CSCI 1670 Midterm Exam

Closed Book
October 24, 2007

Do all of questions 1 through 3.

1. Assume we have a uniprocessor system. Each thread has both a user-mode stack and a kernel-mode stack. Interrupt handlers execute on the kernel-mode stack of the current thread. In the following, please mention in whose context (thread or interrupt handler) actions are taken and which stack is being used. If you use a term such as APC, explain what it means and how it works. A thread may call *SchedYield* (in the kernel) to yield the processor to another thread.

   a. Our system is to be preemptible in user mode, but not preemptible within the kernel. The clock interrupt handler determines if the current thread's time slice is over and, if so, requests that it be preempted. Describe the mechanism used to preempt the thread. Note that keeping interrupts masked the entire time a thread is in the kernel is not feasible.

   b. Our system is now to be preemptible both in user mode and in the kernel. Describe the mechanism used to preempt a thread.

2. In the Fortran programming language, arrays are stored in column-major order, meaning that adjacent entries in each column are stored in adjacent memory locations. For example (using C notation for matrices), Matrix[i][j] is stored just before Matrix[i+1][j]. However, most scientific programs written in Fortran access arrays using loops similar to the following (again, using C notation rather than Fortran):

```
int Matrix[M][N];
for (i=0; i<M; i++)
   for (j=0; j<N; j++)
      Matrix[i][j] = somefunction(i, j);
```

   For example, if M is 6 and N is 4, the elements of the matrix are stored in the order shown below:

0	6	12	18
1	7	13	19
2	8	14	20
3	9	15	21
4	10	16	22
5	11	17	23

   This caused no problems until such programs were run on machines supporting virtual memory, where performance (particularly for large values of M and N) turned out to be much worse than on machines that didn't support virtual memory. Explain why performance was worse. (Hint: consider the page-replacement policy.)

3. The layout of the file system on disk is an important factor in file-system performance. Since files are often accessed sequentially, the performance of sequential access is important.

   a. In the early implementations of the FFS file system, the kernel was constrained to accessing files one block at a time — combining two one-block requests into one two-block request was not possible. (To keep your answer relatively simple, assume that files are allocated in whole blocks and that fragments are not used.) You would like to eliminate this constraint, allowing multiple one-block requests to be combined into a single multiple-block request. What constraints would be necessary on the layout of the blocks for this to be possible? How could the file system determine that these constraints are satisfied? (Hint: think about metadata.) What modifications would be necessary to the the kernel's buffer-management code to support such multi-block transfers?

   b. Would block-interleaving still be necessary if this approach were used? Explain. (Assume that the disk controllers employed do not buffer tracks internally.)

   c. Modern disk controllers read the contents of entire tracks into internal buffers, from which the data can be transferred directly to primary memory in the computer. (Assume the transfer from the buffer to computer memory takes neglible time.) Would this make unnecessary the scheme described in part a? (We are concerned only with reading data, not with writing data.)

If you do all of the following correctly, you'll get an A regardless of how well you do on the first three problems. If you miss any of the following, your grade will be based solely on how well you do on the first three problems.

4. What is a Rhinopias (other than a kind of disk)?

5. Why is the Sun Lab called the "Sun Lab"?

6. In the early days of the Brown CS department, most instructional computing was done on computers made by Apollo and most research computing was done on computers made by Digital. Neither company currently exists. What one company now owns what remains of both of them?

7. The CS Department purchased an academic license for Unix in 1979. How much did it cost? How much would a commercial license have cost?

8. What did the original Unix license prohibit the teaching of?

9. Two CS faculty members hold endowed chairs named for computer-industry pioneers. Who are the faculty members and who are the pioneers?

10. Your professor has been teaching the operating systems course at Brown for a long time. Which current Brown faculty member taught it before he did?