Chalmers University of Technology and Gothenburg University

Operating Systems
EDA092, DIT 400
Exam 2015-04-14

Date, Time, Place: Tuesday 2015/04/14, 14:00 18:00, "Väg och vatten"-salar

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"):
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.

Good luck !!!!

1. (12 p)

   (a) (3p) Discuss what is and the benefits of the Copy-On-Write mechanism on process creation.

   HINT: Parent and child initially share the same memory, pages are copied only when one the processes modifies them. Benefit: more efficient creation.

   (b) (3p) Given the following reference string, and supposing 3 frames (initially empty) are available: *7,2,0,1,1,0,1,2,0,8,3,0,0,3,1,4,2,3,0,7,2,0,8,3*
   How many page faults will occur using the Least Recently Used algorithm?

   HINT: 15

   (c) (3p) Describe the equal and proportional fixed frame allocation schemes.

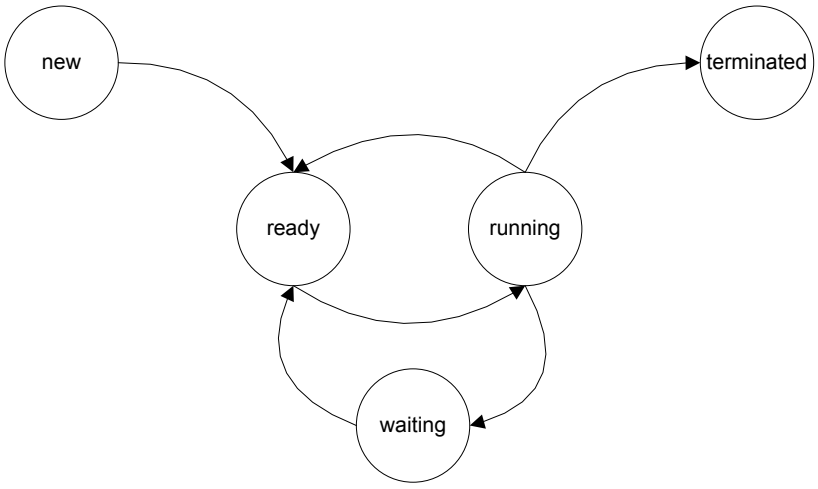
   HINT: Fixed: Each process receives the same number of frames. Proportional: the number of frames is proportional to the size of the process

   (d) (3p) What is the valid / invalid bit used for in the page table in the context of virtual memory?

   HINT: Check if a frame is referenced frame is actually in memory or if an interrupt should be used to bring the frame in memory.

2. (12 p)

   (a) (3p) Describe the process states presented in the figure and discuss when processes transit from one state to the other.



   HINT: Please refer to page 7 of class 2 (processes)

   (b) (3p) Consider a program that contains a serial portion S that covers 30% of it. Can you estimate the maximum speed up when such a program is run in parallel on 8 cores instead of 1?

   HINT: Given Amdahl's law, up to 2.58 times

   (c) (3p) Describe the difference between concurrent and parallel execution of a process.

   HINT: Concurrent: different threads of the same process make progress, but one at the time on the same CPU. Parallel: at the same time on different CPUs

(d) (3p) Present the main inter-process communication models and their tradeoffs.

HINT: Shared memory or message passing. First is area of memory under control of user processes, fast but needs synchronization. Second means process send messages to each other, no shared variables. No as fast, but less synchronization overhead.

3. (12 p)

(a) (3p) Discuss the pros and cons of keeping track of free blocks using the linked list or bitmap approach.

HINT: With linked list each disk block can keep track of multiple blocks, only keeping track of free blocks. With bitmap, each block is represented by a bit (less space) but need to keep track of all the blocks in the disk. Each is better depending on the overall number of free blocks.

(b) (3p) Discuss the pros and cons of the contiguous allocation scheme.

HINT: pros: only 2 numbers per file, good for read performance. cons: fragmentation, need to know in advance the size of the file.

(c) (3p) Discuss the basics of the interrupt mechanism.

HINT: Please refer to slide 17 of lecture 10 (I/O systems)

(d) (3p) In the context of I/O systems, what is the role of the Application I/O interface?

HINT: Abstract away the detailed differences in I/O devices by identifying a few general kinds. Encapsulate device behaviors in general.

4. (12 p)

   (a) (2p) Which of the following scheduling algorithms could result in starvation? Explain why. a. First-come, first-served b. Shortest job first c. Round robin d. Priority

   HINT: sjf, priority

   (b) (6p) Consider a system where you know that the offered load consists of periodic real-time tasks and interactive processes. As a system designer you are able to decide on the scheduling policy to use. Discuss the design of two scheduling policy alternatives suitable for such a system. (i.e. discuss how you would think in order to decide on a policy to use, the advantages and problems of the alternatives you are considering, how these methods could be implemented, whether you could you make use of additional information)

   HINT: one possibility: use EDF, with dedlines for interactive jobs = time of issue + maximum reasonable response time for the user (if available)
   other: maintain 2 queues, one for RT, one for interactive, serve the RT first with EDF or RM, when empty move to interactive. If use EDF in the RT and the offered RT load (sum[(exec-time)/period] is less than 1, there will be time left for interactive ones, else, there may be starvation to interactive jobs.
   alternative: insert the interactive jobs with fixed priorities in the same queue as the RT tasks. fixing those priorities would need knowledge of how critical it would be to miss an interactive process (i.e. to let it starve, or to eliminate it from the queue in case of congestion) and what are the tolerated response-time margings for these processes.

   (c) (4p) An issue in multiprocessor scheduling is how to design the ready queue. Describe this problem and its common solutions.

   HINT:shared versus per-processor; can facilitate loab lanacing but can become hotspot + no-processor-affinity can result in large migration costs

5. (12 p)

   (a) (6p) Design a solution to the mutual exclusion problem for arbitrary number of processes/threads in a system with SPARC processors, where the following *atomic instruction*, called Compare-and-Swap, is available by the hardware. Discuss carefully the properties of your solution.

```
int CAS(int *addr, int old, int new)
  if (*addr == old) { *addr = new; return(SUCCESS) }
  else return(FAILURE)
```

   Answer sketch: One can use CAS to immitate the behaviour of test-and-set or exchange instructions and design a solution similar to the ones we discussed in that context. the solution ensures mutual exclusion, no deadlock, but may suffer from starvation. A solution free from starvation employs an idea as in the n-process mutex-algo by Peterson. Answers along the first direction that discuss the starvation possibility, are also accepted as fully correct.

   (b) (4p) Consider the following suggested solution to the readers-writers problem. Does it correctly solve the problem? Argue why or why not.

```
int readcount; // (initial value = 0)
semaphore w; // ( initial value = 1 )

//READER
            readcount++;
            if (readcount == 1)
              wait(w);
        // reading is performed
          readcount--;
          if (readcount == 0)
            signal(w);


//WRITER
        wait(w);
         // writing is performed
        signal(w);
```

Answer sketch: readcount is not protected when ++ or –, hence a race consition can cause the if statement to be false even through there can be eg. 2 readers and a writer and thus cause violation of mutual exclusion between the readers and the writer.

(c) (2p)What is the meaning of the term "busy-waiting"? What other kinds of waiting can there be in an operating system?

Answer sketch: spinning + block