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

Operating Systems
DIT 400, EDA092

Exam 2010-08-24

Date, Time, Place: Tuesday 24/8 2010, 14:00-18:00, V 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) There is a number of different RAID levels for using multiple disks as one unit. Describe how RAID 0+1 and RAID 1+0 works. (2p)

   (b) How is RAID 0+1 and RAID 1+0 affected by two erroneous disks? (2p)

   (c) Describe the SCAN and SSTF scheduling methods for optimizing head movement in disk memories. (4p)

   (d) Give two reasons why it is difficult for an operating system to improve the rotational delay when reading from a disk. (2p)

2. (10 p)

   (a) Unix systems use inodes to keep track of used blocks in a filesystem. Describe this inode based method for block allocation. (3p)

   (b) Consider a system where free space is kept in a free-space list. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list and in case it is possible, how could it be done? (2p)

   (c) Explain how a system call is done at assembly level (2p)

   (d) Which system call is used in Linux to create threads that execute in the kernel? (1p)

   (e) What happens when we have user-level threads mapped to a single kernel process and one thread issues a read while there are no data to read? What disadvantage does that imply? (2p)

3. (10 p)

   (a) When doing the original implementation of NFS, an extra layer called VFS (Virtual File System) was added to the system interface. Why was the VFS layer added? (1p)

   (b) In NFS, an encoding called XDR (External Data Representation) is used for data sent between client and server. What is the reason for using this encoding? (1p)

   (c) Describe two fundamental differences between NFS version 3 and NFS version 4. (2p)

   (d) Explain how paravirtualization works. (1p)

   (e) Give one advantage and one disadvantage with paravirtualization compared to traditional virtualization. (2p)

   (f) Explain why virtualization complicates the handling of virtual memory and how the problem is usually solved. (3p)

4. (10 p)

   (a) Describe the hardware support that is needed to support demand paging. (2p)

   (b) Assuming a page size of 4 Kbytes and that a page table entry takes 4 bytes, how many levels of page tables would be required to map a 64-bit address space if the top-level page table fits into a single page? Explain your answer. (3p)

   (c) Why are segmentation and paging sometimes combined into one scheme? (2p)

   (d) Consider a demand paging system with the following time-measured utilizations: processor utilization 20 %; paging disk 97.7 %; other I/O devices 5 %. What is the probable cause of thrashing?
   i. Repeated parity errors;
   ii. Disk crashes;
   iii. A local page replacement algorithm;
   iv. Process (e.g., being unable to establish their working set of pages);
   v. A FIFO page replacement algorithm.
   Explain your answer. (3p)

5. (13 p)

   (a) Describe Lamport's bakery algorithm for mutual exclusion among n threads. Argue for its properties regarding correctness, fairness and deadlock-freedom. (5p)

   (b) (i) What is the meaning of the term "busy-waiting"?
   (ii) What other kinds of waiting can there be in an operating system?
   (iii) Why do Solaris, Linux and Windows XP use spin-locks as a synchronization mechanism in multiprocessor systems and not on single-processor systems? (3p)

   (c) Can a multithreaded program using user-level threads achieve better performance on a multiprocessor system than on a single-processor one? Explain your answer. (2p)

   (d) Can a system be in a state that is neither deadlocked nor safe? If so, give an example. If not, argue why this is impossible. (3p)

6. (7 p)

   (a) Suppose a short-term scheduling algorithm favors those programs that have used little processor time in the recent past. Explain why this algorithm favors I/O-bound programs and yet does not permanently deny processor time to processor-bound programs. Name some example of an operating system where this approach is followed. (3p)

   (b) Consider three periodic real-time processes, P1, P2 and P3, with periods and processing times p1 = 40, t1 = 20; p2 = 30, t2 = 15; p3 = 25, t3 = 10. Can they be scheduled using Earliest Deadline First scheduling without missing their deadlines? Explain why or why not. If yes, show also the schedule; if not, explain whether they can be scheduled using another method. (4p)