**In-Class Exercise # 6 – “Start Assembly Programming”**

Due Day: 2017/03/23, Friday, 12:00

Objective: To understand how to use conditional JUMP in a simple Assembly program.

Explanations：

1. Use the course content to learning the memory changes and how to use conditional JUMP instruction. Analyze data in myID, store how many odd numbers in the myID to myID\_odd, store summation of even number in the myID to myID\_even, and myID\_result is the result of multiplication (using addition and loop) of myID\_odd and myID\_even.

|  |
| --- |
| .data  myID byte \_, \_, \_, \_, \_, \_, \_, \_, \_ ; Student ID  size\_ID = ($-myID) ; size\_ID is length of myID  myID\_odd \_\_\_\_\_ ?  myID\_even \_\_\_\_\_\_\_ ?  myID\_result \_\_\_\_\_\_\_ ?  .code  start@0 PROC    [do something...]    mov \_\_\_\_\_\_\_, \_\_\_\_\_\_\_ ; save to myID\_odd  mov \_\_\_\_\_\_\_, \_\_\_\_\_\_\_ ; save to myID\_even  ; multiplication using addition and loop  mov \_\_\_\_\_\_, myID\_odd; mov myID\_odd to a register  mov \_\_\_\_\_\_, myID\_even; mov myID\_even to a register  ; set myID\_odd as looping count  [do something...]  L:  add \_\_\_\_\_\_\_, myID\_even; add myID\_even to the register  [do something...]  loop L  mov myID\_result, \_\_\_\_\_\_\_; save the multiplication result  exit  start@0 ENDP  END start@0 |

1. Use WINdbg to show registers and memory status when the program executed and add screenshots to the report in Word files.
2. Compress(.zip,.rar) the following file with the name of the group (e.g. group\_1.zip)
   * 1. Code(**exercise6.asm**)
     2. Report (**group\_1**.doc/.docx or **group\_1**.pdf)
        1. Report Title
        2. Group, name, student ID
        3. Program execution flow, memory (register) status
        4. Screenshots description code Description
        5. Reviews
3. To update the contents of the report, you can directly re-upload the file with the name of the new version (e.g. Group\_1\_v2.zip)

Note：

1. Each group, one report

Upload the report to (<http://lms.ncu.edu.tw/ncu>) before Friday, 12:00.