CS221 Assembly Language

Lab5

Prof. Mohamed Ben Othman

1- **Practice**:

Conditional Structures: even if you may have not studied them yet, this lab will cover at a certain level this area. Conditional instructions are generally used for loops (for, while …) and conditional statements (if, switch). These instructions are jump and loop instructions in different formats. There are some other (few) specific instructions which will not be covered in this lab.

loop instructions: You already saw the loop instruction which is using the register ***ecx***: it decrements this register and if it is not 0 it branches (jumps) to the label.

There are some other loop instructions:

all loop instructions have the same syntax: the loop instruction followed by a label to jump to.

1- loopz (or loope): branches if the flag Z(ero) is set
2- loopnz (or loopne): branches if the flag Z(ero) is cleared

Jump instructions: You already saw ***je*** and ***jne*** in previous labs. All instructions in this type have the same syntax: the jump instruction followed by a label to jump to when the condition is met. You saw also, ***jmp*** which jumps always without condition.

All these instructions, except ***jmp***, use the current values of some flag bits in the flag register to jump. These flags are always set and cleared upon the result of the last executed operation (add, sub, cmp, …). the cmp instruction has to operands (***cmp*** leftop,rightop). This latter is executed by doing the subtraction leftop-rightop and only the flags are modified: Z is set if the result is zero and cleared if not, S (sign) is set if the result is negative and cleared if not, C (carry) is set if there is carry and cleared if not … The same process is done with other instructions. Tables 1,2, and 3 give the most used jump instructions.

Table 1: Jumps Based on Specific Flags

Mnemonic	Description	Flags
JZ	Jump if zero	ZF = 1
JNZ	Jump if not zero	ZF = 0
JC	Jump if carry	CF = 1
JNC	Jump if not carry	CF = 0
JO	Jump if overflow	OF = 1
JNO	Jump if not overflow	OF = 0
JS	Jump if signed	SF = 1
JNS	Jump if not signed	SF = 0
JP	Jump if parity (even)	PF = 1
JNP	Jump if not parity (odd)	PF = 0

Table 2: Jumps Based on Equality

Mnemonic	Description
JE	Jump if equal (*leftOp* = *rightOp*)
JNE	Jump if not equal (*leftOp* ≠ *rightOp*)
JCXZ	Jump if CX = 0
JECXZ	Jump if ECX = 0

Table 3: Jumps Based on Signed Comparisons

Mnemonic	Description
JG	Jump if greater (if *leftOp* > *rightOp*)
JNLE	Jump if not less than or equal (same as JG)
JGE	Jump if greater than or equal (if *leftOp* >= *rightOp*)
JNL	Jump if not less (same as JGE)
JL	Jump if less (if *leftOp* < *rightOp*)
JNGE	Jump if not greater than or equal (same as JL)
JLE	Jump if less than or equal (if *leftOp* <= *rightOp*)
JNG	Jump if not greater (same as JLE)

There are some other jump instructions called unsigned (use mainly ***above*** and ***below*** notations).
In this lab you are asked to practice these instructions:

Write a small program that asks the user to enter two integers (ReadInt puts the entered value in eax), put

them in different registers, compare them and use all jump instructions to show a message (taken from the second column of the table: example "*leftopr is equal to rightopr*"
…
try to loop in this program forever using *jmp* (until you force stop by ctrl-c).

Exercise:

If you practice well, you are ready to do this easy exercise: as you may know there are different switch instruction formats in different programming languages. Some give more flexibility than others:

In C like programming languages	Python

switch (expr)
{
case value1: stmts;
case value2: stmts;
…
default: stmts;
}
	if condition1: stmts
	elif condition2: stmts
	…
	else: stmts

Note: stmts can contain any statement including break.

You are asked to mimic these instructions in assembly language: The (C like) *switch* is less flexible than the python's one. It is using only equality (you can use ***cmp*** and then ***je*** (***jz***) or ***jne*** (***jnz***) and ***jmp***) for the expr use only a value that is entered by the user and for the stmts only print a message that is giving the case number.

Pyhon is more flexible: you can use all jump instructions by asking for two integers, compare them, and give a message that tells which condition is met. Note that for specific condition (like JCXZ, JECXZ) use the specific register.

```
Note: You should practice and do the exercise home and, in the Lab time: the first
half the instructor will check your homework, take notes, and answer your questions
if you have any. The second half will be a quiz.
```

2- Quiz:

A Quiz will be given in class this week.