

T-106.1150 Computer and operating system

The exam contains six questions. The maximum points for each question are listed in the beginning of the questions. Read the questions carefully. Give clear and compact answers. Remember to write the name of the course and your own personal information on each of your answer papers.

- 1 (6p) Give short and clear definitions for the following. Do not use long explanations or examples. (One point per question.)
 - a) What are Karnaugh maps?
 - b) What are finite state machines?
 - c) What is a Wallace tree?
 - d) What is DMA?
 - e) What is big-endian?
 - f) What is virtual memory?
- 2 (6p) By using full adders, design and draw a 4-bit adder. Explain how it operates both for positive and negative numbers.
- 3 (4p) Explain the execution cycle of a typical processor.
- 4 (4p) What is an operation system kernel? How does it work?
- 5 (6p) Why clocks are so important for digital logic? Write an essay on the issue that is not longer than 45 lines.
- 6 (6p) Write a MIPS-program, using the instruction set on the other side of the paper, that reads a sequence positive integers (terminates at a non-positive one), and sum of squares of the positive integers. order. Given the sequence of integers 1, 2, 3 and -1, the program should output 14.

Comment your code to make it more readable. Explain when and for what your program uses the operating system.

R instruction structure:

Label: INSTR R1, R2, R3

where INSTR the instructions symbolic name
R1 storage register
R2 first operand register (optional)
R3 second operand register (optional)

I instruction structure:

Label: INSTR R1, ADDR(R2)

where INSTR the instructions symbolic name
R1 storage register (register R0..R7)
ADDR memory address (optional)
R2 address offset register (optional)

J instruction structure:

Label: INSTR ADDR

where INSTR the instructions symbolic name
ADDR memory address

Arithmetic instructions:

- add, addi, sub, sub, subi, mul, div
- and, andi, or, ori, xor, nor, slt, slti
- sll, slr, sra

Storage instructions:

- lw, sw
- mfhi, mflo

Jump and branch instructions:

- j, jr
- beq, bne, bgt, bge, blt, ble

Other instructions:

- nop, syscall

Assembler directives:

Code Action

.data Begins programs data part
.text Begins programs instruction part
.word N Defines 32-bit variable in memory with initial value N
.word N1..Nn Defines n successive variables in memory with given initial values

SPIM system calls:

Integer code in \$v0 determines action of syscall instruction

Code Action

1 prints integer stored in \$a0
5 reads integer and stores it in \$v0
10 halts the program