HW2-Procedure

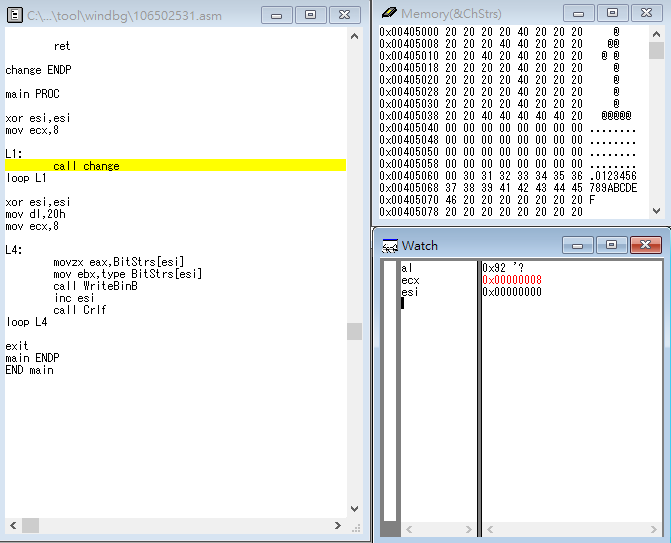
106502531 洪仲杰

Code description:

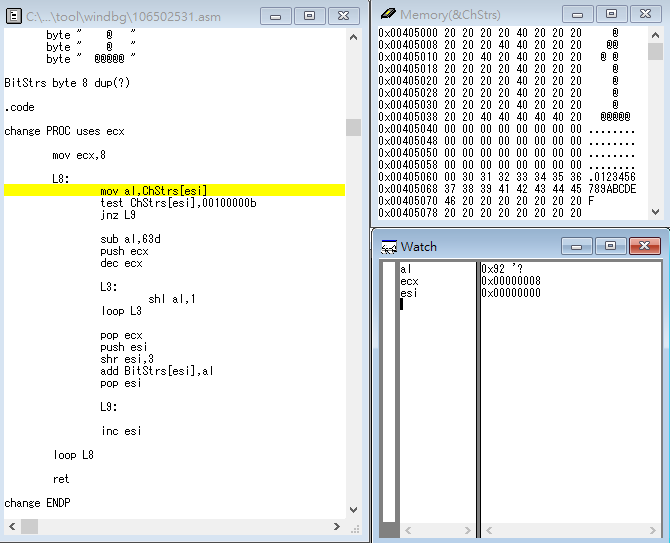
1. Initial state.

Set esi = 0, ecx = 8.

Call change procedure.

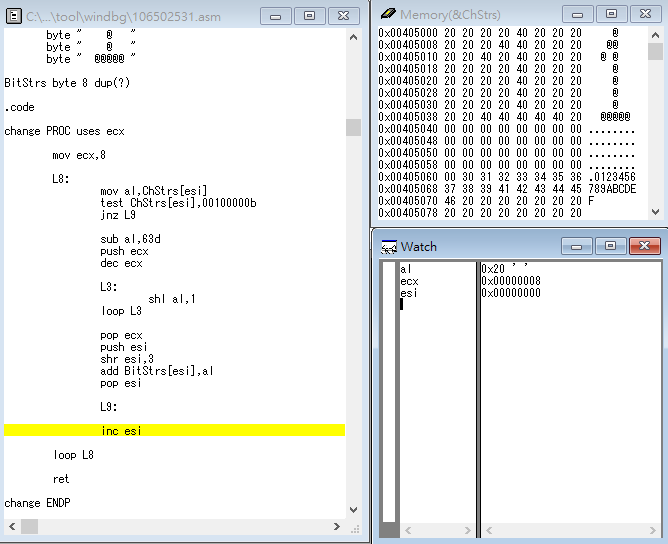


1. Uses ecx, set ecx = 8.



1. Let al = ChStrs[esi] (esi now equal 0).

Test if (ChStrs[esi] == (space)), true then jump to L9.

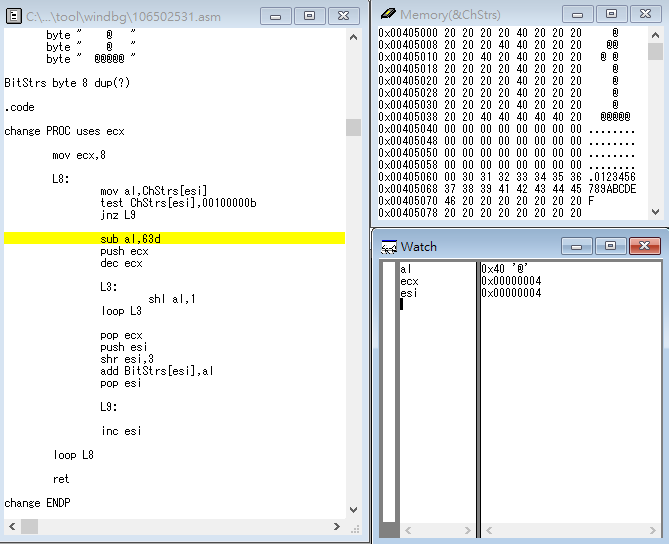


1. esi ++

loop L8 until ChStrs is not (space).

Test if (ChStrs[esi] = = (space)),

false then keep going.



1. ChStrs[esi] is ‘@’ (not (space)),

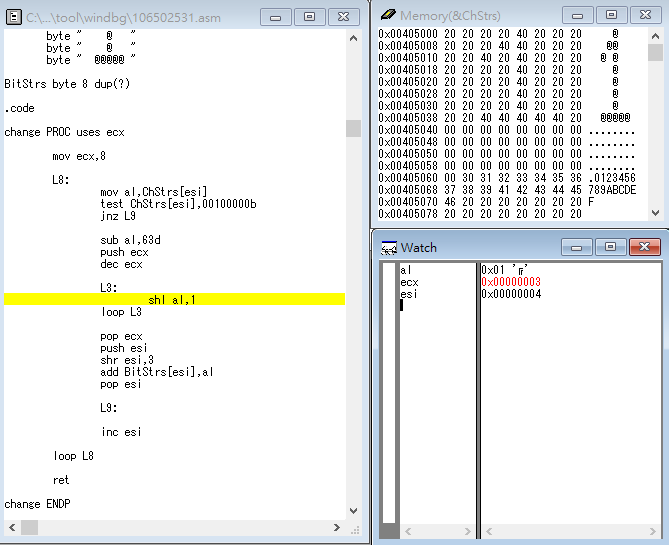
al (64d) - = 63d (equal 1).

Push ecx - 1.

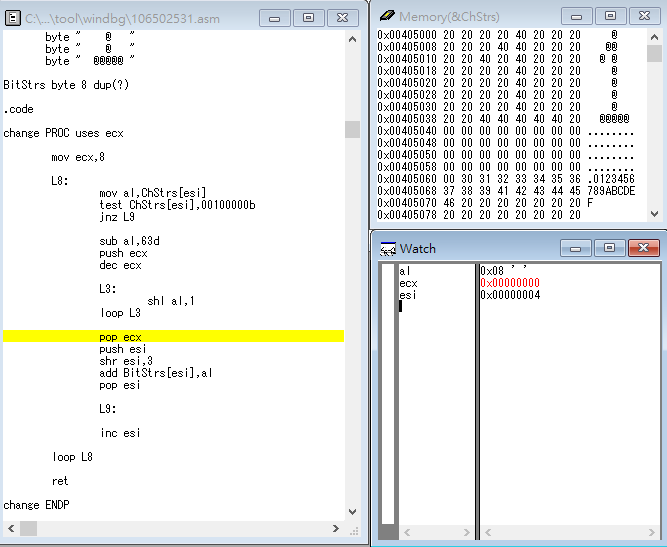
al times 2 and loop L3 until ecx is 0

(esi point from left to right,

but bit increase from right to left).



1. ecx = 0 and break.



1. Pop ecx.

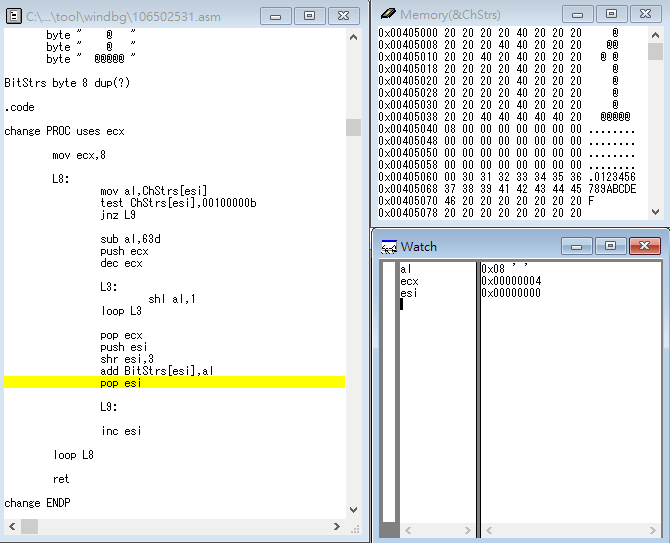
Push esi.

esi divide into 8 (because BitStrs has only 8 bytes).

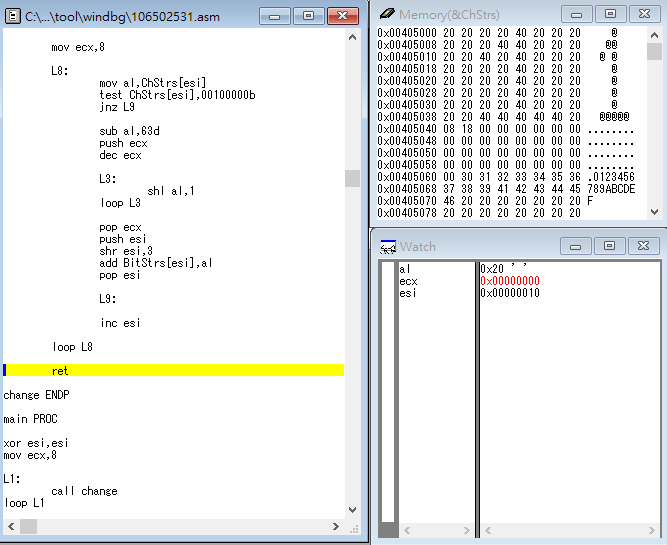
BitStrs[esi] + = al.

Pop esi.

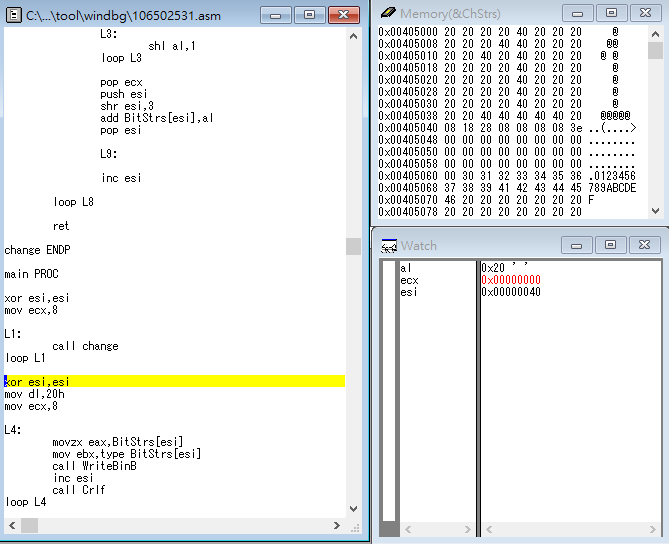
Loop L8 until ecx = 0.



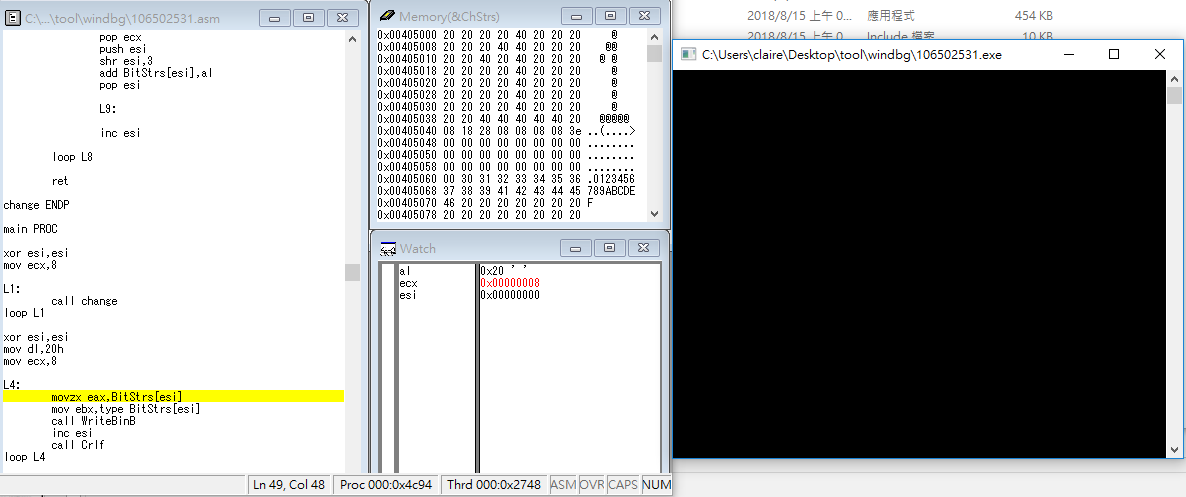
1. The state of the 2nd change procedure end.



1. Loop L1 until ecx = 0.



1. Reset esi = 0, Reset ecx = 8.

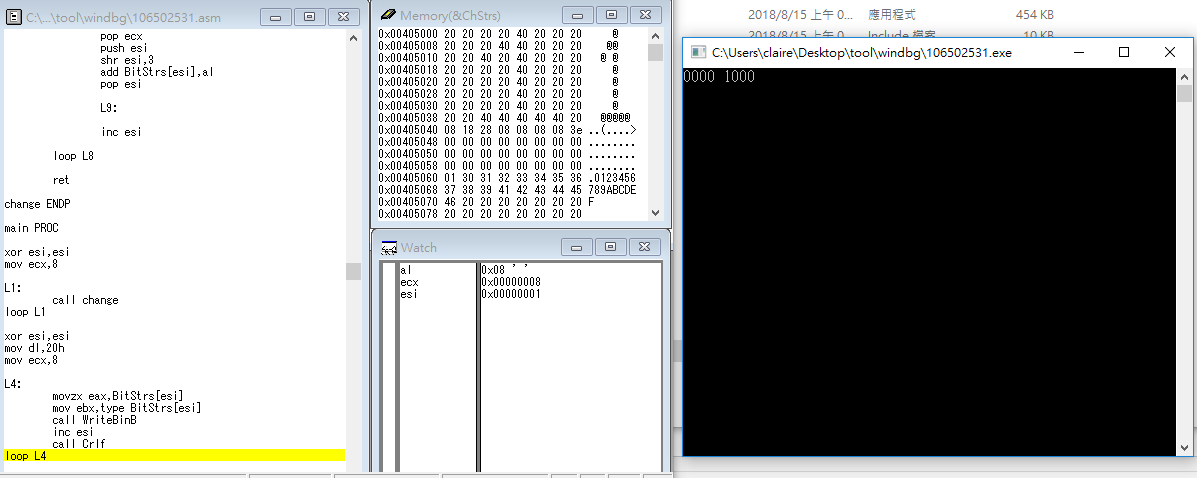


1. Call WriteBinB.

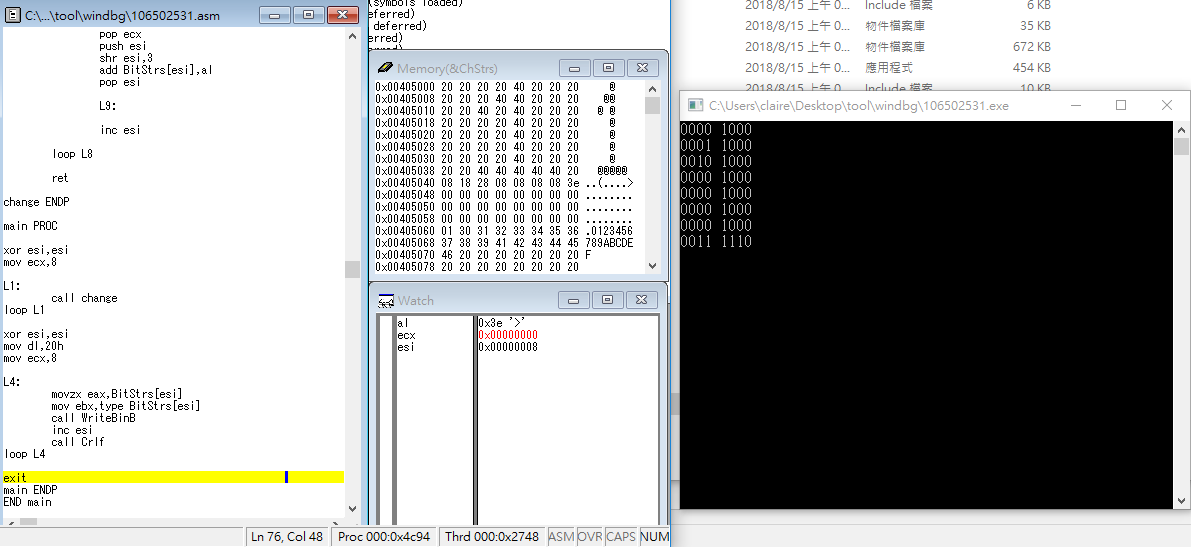
esi ++

Call Crlf.

Loop L4 until ecx = 0.



1. Result.



Review:

I considered that homework2 is quite hard for me, but I felt excited after finished it. In this homework, I used almost everything I had learned in class and lab. For example: using many Label to loop something and push/pop ecx, esi to use them repeatedly. Anyway, I thought I had made progress after I passed this challenge!