Chalmers University of Technology and Gothenburg University

Operating Systems
EDA092, DIT 400

Exam 2015-01-14

Date, Time, Place: Wednesday 2015/01/14, **8:30-12:30, M building**

Course Responsible: Vincenzo Gulisano, Marina Papatriantafilou

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 30-39 p, 4:a 40-49 p, 5:a 50-60 p
GU: Godkänd 30-49p, Väl godkänd 50-60 p

Exam review ("Granskningstid"):
Friday 6th February, 12.00-13.00, Room 5128, EDIT building (west wing)

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.

Good luck !!!!

1. (12 p)

   (a) (3p) Discuss how the rate at which data is exchanged with a disk and the space utilization of the disk vary depending on the block size.

   HINT: Large block size → higher rate, lower utilization / Small block size → lower rate, higher utilization

   (b) (3p) Discuss why the first portion of a disk is reserved for the Master Boot Record (MBR) and what information the MBR contains.

   HINT: To discover available partitions, contains starting and ending points of each partition

   (c) (3p) Describe the pros and cons of the Linked-List allocation scheme with and without a File Allocation Table

   HINT: + Faster to traverse files to find block n, not using part of the blocks to point to next block

   HINT: - Consuming main memory

   (d) (3p) Given the following list of files:

   - *- r w x r - x r - x* *userA* *groupA* *file1*
   - *- r w x - - x - w x* *userB* *groupA* *file2*
   - *- r w x - - x - - -* *userC* *groupB* *file3*

   answer the following questions:

   - Can *userA* delete *file2*?
   - Which of the available files can be executed by *userC*?
   - Can *userB* execute *file3*?

   HINT: Assuming userX belongs to groupX: No All Yes

   HINT: Assuming userX does belongs to groupX: Yes All No

2. (12 p)

   (a) (2p) Why is beneficial to have the stack of a process in memory to grow “down” while the heap grows “up” (also in relation to the Virtual Memory mechanism).

   HINT: Maximizes space use, unused space is a hole –¿ less pages in physical memory

   (b) (3p) Describe the main steps required to handle a page fault.

   HINT: Trap to OS, check trap, load disk to memory, reset page table, restart instruction

   (c) (2p) Describe what is the purpose of page replacement algorithms

   HINT: Free memory if no frame available...

   (d) (5p) Given the following reference string, and supposing 4 frames (initially empty) are available: *7,2,1,2,0,1,1,4,2,3,0,7,0,1,2,0,3,0,0,3,0,3*
   Which is the minimum number of page faults that will occur?
   How many page faults will occur using the FIFO algorithm?

   HINT: The minimum (given by the optimal algorithm) is 7. FIFO is 11.

3. (12 p)

   (a) (3p) Describe the steps performed by the OS during a context switch from process 1 to process 2 and back to process 1.

   HINT: save state of process 1 in PCB 1, load state of process 2 PCB 2

HINT: save state of process 2 in PCB 2, load state of process 1 PCB 1

(b) (3p) Describe (line-by-line) the steps performed in the following code:

```
int main()
{
  pid_t pid;

  pid = fork();

  if (pid < 0) {
    return 1;
  }
  else if (pid == 0) {
    execlp("/bin/ls","ls",NULL);
  }
  else {
    wait(NULL);
  }

 return 0;
}
```

HINT: This code is presented and discussed in the book. Please refer to it

(c) (3p) Describe what are pipes used for and the difference between ordinary and named pipes.

HINT: Used to have 2 processes communicating. ordinary cannot be accessed outside the process that creates them (usually parent-child) named ones can be accessed without a parent-child relationship

(d) (3p) Name and describe the different parts of a process maintained in memory.

HINT: Text (the program code), Data - with global variables, heap and stack (with temporary data).

4. (12 p)

(a) (3p) A variation of the round-robin scheduler is the regressive round-robin scheduler. This scheduler assigns each process a time quantum and a priority. The initial value of a time quantum is 50 milliseconds. However, every time a process has been allocated the CPU and uses its entire time quantum (does not block for I/O), 10 milliseconds is added to its time quantum, and its priority level is boosted. (The time quantum for a process can be increased to a maximum of 100 milliseconds.) When a process blocks before using its entire time quantum, its time quantum is reduced by 5 milliseconds, but its priority remains the same. What type of process (CPU-bound or I/O-bound) does the regressive round-robin scheduler favor? Explain.

HINT: CPU bound get more time and higher priority, ie favoured. but IO bound not penalized (done in scheduling excercises)

(b) (3p) Explain two main issues that make scheduling in multiprocessor/-multicore systems a more complex problem compared to single-processor scheduling.

HINT: assignment of threasd/processes to processors and migration, dependencies/synchronization among threads; slides 42-46 scheduling

(c) (6p) Consider three processes, P1, P2 and P3, where p1 = 60, t1 = 20, p2 = 30, t2 = 5, p3 = 75, and t3 = 30. (a) Can these processes be scheduled using rate-monotonic scheduling? Illustrate your answer using a Gantt chart. (b) Can they be scheduled using the earliest-deadline-first (EDF) scheduling? Explain and illustrate your answer.

HINT: similar to 5.22 done in scheduling excercises; EDF schedulable iff Sum(t/p) less or equal to 1

5. (12 p)

   (a) (3p) Servers can be designed to limit the number of open connections. For example, a server may wish to have only N socket connections at any point in time. As soon as N connections are made, the server will not accept another incoming connection until an existing connection is released. Explain how semaphores can be used by a server to limit the number of concurrent connections.

   HINT: integer semaphore S, init N; accept connection: wait(S); release connection: signal(S); max N connections possible simultaneously

   (b) (3p)
   (a) Explain the four necessary conditions for a deadlock to occur in resource allocation among threads/processes.
   (b) Why is it possible to prevent deadlock by preventing one of these conditions?

   HINT: (a) slide 4 resource allocation-deadlocks (b) by definition of the term necessary conditions

   (c) (3p) Assume that a system has multiple processing cores. For each of the following scenarios, describe which is a better locking mechanism: (i) a spinlock or (ii) a mutex lock where waiting processes sleep while waiting for the lock to become available: (a) The lock is to be held for a short duration. (b) The lock is to be held for a long duration. (c) The thread may be put to sleep while holding the lock.

   HINT: a spin; b mutex or sleep depending on the cost of context switching compared to the spinning; c: mutex

   (d) (3p) Consider two threads A and B, which, after having computed some results need to write them in a file. We need to have the results of thread A written before the results of thread B. Explain how this can be done using a binary semaphore. Justify your solution.

   HINT: slide 25 synchronization