Course ECE354 Real-time Operating Systems

Final Exam

Aug 10, 2010

Be concise, short, and to the technical point in your answers. Otherwise, you will be unable to answer all questions in time.

1. What material did you **mainly** use to prepare for the final exam? Pick only one; any selection will get you full points. *[1p]*

   □Printed book □Electronic book □Own notes on slides □Abstain

2. Explain the conceptual difference between threads and processes by explaining what is a thread and what is it used for, and what is a process and what is it used for. *[6p]*

3. What is a zombie process? *[3p]*

4. Describe how a circular buffered I/O system works and its main advantage over a double buffered I/O system. *[10p]*

5. Explain how an I/O system that only maintains data buffers in the process's user space can run into a deadlock situation. *[5p]*

6. Explain the conceptual difference between working set and resident set by explaining each term and then describing the differences. *[15p]*

7. On Slashdot.org an Anonymous Coward posted that adding more cores to the computer always makes a multi-threaded application run faster. Show that coward that you learnt more than he did during your undergrad studies. *Note: Feel free to discuss the computer architecture as well as the operating system perspective.* *[10p]*

8. The I/O subsystem receives the following track read requests: 115, 143, 62, 66, 103, 200, 189, 107, 8, and 81. The current location is 100 and the previously read track was 101. Schedule the I/O using C-SCAN. Show the access behaviour in a figure with time on the x-axis and the track number on the y-axis. The disk's read head moves 3 tracks per mm on the x axis. *[10p]*

9. Assume a system which uses virtual memory with segmentation and paging:

   - Explain in one to two sentences each of these elements: (1) Virtual address, (2) segment table, (3) segment table pointer, (4) page table, (5) physical address, and (6) main memory. *[10p]*
   - Draw a diagram of the translation mechanism. *[5p]*
   - Explain why there is no need for a page table pointer. *[3p]*

10. Explain three advantages of kernel-level threads over user-level threads. *[9p]*

11. Discuss the relationship between the page size and the number of page faults. *Hint: Draw a figure with the page size on the x-axis and page faults on the y-axis, and then discuss the figure.* *[12p]*

12. Explain compaction in memory management and whether it tackles internal or external fragmentation. *[10p]*

13. List a threat to (a) data confidentiality, (b) communication integrity, and (c) hardware availability. *[12p]*

14. Explain what a rootkit is and how it works. *[15p]*

15. Disk striping can improve the data transfer rate when strip size is small compared to the I/O request size.

    - Why and how does stripping improve I/O performance compared to a comparable disk array without stripping in RAID systems? *[10p]*
    - Does this also apply to RAID 0? *[3p]*

16. Which address register does gcc use as the frame pointer? *[3p]*

17. An I-process is running and calls request_memory_block() primitive while the system does not have any memory blocks available. Will the I-process be blocked? *[3p]*

18. Bonus question: Find and correct all bugs in the implementation of the reader-writer's problem of the listing shown below. Write a one sentence explanation per bug. If you need to insert or delete lines, still refer in your answers to the original lines for subsequent bugs. *[17p]*

```
1 int readcount=1,writecount=-1;
2 semaphore x = 0, y = 0, z = 0, wsem = 0, rsem = 1;
3
4 void reader() {
5   while (true) {
6       semWait (z);
7       semWait (rsem);
8       semWait (x);
9       readcount++;
10      if (readcount > 1)
11        semWait (wsem);
12      semSignal (x);
13      semSignal (rsem);
14      semSignal (z);
15      READUNIT();
16      semWait (x);
17      readcount--;
18      if (readcount < 0)
19        semSignal (wsem);
20    }
21 }
22
23 void writer () {
24   while (true) {
25      semWait (y);
26      if (writecount > 1)
27        semWait (rsem);
28      writecount++;
29      semSignal (y);
30      semWait (wsem);
31      WRITEUNIT();
32      semSignal (wsem);
33      semWait (y);
34      semWait(z);
35      writecount--;
36      if (writecount < 0)
37        semSignal (rsem);
38      semSignal(z)
39      semSignal (y);
40    }
41 }
42
43 void main() {
44   parbegin (reader, reader, reader, reader, reader,  writer, writer, writer);
45 }
```

End of quiz. Total points (incl. bonus): 172