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

08 Mar 2013

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 of the angry diva, she was upset by the color of what candy?
   ○ Skittles. ○ Reeses' Pieces. ○ M&Ms. ○ Sour Patch Kids.

   (b) Which of the following is an inter-process (IPC) communication mechanism?
   ○ Address spaces. ○ Virtual addresses. ○ Pipes. ○ Grapes.

   (c) Process terminate by calling
   ○ `kill_me_now()`. ○ `kthanxbai()`. ○ `_exit()`. ○ `terminate()`.

   (d) What does `mmap()` do to the process file table?
   ○ Copies it. ○ Opens `STDIN`, `STDOUT` and `STDERR`. ○ Nothing. ○ Waits for it.

   (e) Which of these schedulers cannot be implemented?
   ○ Rotating Staircase. ○ Multi-Level Feedback Queues. ○ Shortest-Job First. ○ Random.

   (f) Which of the following is a requirement for deadlock?
   ○ A single resource request. ○ A cycle in the dependency graph. ○ Unrestricted access to shared resources. ○ Preemptible resources.

   (g) Which of the following is *not* an example of an operating system mechanism?
   ○ Context switching. ○ Address translation. ○ Preferring interactive threads. ○ `panic()`.

   (h) An example of a continuity-oriented computer process would be
   ○ backing up files. ○ responding to a click event. ○ opening a new Firefox tab. ○ watching a movie.

   (i) The bounded buffer producer-consumer synchronization problem can be solved using
   ○ two semaphores. ○ one semaphore. ○ locks. ○ recursive locks.

   (j) Address spaces are *not* usually
   ○ big. ○ fragmented. ○ sparse. ○ organized.

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) We have discussed two cases where operating systems add a level of indirection to achieve a desirable property. Identify each, and explain why indirection is helpful in both cases.

UB ID: □ □ □ □ □ □ □ □

3. (5 points) Identify three similarities or differences between Multi-Level Feedback Queues (MLFQ) and the Rotating Staircase Deadline Scheduler (RSDL).

4. (5 points) Provide one example of a exception that would terminate a running process and one example of an exception that would not. Describe what happens when an exception occurs.

UB ID: □□□□□□□□

5. (5 points) Describe the process life cycle and related system calls.

UB ID: □ □ □ □ □ □ □ □

6. (5 points) Distinguish between *atomicity* and *concurrency*. Both can be real, or illusions—describe examples of each, and describe how the illusion is implemented in each case.

UB ID: □□□□□□□□

Starting Virtual Address	Bound	Base Physical Address	Permissions
0	100	150	R,W
348	56	2040	W
3678	510	260	R,W,E
288	16	10,000	R

Table 1: **Segmentation Table.**

7. (5 points) Given the current MMU segment table above, indicate the result of the following five load, stores, and fetches (load and execute.) **Note: to make things easier on everyone the question uses base-10 arithmetic.**

   1. fetch 78
   2. load 388
   3. fetch 4000
   4. store 244
   5. store 0

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. **Wait Time Prediction.** When discussing schedulers, one of the aspects of future that we wanted to be able to predict was the wait time when a thread blocks. To be more concrete, when a thread blocked and was moved to the waiting queue, we might want to know *how long* the wait will take.

      First, discuss how online prediction of wait times might be accomplished on a real system—no crystal balls allowed. Identify a few things that a thread might block waiting for, and for each discuss whether a prediction algorithm makes sense and how it might be implemented. You might think about applying some of the system design principles we've discussed in class.

      Second, discuss how to incorporate this information into a scheduling algorithm. Feel free to choose one of the scheduling algorithms we've discussed in class in order to make your solution more concrete. You should explain how to use the output of your wait predictor and argue that it can improve some aspect of scheduler performance.

   2. **M:N Threading Implementations.** We've discussed both the M:1 threading model, where all threads are implemented by userspace libraries, and the 1:1 threading model, where all threads are implemented by the kernel. Another model exists: the M:N threading model, where the kernel supports multiple threads (N) but userspace libraries use them to support $M - N$ more. (We consider M to be strictly greater than N.)

      First, explain why this could be preferable to both the M:1 and the 1:1 threading models. Which benefits of each are preserved? Which are lost?

      Second, discuss how to implement the M:N threading model. Describe the threading interface available to processes, interaction between the threading library and the kernel, and how thread scheduling is performed.

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: □ □ □ □ □ □ □ □