Chalmers Un. of Technology and Gothenburg Un. Comp. Science and Engineering Department

Operating Systems
DIT 400, EDA092

Exam 2011-03-15

Date, Time, Place: Tuesday 15/3 2011, 8:30-12:30, M building

Course Responsible: Arne Dahlberg, Marina Papatriantafilou (Tel: 772 1705, 772 5413)

Auxiliary material: You may have with you
- An English-Swedish, Swedish-English dictionary.
- No other books, notes, calculators, PDA's etc.

Grade-scale ("Betygsgränser"):
CTH:3:a 30-38 p, 4:a 39-47 p, 5:a 48-60 p
GU: Godkänd 30-47p, Väl godkänd 48-60 p

Instructions

- Do not forget to write your personal number, if you are a GU or CTH student and at which program ("linje").
- Start answering each assignment on a new page; number the pages and use only one side of each sheet of paper.
- Write in a **clear manner** and **motivate** (explain, justify) your answers. If it is not clear what is written, your answer will be considered wrong. If it is not explained/justified, even a correct answer will get **significantly** lower (possibly zero) marking.
- If you make any assumptions in answering any item, do not forget to clearly state what you assume.
- The exam is organized in groups of questions. The credit for each group of questions is mentioned in the beginning of the respective group. Unless otherwise stated, all questions in a group have equal weight.
- Please answer in English, if possible. If you have large difficulty with that and you think that your grade can be affected, feel free to write in Swedish.

Good luck !!!!

1. (10 p)

   (a) Assume that one disk in a RAID array becomes useless. Which measures need to be taken to recover full access to the data if using
   i. RAID 1 (1p)
   ii. RAID 5 (1p)

   (b) Mention two types of errors that RAID do not protect against. (1p)

   (c) Describe the SCAN scheduling method for optimizing head movement in disk memories. (2p)

   (d) Consider the disk scheduling algorithms SCAN and FCFS.
   i. Which of the methods is likely to give best performance for a heavily loaded system? (1p)
   ii. Which of the methods is likely to give best performance for a very lightly loaded system (only one request in the queue)? (1p)

   (e) The BSD filesystem used rotationally optimal placement of data blocks. Newer filesystems like the Linux ext3 filesystem use sequential placement.
   i. Which change in hardware have lead to sequential access being preferred today? Explain why. (2p)
   ii. Why are optimizations based on rotational position not useful today? (1p)

2. (10 p)

   (a) How is a file system directory implemented in a UNIX file system? (1p)

   (b) UNIX have a *link* system call for creating a hard link to a specified file. Describe the internal function (implementation) of the link system call. (2p)

   (c) If a user program reads a file on a local file system or an NFS file system, the same system call will be used in both cases. Explain the mechanism that makes the use of different file systems transparent in file system accesses. (2p)

   (d) What does it mean that the naming method in a distributed file system is *location independent*? (1p)

   (e) Under which circumstances is a *location independent* naming method needed. (1p)

   (f) The Andrew filesystem uses a consistency semantics called *Session Semantics*. What is the main advantage with using *Session Semantics* compared to UNIX semantics? (1p)

   (g) Describe two fundamental differences between AFS and NFS. (2p)

3. (10 p)

   (a) Protection mechanisms are sometimes described by an access matrix. What is an access matrix? (1p)

   (b) What is the main difference between access lists and capability lists? (1p)

   (c) In UNIX 4.2 BSD there was a new system call (socket) for process communication. Give two problems with pipes that may have motivated the introduction of sockets. (2p)

   (d) Sockets is a general mechanism for process communication. The client side will normally open only one file descriptor but at the server side two or more file descriptors are usually opened. Why is more than one file descriptor used at the server? (1p)

   (e) VMware workstation can use a technique called *binary translation* to run an unmodified guest OS atop a hypervisor on X86 processors.
   i. Explain how *binary translation* works. (3p)
   ii. Why was *binary translation* used in VMware. (1p)
   iii. How would it today be possible to run an unmodified guest OS atop a hypervisor on an X86 processor without using *binary translation* (1p)

4. (10 p)

   (a) (4 p) The following data is given for a system: Demand paging, with the page table in internal registers. The memory access time is 100 ns. A page fault and the following read operation requires 8 ms plus another 12 ms if a modified page is replaced. Assume that 70% of the replaced pages are modified. What is the maximum page fault frequency that can be accepted if an effective access time of maximum 200 ns is wanted? Explain the method that you used for solving the question.

   (b) (4 p) (i) Describe the clock algorithm for page replacement.
   (ii) Assume that you are monitoring the rate at which the pointer in the clock algorithm (the one that indicates the candidate page for replacement) moves. What can you say about the system if you notice that it is moving fast? What if it is moving slow?

   (c) (2 p) Consider a logical address space of eight pages of 1024 bytes each, mapped onto a physical memory of 32 frames.
   i. How many bits are there in the logical address?
   ii. How many bits are there in the physical address?

5. (10 p)

   (a) Can a multithreaded program using user-level threads achieve better performance on a multiprocessor system than on a single-processor one?

   (b) Describe how communication and synchronization among threads can be a factor that can affect scheduling decisions in multiprocessor systems. Use an example to illustrate your description.

   (c) Why is it important for the scheduler to distinguish between I/O-bound and CPU-bound programs?

   (d) What would happen if you executed the following piece of code:

```
main()
{ for( ; ; )
  fork();
}
```

6. (10 p)

   (a) (3 p) What is a deadlock? What are the necessary conditions for a deadlock to occur?

   (b) (4 p) Describe using pseudocode a solution to the dining philosophers problem. You may use sempahores and read/write variables. Explain how this solution satisfies the requirements for exclusion (between neighbouring philosophers) and no-deadlock. Discuss the fairness properties of your solution.

   (c) (3 p) What is lock-free synchronization? What can it be good for?