

LAB 3: Types and Number of Operands

Objectives:

- a) Understand the various types of common operands used in assembly language.
- b) Understand why various operations require a different number of operands.
- c) Investigate various operations and their operands.

- 1) Background: In this lab you will investigate aspects of various operands. We will consider zero, one and two operand instructions.

Consider the default program in Frances-A. The first instruction of the assembly code is

```
lea 0x4(%esp), %ecx
```

The *lea* is the operation. It is followed by two operands, *0x4(%esp)* is the source and *%ecx* is the destination. It is taking information from the source and using it in the destination. The next two instructions are also two operand instructions. The fourth instruction is a one operand instruction.

```
push %ebp
```

This instruction is implicit to the stack. In other words, it is using the operand (*%ebp*) and using the stack as well. In this case it is putting the value located at *%ebp* on the stack. Finally, the last instruction is a zero operand instruction. It is a simple command and no operands are needed. In this case, *ret* simply returns from the program (ends this program).

- 2) Exercises:

- a) Go to the Frances-A website and compile the default program.
 - i) Step through the program and list the number of bytes given to each instruction.
 - ii) List the location of each instruction as given in eip.

- b) Type the following code in the Frances-A code window.

```
int main(){
    int x=1;
    float y=3;
    if((x+1 <2)&& y<-1)
        x=x+y;
}
```

- i) Step through the program and list how many bytes are given to each instruction.
 - ii) Where is the newLabel3 relative to the rest of the code?
- c) In these code examples how many bytes are given to zero, one and two operand instructions?