# Exam in EDAF15 Algorithm Implementation

## August 19, 2011, 08-13

Inga hjälpmedel!

Examinator: Jonas Skeppstedt, tel 0767 888 124

30 out of 60p are needed to pass the exam.

1. (10p) Explain why a five-stage pipelined RISC processor usually does not see a five-time speedup when running most programs. Describe a program that can come close to a five-time speedup.

   ***Answer***

   - *pipeline stalls due to taken branches,*
   - *pipeline stalls due to sequences such as a load directly followed by use of the fetched value,*
   - *even more delay due to cache misses,*
   - *instructions such as divide which take multiple cycles in the execute pipeline stage.*

   *A program which can come close to a five-time speedup is one which does not suffer cache misses, and none of the other problems above, which sometimes can be achieved by loop unrolling and many integer computations.*

2. (5p) A set-associative cache has multiple comparators which can work in parallel. What are they comparing and why are fully associative caches not used for instructions or data caches?

   ***Answer*** *For each cache line in the set, they are comparing the address of the stored cache block with the requested address (actually, to be more precise, for a block size larger than one byte, they are comparing the requested cache block number with the stored cache block number — the cache block number is the address divided by the cache block size (but no division is used, of course, but only the bits of the address containing the cache block number is used)).*

3. (5p) What does the term cache block size refer to? Explain why a cache block size of $1$ certainly is too small and of $1024$ probably is too large.

***Answer*** *The cache block size is the number of bytes that are copied from memory to the cache at a cache miss. To benefit from spatial locality it should usually be at least 32 bytes but using 1024 leads to a high risk that only a few of the 1024 bytes really are useful and since they are fetched anyway there is a large risk that some other data in the cache must be overwritten which thus leads to more cache misses.*

4. (20p) Implement a linked list of integers with the following functions:

   - (3p) a function `new_list` to create a new and empty list.
   - (3p) a function `free_list` to deallocate the list.
   - (3p) a function `length` to compute the length of the list.
   - (3p) a function `insert_first` to add an integer at the beginning of the list.
   - (3p) a function `insert_last` to add an integer at the end of the list.
   - (5p) a function `reverse` to reverse the list.

   Only `new_list` and `length` are allowed to have a return value (the others are `void` functions). No memory may be allocated from the heap by `reverse`. You decide whether you want to use a single or double linked list, and whether you want to have a seperate header node in which no integer is saved. When you call a function which can fail, you must check the return value, and print an error message and terminate the program upon failure.

   ***Answer*** *See book.*

5. (10p) Explain what the strengths are of each of the following tools when tuning a C program:

   (2p) `gcov`

   ***Answer*** *Needs special flags for the compilation and affects compiler optimization. Counts exact execution counts for each source line as well as branch frequences (taken/non-taken).*

   (2p) `gprof`

   ***Answer*** *Same disadvantages as for* `gcov`*. Advantages include that execution times are shown for the different functions and that the number of times each function is called and by which function.*

   (3p) `Oprofile`

**Answer** *Oprofile does not need any special compiler flag and can (should) be used with the highest compiler optimization. Other advantages include that it can count the number of times certain events happen such as clock cycles, executed instructions, cache misses. This information can be presented at the source code line level or instruction level. A disadvantage is that to change what it should count, the user must have root privileges on the machine.*

(3p) `valgrind`

**Answer** *Valgrind does also not require any special flag but it works best when debugging information is available. A disadvantage is that it is slower than the other tools (e.g. 100 times slower execution). Advantages include that it can detect many kinds of pointer and memory allocation errors. It can also produce statistics of heap memory usage and cache performance.*

6. (5p) Modify your solution to question (4) in order to make it probable that the number of calls to `malloc` and `free` is reduced.

   **Answer** *Use a free-list — but the solution must be an implementation such as you did in a lab, and is shown in the book.*

7. (5p) Instead of dividing a variable of type `int` by a power of two, it is much faster to use a shift instruction. That is not always sufficient, however. Explain why and what you can do about it.

   **Answer** *See book on page 117.*