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 is RISC?
b) What is a trap?
c) What is a TLB?
d) What is page eviction?
e) What is big-endian?
f) What is write back?

2 (6p) By using the basic gate elements (and, or, not, etc.), draw a picture of a full adder. Explain how it operates and what it can be user for.

3 (4p) Explain addressing modes of a typical processor.

4 (4p) Explain what are the main tasks of the process management of a typical operating system.

5 (6p) Considering interrupt handling by operating systems, write an essay that is not longer than 45 lines.

6 (6p) Write a MIPS-program, using the instrucion set on the other side of the paper, that reads two integers (lets call them A and B) form the user and prints all multiples of A from A to A * B. Given the integers 3 and 4 the program would thus output the integers 3, 6, 9 and 12. You may assume that the read integers are positive.

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 instrucitons:

- add, 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 instrucions:

- 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 inital 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
```