Chalmers University of Technology and Gothenburg University

Operating Systems
EDA093, DIT401

Exam 2019-10-26

Date, Time, Place: Saturday 2019/10/26, 08.30-12.30, "Maskin"-salar

Course Responsible:
Vincenzo Gulisano (031 772 61 47),
Marina Papatriantafilou (031 772 54 13)

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

Grade-scale ("Betygsgränser"):
CTH: 3:a 30-39 p, 4:a 40-49 p, 5:a 50-60 p
GU: Godkänd 30-49p, Väl godkänd 50-60 p

Exam review ("Granskningstid"):
Will be announced after the exam.

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.
- Answer questions 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.

1. (12 p)

   (a) (4 p) Assume you rely on priority scheduling with multiple queues in an interactive system. All processes start with the highest priority. If they use up all the quanta allocated to them, they are moved down one class and their quanta are doubled. What would be the benefit of such a scheme?

   (b) (4 p). Explain why having one OS running independently in each core of a multi-core CPU can lead to data inconsistencies. Do **not** make examples.

   (c) (4 p). Prove that the Shortest Job First (non-preemptive) is optimal to minimize turnaround time.

2. (12 p)

   (a) (4 p) Describe at least 2 benefits of running several user-level threads mapped to the same kernel thread (many to one).

   (b) (4p) Write a C program that can be used to parallelize the search of an element inside a non-sorted array. Do **not** use `phtread_create` nor `phtread_join`.

   (c) (4p) Suppose you have a process composed of 3 threads. Make an example about a possible execution of such threads over time that shows the threads run both concurrently and in parallel.

3. (12 p)

   (a) (4 p) What are the trade-offs (pros vs cons) associated with the PageSize parameter?

   (b) (4 p) Consider the following reference string 0 1 2 3 0 1 4 0 1 2 3 4 of memory page accesses, and a physical memory of 3 frames that is initially empty. How many page faults will occur using (i) Fifo, (ii) Optimal, (iii) Least Recently Used (LRU) page replacement algorithm? In which of the previously cited algorithms (i, ii and iii), the number of page faults can increase if more frames are available (*Bélády's Anomaly*)?

   (c) (4 p) Let us consider a CPU with 4 cores, 32K of L1 cache, 256K of L2 cache, 4M of L3 cache, and a Translation Lookaside Buffer (TLB) with 1536 entries. Physical memory is 1GB, Virtual memory is 4GB, and PageSize is 4K. Explain briefly your answer to the following questions:

      i. How many entries has the page table of each process?

      ii. How many different pages can be accessed before a TLB miss occurs?

      iii. In our example, is it possible that the code and data corresponding to the working set of one process fits in the L2 cache but the system still experience TLB thrashing?

4. (12 p)

   (a) (4 p) Is "defragmentation" a good idea on a Solid-State-Disk (SSD)? Explain why.

   (b) (4 p) Let us consider a file system with 10000 files and where the distribution of file sizes on the file system is as follows:

Size in bytes	Proportion
0	10%
1 - 512	20%
513 - 4096	20%
4097 - 16KiB	40%
16KiB - 1GiB	10%

What would be the maximum amount of wasted space on disk in internal fragmentation when using a I/O Block size of 1024 bytes?

(c) (4 p)

i. We execute the following commands in a shell

```
user:machine$ touch bob
user:machine$ chmod 000 bob
user:machine$ ls -al
total 8
drwxrwxr-x 2 user groupA 4096 Oct 18 17:23 .
drwxrwxr-x 3 user groupA 4096 Oct 18 17:22 ..
---------- 1 user groupA    0 Oct 18 17:23 bob
user:machine$ rm bob
```

What is the result of executing the last command?

ii. How would you reverse `sudo chmod 000 \`which chmod\``? And what about `sudo chmod -R 000 /`?

5. (12 p)

(a) (4 p) (i) Describe the safety, progress and fairness requirements from a solution to the critical section problem.
(ii) Explain how it is possible to solve the critical problem for n threads, using binary semaphores. Describe the solution using pseudocode.
(iii) Argue about the properties of the solution that you describe with respect to the requirements.

(b) (8 p) Consider the readers-writers critical section (CS) problem: a shared resource, say a file, is accessible by several concurrent threads for reading or writing. Because the readers do not change the file, any number of them may be granted simultaneous read access. On the other hand, only one writer is allowed to write on the file at a time and the write should not be concurrent with any read access. Consider the following proposal for solving the problem:

```
shared var general semaphore access initialized to m;

reader-thread {
    repeat
        wait(access) ;
        // ... read-CS ...
        signal(access) ;
        // other operations
    forever
}

writer-thread {
    for k = 1 ... m do wait(access) end-for;
    // ... write-CS ...
   for k = 1 ... m do signal (access) end-for;
   // ... other operations ...
```

```
}
```

(i) Describe the safety, progress and fairness requirements from a solution to the problem.
(ii) Argue about the proposed solution with respect to the requirements, under the assumption that there are m reader threads and 1 writer thread that may need access to the file.
(iii) Argue about the proposed solution with respect to the requirements, under the assumption that there are m reader threads and $n > 1$ writer threads that may need access to the file.