Assembly Language and System Programming Lab5 Report

Group: 2

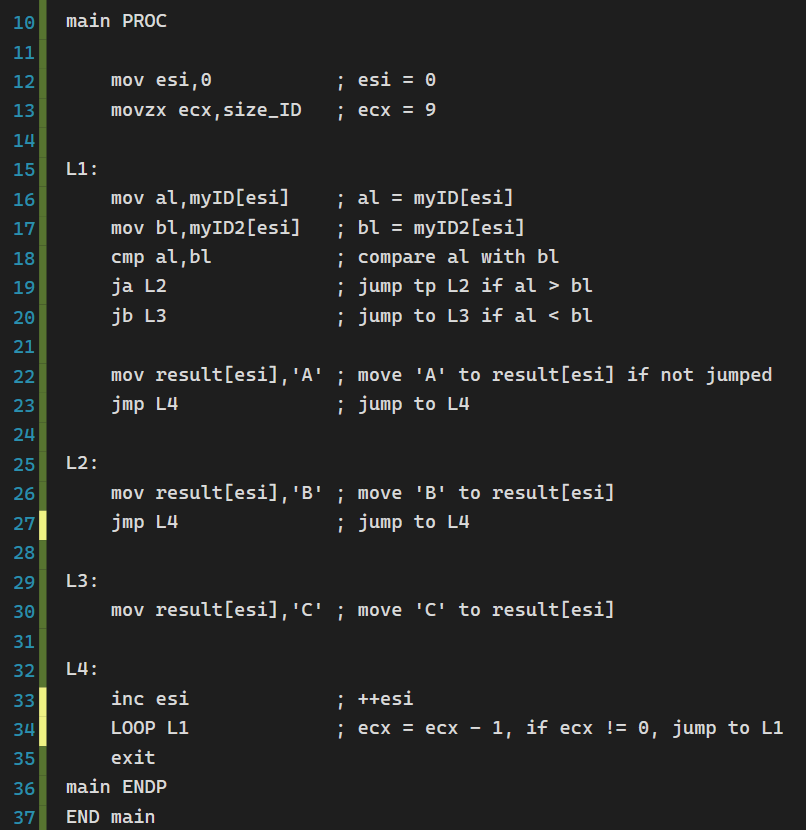
Names: 林柏廷(Brian)、蔡淵丞(Vincent)

Student ID: 110502531、110502567

**Objective:**

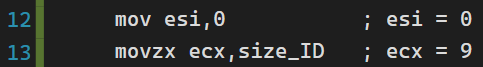
Compare the numbers of myID and myID2. If they are equal, the resulting number will be ‘A’. If the number of myID > myID2, the result will be ‘B’, else the result will be ‘C’.

**main PROC:**

****

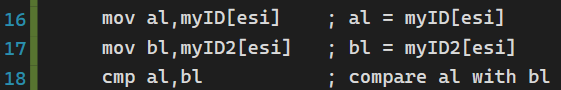
**Before The Loop:**

Use ***esi*** as an index of the strings. Let ***ecx*** equals to the size of the strings.

****

**L1:**

Store the number of myID and myID2 we’re currently at into ***al*** and ***bl*** respectively. Then, we compare ***al*** with ***bl***, the operator cmp would result in the change of flags.

****

**If al > bl:**

The operator ja would check the flags to decide whether ***al*** is greater than ***bl***. If it is, jump to L2, else continue the program.

****

**If al < bl:**

The operator ja would check the flags to decide whether ***al*** is less

than ***bl***. If it is, jump to L3, else continue the program.



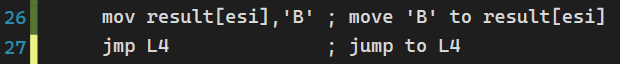
**If al == bl:**

Store ‘A’ to result[***esi***], and jump to L4 to avoid entering L2 and L3 since they are used for different conditions.



**L2:**

Store ‘B’ to result[***esi***] since ***al*** > ***bl***, also jump to L4 to avoid entering L3.

****

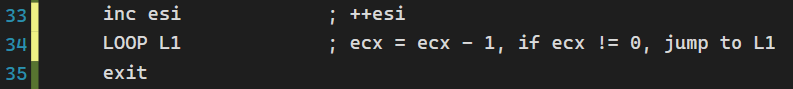
**L3:**

Store ‘C’ to result[***esi***] since ***al*** > ***bl***.

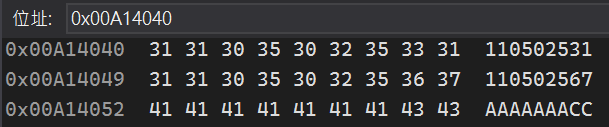


**L4:**

Increase ***esi*** by one to get the next number in our strings. Jump back to L1 to process the next number.



**Results:**

****

**Review:**

A conditional jump instruction jumps to a label if certain condition of register or flag are met.

In this lab, we used ja and jb combined with cmp to compare numbers, but there are actually many different conditional jump operators to use.

**Jz**: jump if zero flag is 1 (a == b)

**Jc**: jump if carry flag is 1

**Ja**: jump if zero flag is 0 and carry flag is also 0 (a>b) (for unsigned)

**Jb**: jump if zero flag is 0 and carry flag is 1 (a<b) (for unsigned)

**Je**: jump if zero flag is 1 (a=b)

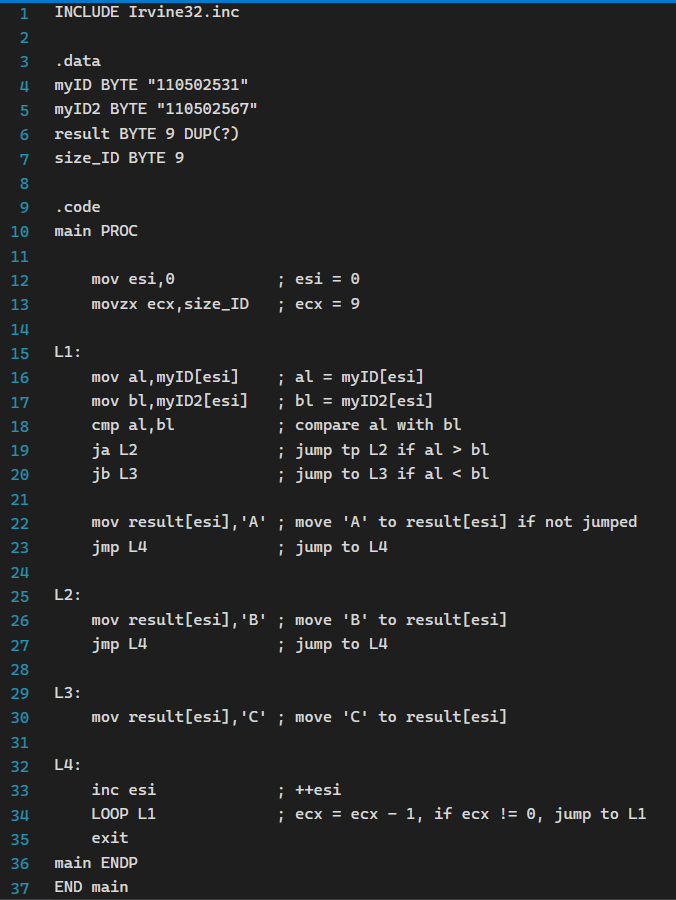
**Jg**: jump if zero flag is 0 and carry flag is also 0 (a>b) (for signed)

**Jl**: jump if zero flag is 0 and carry flag is 1 (a<b) (for signed)

**JnX**: jump if not

Notice that there are different instructions for signed and unsigned number. Also, we need to be extra careful if we are using both jump and loop, or it may lead to unexpected execution of the program.

**Full code:**

****