dewey, Cheatham, and Howe are car mechanics at the Good News Garage. One day Ray brings in his old MG which is, of course, in terrible shape. There are three distinct problems to solve. Dewey, Cheatham and Howe divide up the tasks between them, grab their tools and get to work.

   Like most car mechanics, Dewey, Cheatham and Howe share a set of tools (they're cheapskates), only pick up one tool at a time (the other hand is holding a beer), grab all the tools that they need before beginning a task (so that they don't have to move after they begin), and don't surrender a tool until they have finished whatever it is they are trying to do (or think that they have finished). As they approach the MG, Dewey sees that his problem requires the wrench and the hammer, Cheatham sees that his requires the pry bar and the wrench, and Howe sees that his requires the pry bar, hammer, and blow torch.

   Despite their best intentions, describe a situation in which Dewey, Cheatham and Howe will find themselves unable to get any work done. Propose two separate solutions that will allow them to complete the job and get Ray's car back on the road. Your solutions should allow multiple mechanics to work on the car at once when possible.

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

4. (5 points) We discussed several approaches to doing inter-process communication, or IPC. Below, describe two IPC mechanisms. For each, provide an example of interaction between two processes well-suited to that mechanism.

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

5. (5 points) Explain what happens when a page is swapped in from disk.

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

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

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

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

7. (5 points) It's your first week working at Macrohard, makers of the newly-popular Windix operating system. You have been hired as part of the Desktop Performance Group, focused on improving the somewhat sluggish performance of Windix on consumer machines.

   Your boss Geoff Challen comes to you early one morning perplexed. He has been working on a sophisticated new page replacement algorithm utilizing AI-inspired algorithms to choose the best page to swap out when memory runs low. He has convincing evidence that his new algorithm is doing a much better job of choosing a page than the very simple approach currently used by Windix. However, when he performs performance measurements he finds that a system using his new approach is actually *slower* despite the fact that it is choosing pages to evict more effectively.

   Geoff thinks that there must be something wrong with the testing suite that he is using. You have a more convincing explanation. Detail it below.

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. **Compressed memory.** We have discussed swapping in class, which frees up memory by moving pages to disk. Another approach to freeing up memory compresses the content of one page allowing it to be stored in a smaller amount of memory. When the page is accessed again, the compressed page is decompressed to restore its contents.

      As an example, imagine that I have an algorithm that can reliably compress a 4K page into 2K. First, describe how your system would use this algorithm to free up system memory. Include a detailed description of when compressiond and decompression are performed. Compare the cost-benefit tradeoff of this compression-driven approach to swapping.

      Second, imagine I have a system that can either compress or swap pages to free memory. Describe how the system might choose which approach to use at a given point in time.

---

   2. **User-level system calls.** Context switching is expensive and creates an overhead for kernel entry and hence for making system calls from user space. For some system calls entering the kernel and running kernel code is unavoidable. For other system calls, however, the kernel transition may be avoided.

      Consider the `getpid()` system call. First, propose a way to implement `getpid()` entirely in usermode without entering the kernel. Please be detailed about how your new implementation will work and any changes to user processes or the kernel that will be required.

      Second, imagine that you have implemented your new `getpid()` which does not enter the kernel. Afterwards, however, you notice no speedup to your user programs. Explain why your change to `getpid()` results in no appreciable performance change.

 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: | | | | | | | | |