

Laboratory 03

MASM & LINK

CO 2103 Assembly Language

Objective

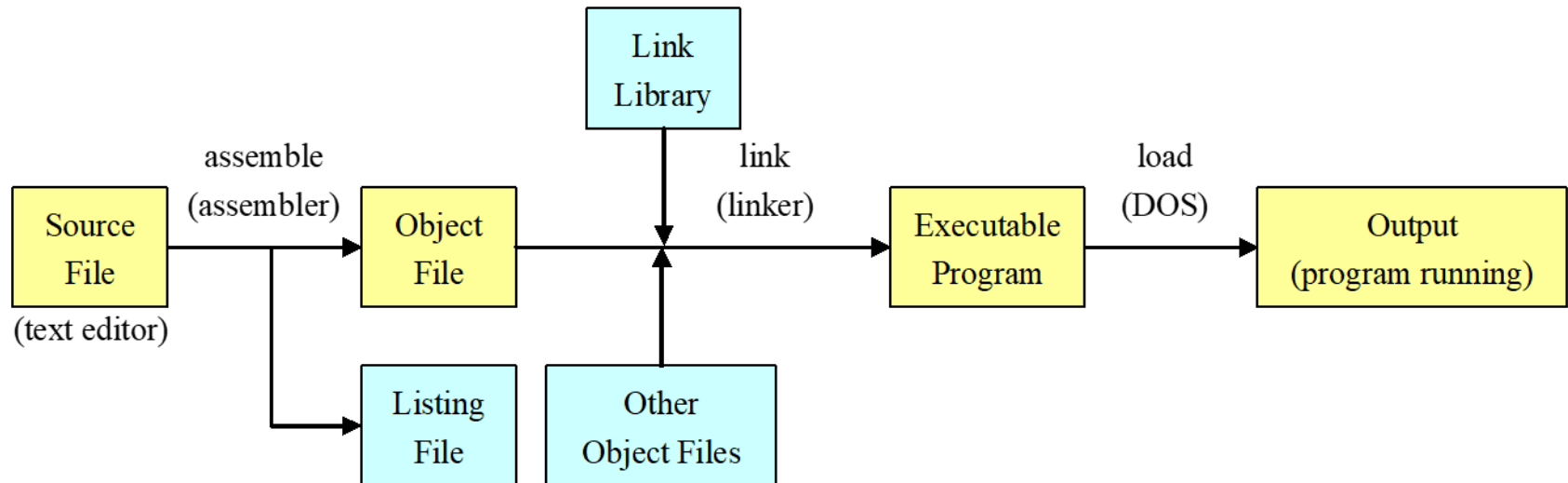
- Use MASM and LINK to develop AL program
 - DEBUG limitations
 - use MASM and LINK
 - simple screen output programs

Lab Instructions

- You may want to use a *pendrive* to store your files and work from there, as the PCs in the lab has *undo card* installed
 - you may want to create a directory to work in
- Download the *software* (*masmlink.zip*) from *myMoodle* and unpack the files (*masm.exe* and *link.exe*) into your work directory
- Download the *source files* (*lab3.zip*) into your work directory
- Work through **Tasks 1 to 5** – **attempt all sub-tasks given at the end of each source (.asm) files**
- While you work on the tasks, you may want to refer to *resources* (full instruction list, interrupt list) in *myMoodle*

AL Programming

- We recap the process of **AL** programming:



- In **Debug**, we perform the whole process (simplified) within the same environment
- **Task 1:** In comparison with any high level language (HLL) programming environment you have used, what are the shortcomings of **Debug**?

DEBUG - shortcomings

- No documentation – comments
- Can't save source file – so can't revise
- No listing file
- Limited editing features, e.g. copy and paste, no label, etc
- Only Hexadecimal – what about decimal numbers?
- Definitely not suitable for program of reasonable size
- Not convenient in the process of producing .exe
- What else?

Tools

- In this laboratory session, you will work with the tools mentioned in [Lecture 05](#):
 - **Editor**: any text editor, e.g. [EDIT](#), [NOTEPAD](#), [EMACS](#), [KEDIT](#)
 - **Assembler**: 80x86 assembler, e.g. [MASM](#), [AS](#)
 - **Linker**: e.g. [LINK](#), [LD](#)
- These tools enable us to systematically develop [AL](#) programs
- To begin with, you will use [EDIT](#) (available in [DOS](#)), [MASM](#) and [LINK](#) (get from [moodle](#))

Program Organization

- Before proceeding to use the tools, refer to [Lecture 05](#) and recap that an [AL](#) program comprises of following sections:
 - [.model](#)
 - [.stack](#)
 - [.code](#)
 - [.data](#) (can be either before or after the [code](#))
 - must end with the directive “[end](#)”

Skeleton in brief (recap)

`.model` small/medium/...

`.stack` size

`.data`

<data declaration>

;specify memory model

;specify stack size (start/bottom of stack)

;data segment

;declare data (variables, etc)

`.code`

`proc1 PROC` (near or far)

<statements>

`proc1 endp`

;code segment

;declare procedure *proc1*

;codes here – end with return instruction

;end of procedure *proc1*

`proc2 PROC` (near or far)

<statements>

`proc2 endp`

;declare procedure *proc2*

;codes here – end with return instruction

;end of procedure *proc2*

`main PROC`

<statements>

`main endp`

;begin of main program

;codes here

;end of main program

`end main`

;end of AL program

owh@ieee.org

CO 2103

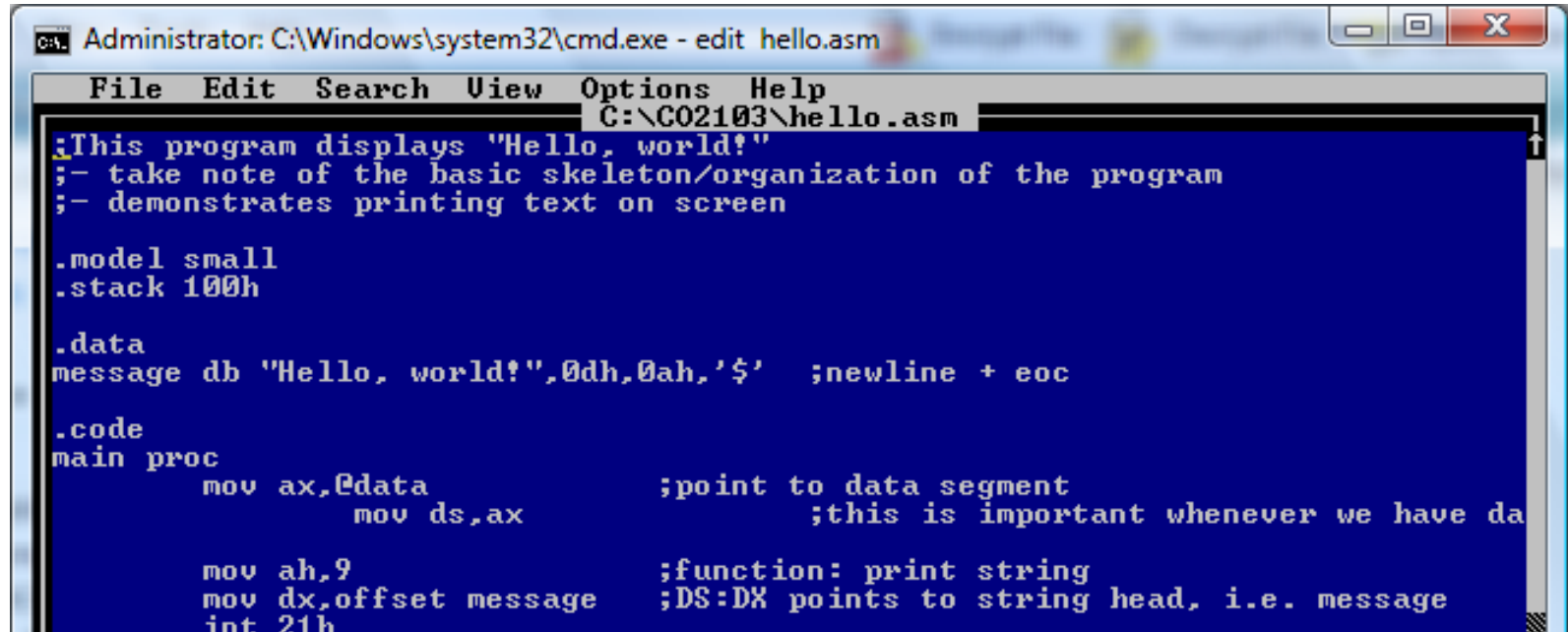
Exercises

For following tasks, use **MASM** and **LINK** to create the programs and use **DEBUG** to test them. Remember to complete the *tasks* at the end of each **.asm** file.

- Task 2:** Refer to following slides on use of **EDIT**, **MASM** and **LINK**. Open up **hello.asm** in **EDIT** and create its executable file. Solve the tasks given at the end of the file.
- Task 3:** Understand **hline1.asm** and solve the tasks given
- Task 4:** Understand **size1.asm** and solve the tasks given
- Task 5:** Write a program that uses a nested loop to **draw a square** of size **n x n** with asterisk ‘*’
 - Is it really square?
 - Perhaps it needs to get spaced out a bit? Solve this problem

EDIT

- Go to **command prompt** and type **EDIT**



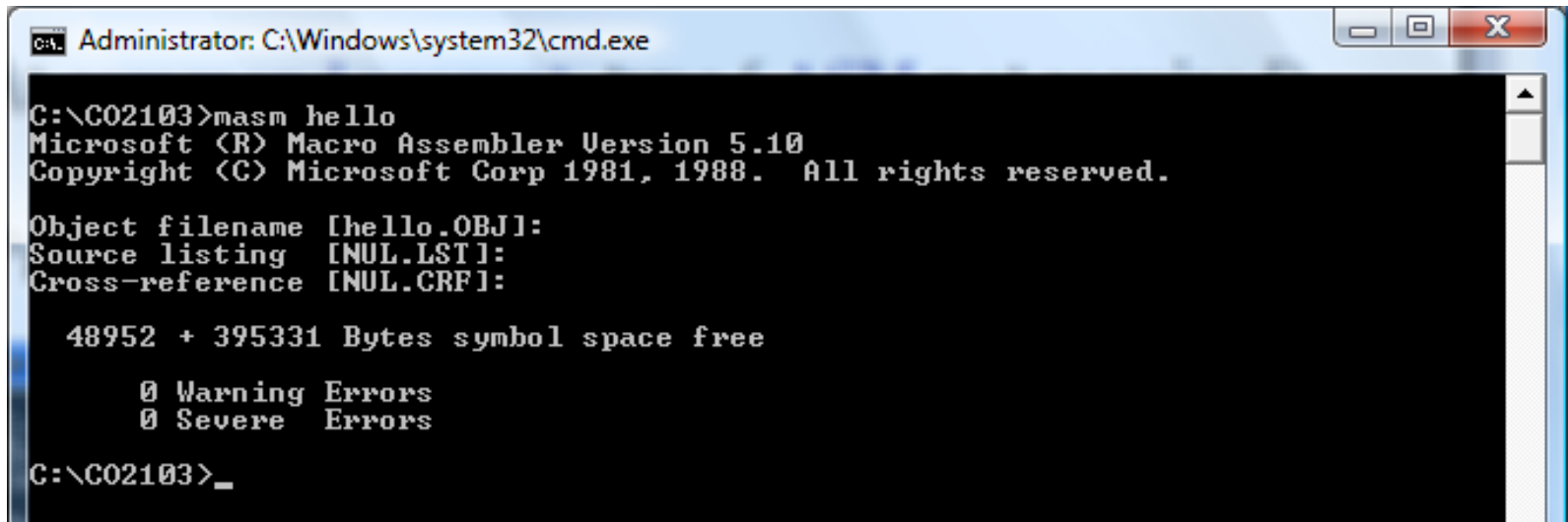
The screenshot shows a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe - edit hello.asm". The window displays the contents of the file "C:\C02103\hello.asm". The code is as follows:

```
;This program displays "Hello, world!"  
;- take note of the basic skeleton/organization of the program  
;- demonstrates printing text on screen  
  
.model small  
.stack 100h  
  
.data  
message db "Hello, world!",0dh,0ah,'$' ;newline + eoc  
  
.code  
main proc  
    mov ax,@data ;point to data segment  
        mov ds,ax ;this is important whenever we have da  
  
    mov ah,9 ;function: print string  
    mov dx,offset message ;DS:DX points to string head, i.e. message  
    int 21h
```

- use **ALT** key to invoke the menu
- save file with extension **ASM**, i.e. **filename.ASM**
- numbers are in decimal unless suffixed, i.e. 20, 20h, 20b

MASM

- At **command prompt**, type (**.ASM** not required)
MASM filename
- Press **enter** accepting default output destinations



```
Administrator: C:\Windows\system32\cmd.exe
C:\C02103>masm hello
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [hello.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

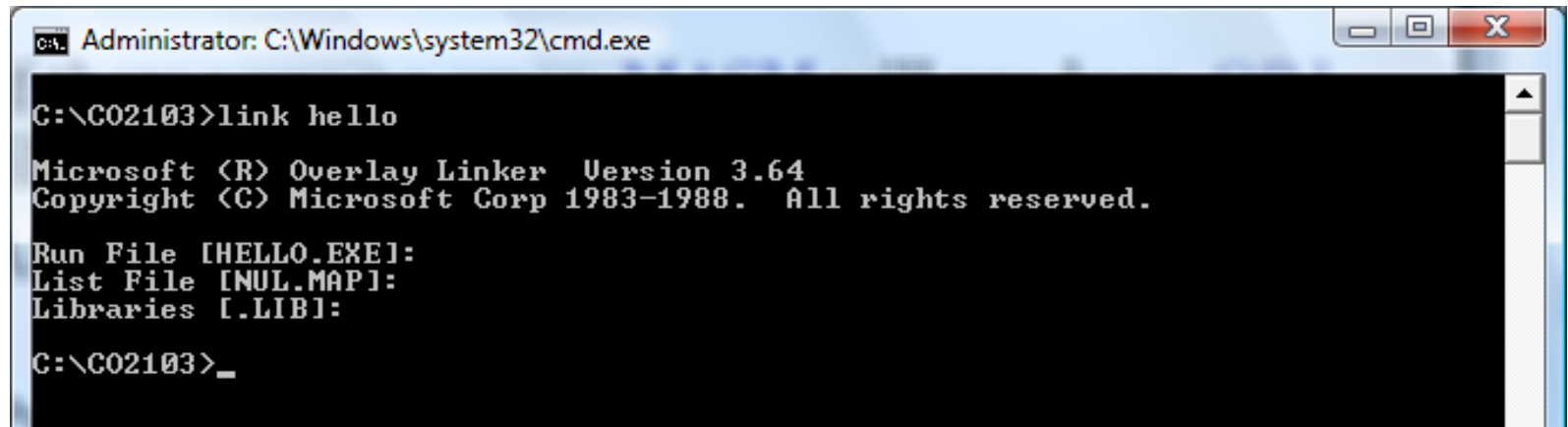
    48952 + 395331 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

C:\C02103>_
```

LINK

- If there were no error, **MASM** will produce **.OBJ** file that will be used by the **linker**
- At command prompt, type (**.OBJ** not required)
LINK filename
- Press **enter** accepting default output destinations



```
Administrator: C:\Windows\system32\cmd.exe
C:\C02103>link hello
Microsoft (R) Overlay Linker  Version 3.64
Copyright (C) Microsoft Corp 1983-1988.  All rights reserved.

Run File [HELLO.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

C:\C02103>_
```