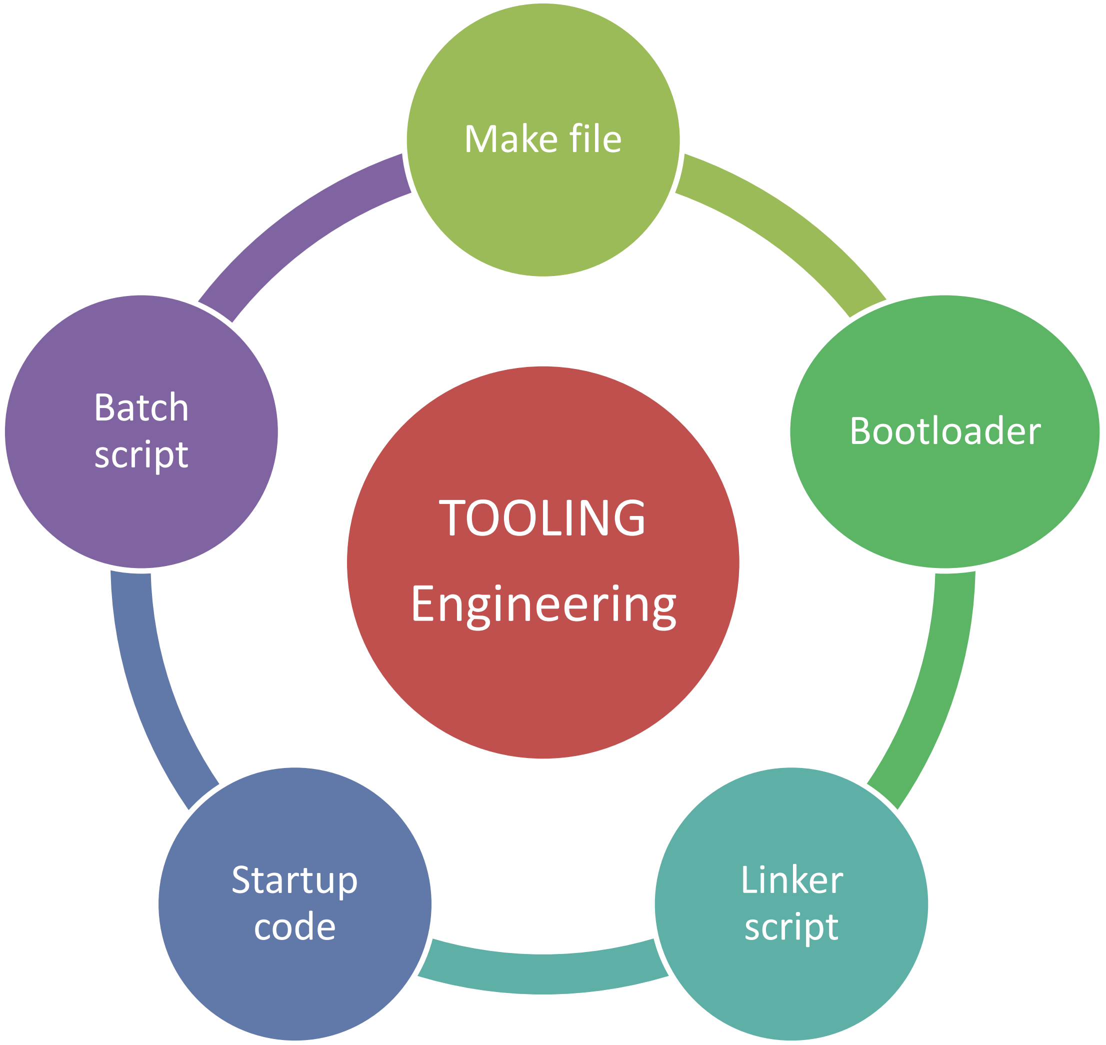
IMT SCHOOL

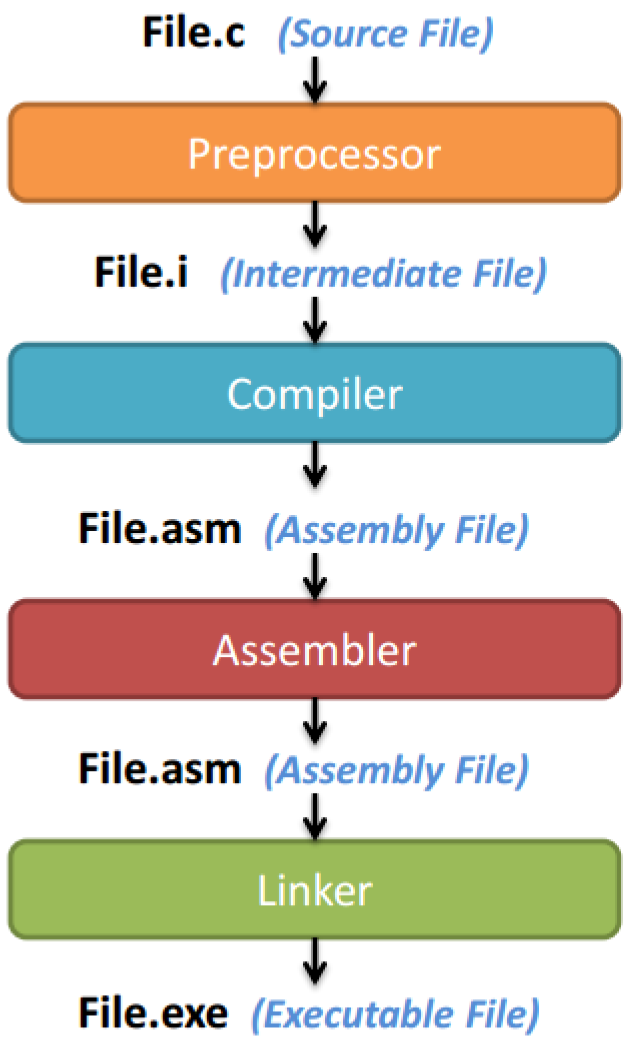
Embedded Systems

TOOLING

TOOLING ENGINEERING



Let's recap
C Building Process



Cmd

```
gcc   -E   file.c   -o  out.i
```

Cmd

```
gcc   -S   file.c   -o  out.asm
```

Cmd

```
gcc   -c   file.c   -o  out.o
```

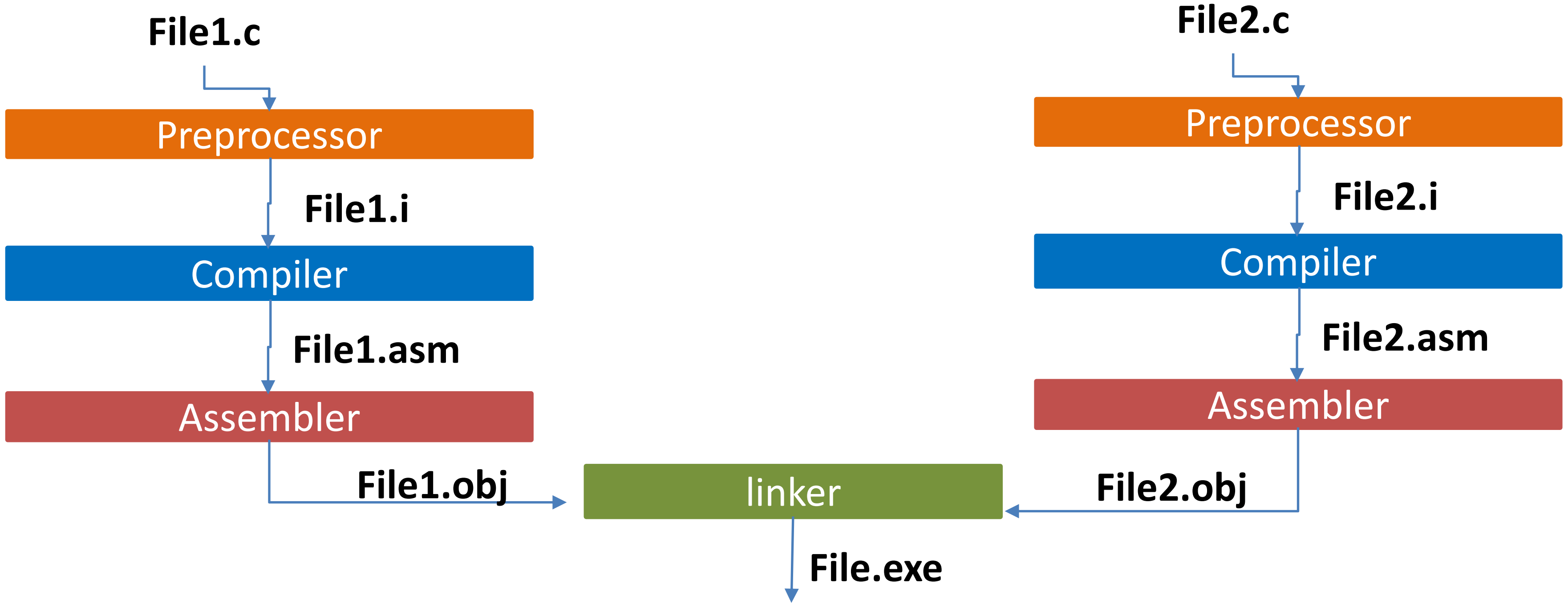
Cmd

```
gcc        file.c   -o  out.exe
```

Let's recap
C Building Process

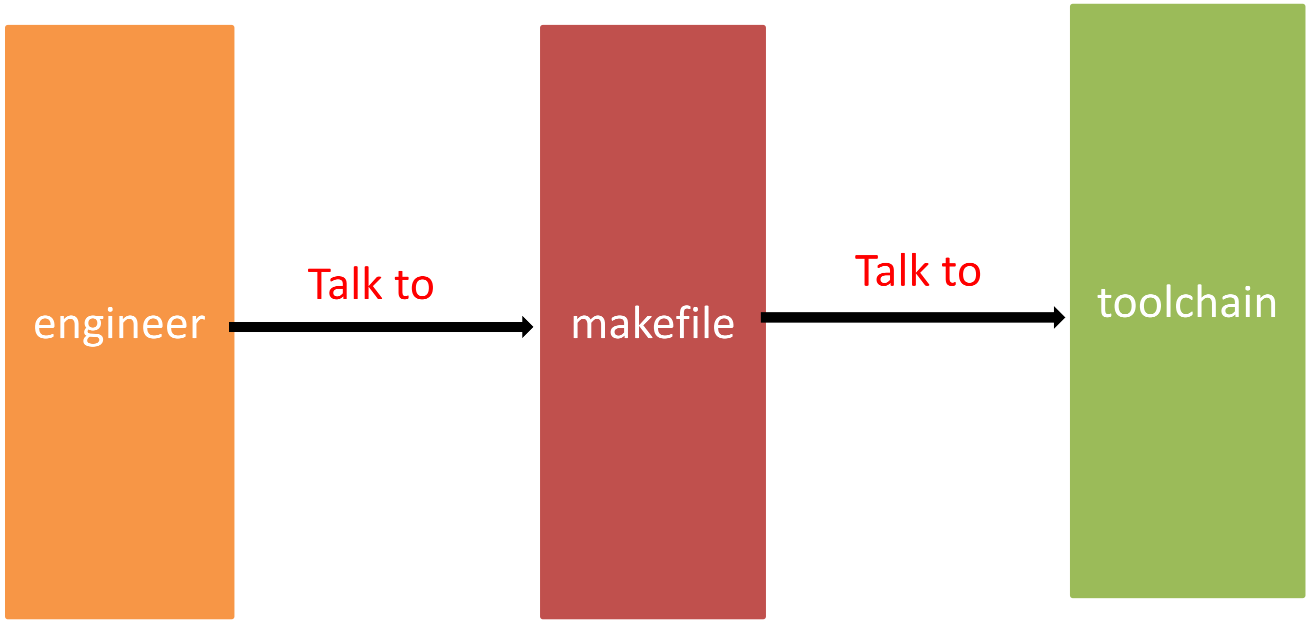
What 's the difference between object file and executable file?

Object file and Executable file are both written in binary (0&1) but the difference is clear when we have more than one file (file1.c,file2.c,…..) in this case every c file has its own object file but one executable file for the whole project.



Make file

Now if we want to debug our code we need to write each command by ourselves and this leads us to execute more effort so we can make use of a magic tool between the user and toolchain , at this case the user sends one command only to this magic tool which in turn sends the required commands to the toolchain. This magic tool is called the (makefile).



Make file

Rules:

1-(makefile) all letters are lower case and without spaces

2- the file has no extensions.

makefile

```
Rule name : dependencies
 tap            command
```

Rule name : This is the word that I want to write to generate makefile

dependencies : This is the used c file

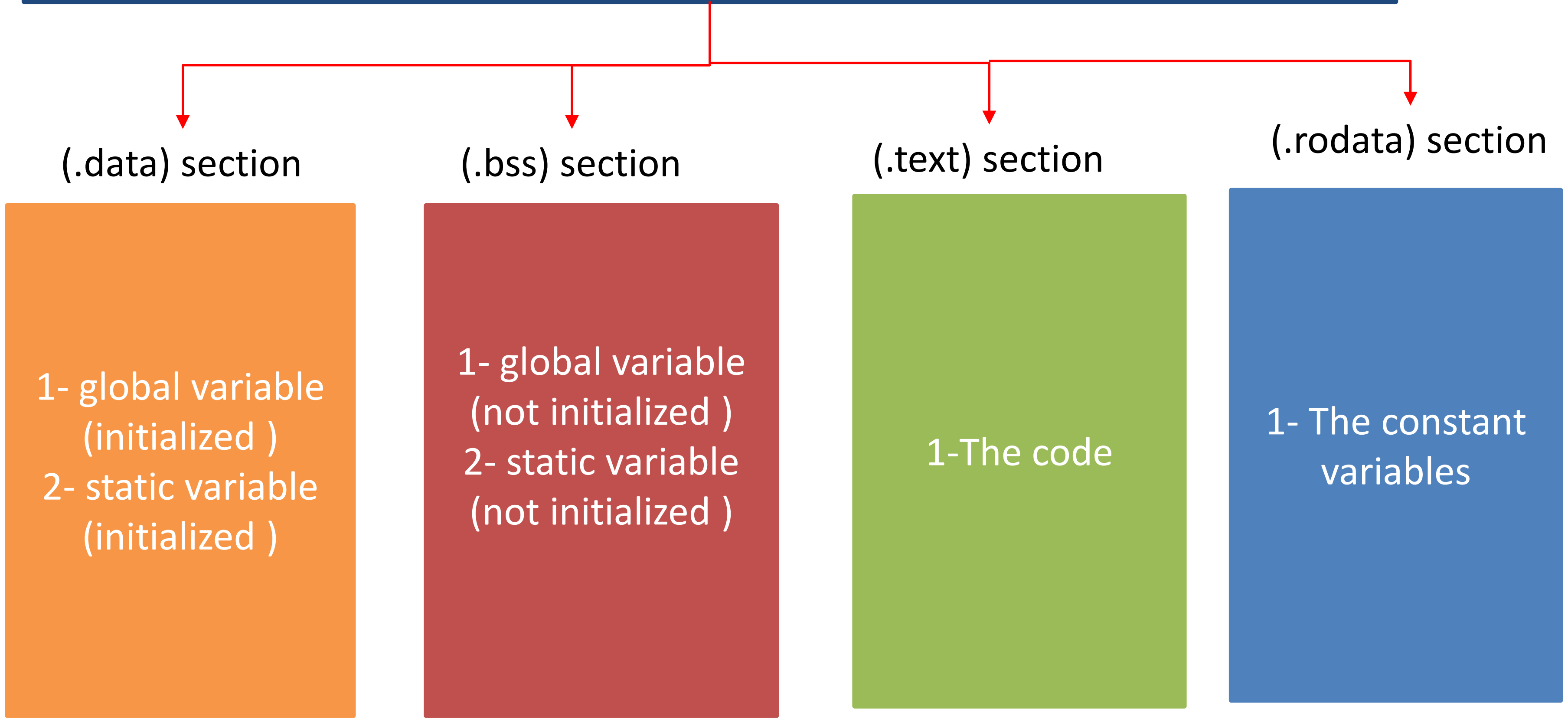
command: This is the command that you want to be generated.

Batch file

The batch file is similar to makefile but without sending any commands by this method you create (file.bat) and write all the same commands that you write in makefile and then you can double click on this file to execute all commands in this file.

Object file sections

The object file is divided into four sections every section has name, address and size.


(.data) section
1- global variable (initialized)
2- static variable (initialized)
(.bss) section
1- global variable (not initialized)
2- static variable (not initialized)
(.text) section
1-The code
(.rodata) section
1- The constant variables

Symbol table

This table is generated with each object file. It defines which of the global components(variables, functions,….) is needed or provided.

Symbol table

Func1	needed
Func2	provided
x	provided

Func1: the owner file has its prototype but it needs the implementation of the function which is located in another file

Func2: the owner file has the implementation of the function

Executable file section

The executable file is divided into five sections every section has name, address, size and type.

Type

Allocatable

This means that it allocates in flash memory and keeps its position there

loadable

This means that it will be loaded from flash to RAM

Startup code

- Startup code is located in the fifth section that is called (.init). This section is allocated in the first location in the flash. This location is the first code to be executed.

- Functions of startup code:

1-load (.data) section from flash to RAM

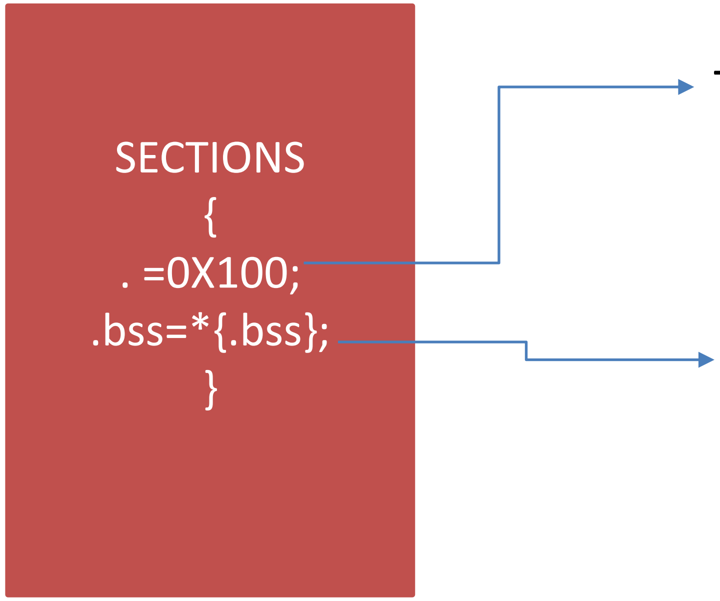
2-reserve (.bss) section into RAM and initialize it

3- initialize interrupt vector table

4-call entry point(main function)

Linker script

- The linker script is a script that the linker function is based on. The linker will give the memory sections its physical addresses based on what written in linker script, so it can be describes as the code of the linker tool.



```
SECTIONS
{
. =0X100;
.bss=*{.bss};
}
```

This means go to this address(0x100).

This means all .bss sections in all object files and assign it in the physical (.bss) in the executable file which is allocated at 0x100

bootloader

- The bootloader is a written code that can change from one communication protocol to another protocol to enable a microcontroller to be programmed with a protocol different from its default protocol.

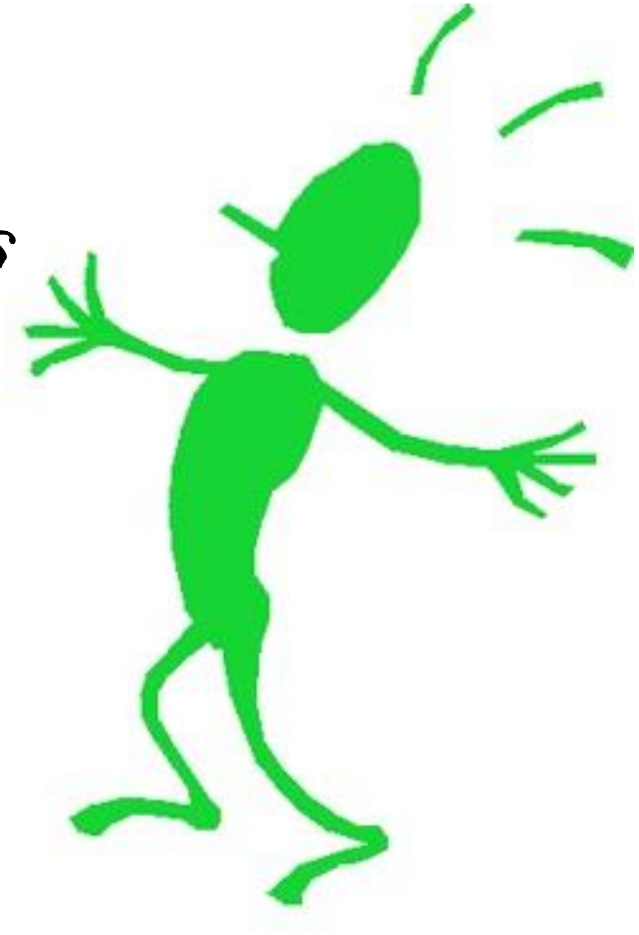
- i.e a controller that is programmed by SPI can now be programmed by UART using UART-SPI bootloader.

- After translating to the required protocol the flash memory of the target controller is updated with the code.

- In this case the processor is working during the program burning as if it programs itself so it is called self programming.

The End …

Any questions …?



Assignment

Expected Output

Assignment Description:

www.imtschool.com

ww.facebook.com/imaketechnologyschool/