

Operating Systems, midterm exam, 8.3.2017

Write in each answer sheet course name, date, your name, signature and student id.

For each question, it is sufficient to give a 1-2 page answer.

Please write the answer to each problem in its own answer sheet.

Each problem is also a miniexam of the same number. Problem 4 is miniexam 4.

NOTE: Please return the answer for each problem in its correct pile!

1. [6 p] Cache.

- a. [1 p] Give examples on spatial locality in code and data references in program code.
- b. [1 p] Give examples on temporal locality in code and data references in program code.
- c. [2 p] 32 KB cache contains 1024 8-word (32 byte) lines. What would be the effect on cache performance if the cache would instead have 256 32-word lines? Under what circumstances would this be better and why?
- d. [1 p] The system has 4 cores, and register reference time is 1 ns. Each core has its own cache (L1). L1 cache has size is 32KB, access time 2 ns, and hit ratio 99%. The main memory is 8 GM and its access time 60ns (for the cache line). What is the average memory access time?
- e. [1 p] Continuation to d. Another level (L2) is added to the cache. All cores share cache L2, which has size 4 MB, access time 5 ns, and hit ratio 90%. What is now the average memory access time?

2. [6 p] Semaphores.

- a. We have a 4-threaded program P, where all 4 threads increase the value of shared variable X (initialized to 0) with the following code:

```
Thread T1 (threads T2, T3, T4 similarly)
int a, b;      /* local variables in thread T1 */
...
i1:  a = X;
i2:  a++;      /* a = a+1 */
i3:  X = a;
i4:  b = a;
...
```

Give a scenario, where the value of X is erroneously 3 at the end of P's execution.

- b. [2 p] Continuation to part a. How should one modify program P, so that it would always work properly (value of X at the end is always 4)? Incrementing X critical and it can not be delayed for a time to do even one process switch.
- c. [2 p] Editor and keyboard driver. Text editor TE reads character buffer B one character at a time, and then makes the changes (based on that character) to the file being edited. Keyboard device driver DD reads the pressed keys (one at a time) and then writes the corresponding characters to the buffer B. Buffer B contains 200 characters.

Routines Put(buf, c) and c=Get(buf) are used to move data into the buffer and from it. Concurrency problems within routines Put() and Get() have been solved in them.

Give the solution to this synchronization and communication problem for TE and DD using semaphores and buffer (B) in shared memory. Present the solution with pseudocode for TE and DD. Declare clearly all your semaphores and other data structures (needed for synchronization)

with their initial values.

3. [6 p] **Monitor and deadlock.**

- a. [2 p] How does a monitor condition variable and its operations differ from a semaphore and its operations?
- b. [2 p] What does the concept "Lampson & Redell signalling semantics" mean? How (and why) do you need to consider it in the monitor code?
- c. [2 p] How can you implement possible overlapping critical sections so that deadlock is just not possible? You can keep a critical sections reserved when it is not used. How can you prove that your solution is correct and does not ever cause a deadlock?

4. [6 p] **Virtual memory.** A 2-level paging virtual memory has 32-bit byte addresses and page size 4 KB.

- a. [2 p] How is the address translation from virtual address 0x11223344 to physical address 0xABCD3344 done?
- b. [2 p] How long roughly will the address translation take in different cases?
- c. [2 p] The Clock algorithm selects a page to be replaced from those frames currently in use. Why is Clock better than FIFO? Why is Clock better than LRU? What does "better" mean in this context?