



Final words and test questions examples

NSWI170 Computer Systems

Jakub Yaghob, Martin Kruliš

Refreshing course contents



- C++
 - Will not be tested directly, but you may need it to understand some questions
- CPUs
 - Principles of instruction encoding and execution
 - Basic instruction types (arithmetic, conditions, jumps, memory operations, ...)
 - Representing higher-language constructs in assembly
 - Registers, memory operations
 - Hardware architectures fundamentals



Refreshing course contents

- Memory
 - Fundamentals (addressing, endianness)
 - Representing data structure (alignment)
 - Memory allocation (principles, algorithms)
 - Hardware details (caches, NUMA)
- Programming languages
 - Compilation, linking, dynamic linking, loading for execution
 - Memory organization (globals, constants, heap, ...), relocations
 - Calling conventions
 - Portability, virtual machines, garbage collector

Refreshing course contents



- Operating systems
 - Fundamentals, architectures
 - Devices (controller, driver, communication), interrupts
 - Process, thread, scheduling (principles, multitasking , algorithms)
 - File systems (files, directories, FS representation)
 - Hardware-related issues of HDD and SSD (request scheduling, wear leveling, RAID)
 - Virtual memory (principles, segmentation, paging)
 - Including page replacement algorithms, file memory mapping, and shared memory
- Parallel processing and synchronization
 - Race condition, deadlock, sync. primitives, basic sync. problems

Consultation



Exams



- General information
 - Dates are in SIS (will become available 18.5., after some reduction)
 - There will be no other dates, unless we detect a major global problem
 - The credit is not required for the test
 - Since lectures and labs were separated
 - Exam is focusing on understanding, not memorizing low-level details
 - Slides are not a complete reference of knowledge
 - Memorizing the slides/lecture will probably not help you
 - Some questions will require you to apply the knowledge from the lectures
 - E.g., to compute something, apply an algorithm and report the result, ...
 - Prior knowledge from NSWI120 (Principles of computers) may be required!

Exams



- Test structure
 - In the labs using online application (still experimental)
 - Strict rules, no materials, no other applications may be running, ...
 - 10 questions, 20 points, 65 minutes (soft deadline)
 - Grading
 - 17-20 points = mark 1 (Excellent)
 - 14-16 points = mark 2 (Well done)
 - 11-13 points = mark 3 (OK)
 - < 11 points = fail

Exams



- Question types
 - Single best answer (radio select)
 - Multiple answers may technically correct, but you need to select the best one
 - The best one is the most precise and the most complete one
 - Multi-choice (checkboxes)
 - Multiple answers may be correct, you need to select all of them
 - There may be no correct answers (not checking anything may be correct)
 - Fill in number(s)
 - The answer is a number or sequence of numbers (e.g., a result of a calculation)
 - New types may be added
 - E.g., put given tokens in the right order

Exams



- Quiz questions
 - 8 questions, 1 point each
 - Example (single best answer question)
 - “Instruction set architecture (ISA) defines:”
 - Instructions semantics and registers
 - Instruction op-codes, memory management, and I/O model
 - Registers, ALUs, decoding units, caches, and memory controllers
 - Instruction mapping to micro-instructions and ALU operations
 - Instructions semantics, op-codes, registers, memory management, and I/O model
 - Instructions semantics, op-codes, ALUs, decoding units, registers, memory management, caches, and I/O model

Exams



- Practical questions
 - A scenario, where a problem is given and you need to solve it
 - Calculate something, decode a data structure, understand a piece of code, ...
 - 4 questions, 3 points each
 - Example (numeric question)

“How many page faults (at most) may occur when you copy 2KiB of memory in user space?”

 - Assume 32-bit address space, 4 KiB pages, and 2-level paging
 - There is enough physical memory (each page may fault at most once)
 - OS is “simple”, it can allocate only one physical frame at each fault
 - Ignore possible page faults caused by instruction fetching

Exams



- Practical questions
 - Example (numeric question)

```
struct S {  
    char type;  
    double deviation;  
    int count;  
}  
S[20] sa;
```

“What is the offset (in bytes) of **sa[3].count** from the beginning of **sa** array? (**double** is 64bit, **int** is 32bit)?”

Exams



- Practical questions

- Example (numeric question)

- FAT16 file system with 2 KiB clusters
 - File **a.txt** directory entry has properties
 - firstCluster = 3
 - size = 7654
 - An user performs
 - Seek to offset 3210
 - Read of 1000 bytes

FAT

2	0	3	10	4	0	5	-1
6	0	7	0	8	5	9	0
10	15	11	0	12	0	13	8
14	-1	15	14	16	0	17	0

- “Write indices of all clusters which will be loaded to satisfy this operation. (assume the FAT table is cached in main memory).”

Exams



- Practical questions

- Example (single best answer)

“Select a C code that was compiled into the given assembly fragment.”

- **if** (x < y) ++x;
- **s += (a-b) * (a-b)**
- **a + b + c - d**
- **(a+b) - (c+d)**
- **(x >= y) ? 0 : x+1**
- **return** x * x
- **while** (j > 0) --j;

```
add $t0, $a0, $a2 // addition
add $t1, $a1, $a3
sub $r0, $t0, $t1 // subtraction
```

Discussion

