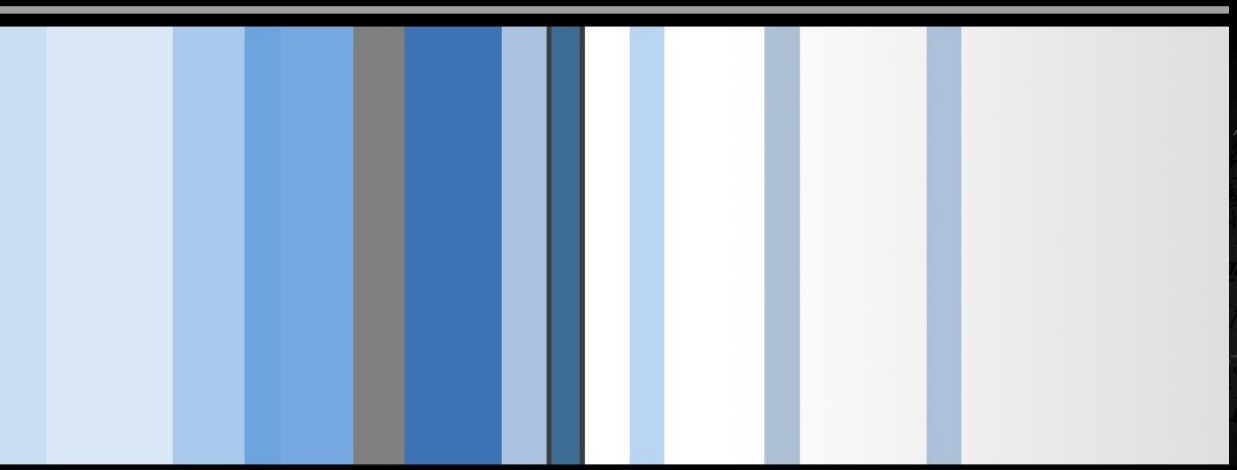
Class 12

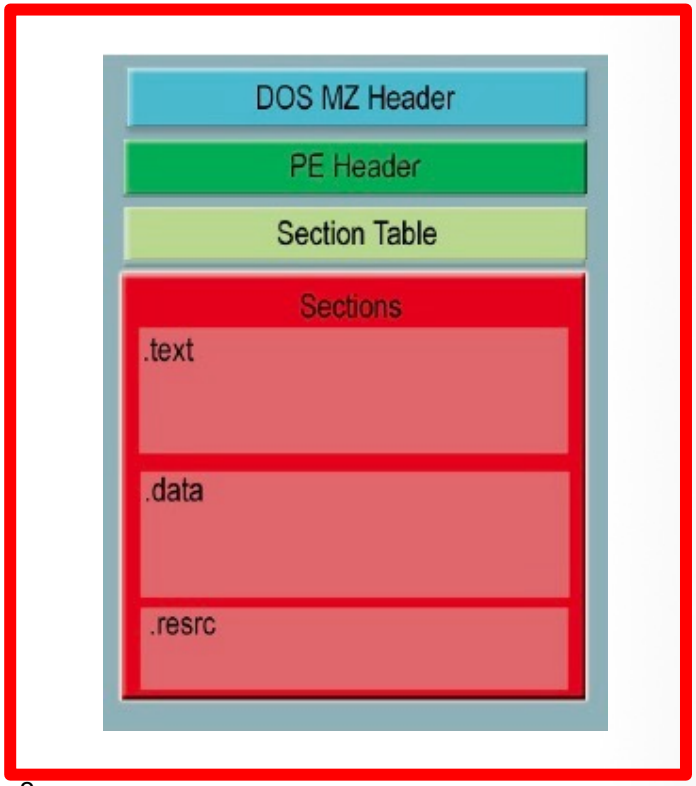
CSC 471 Modern Malware Analysis
PE Structure (1)

Si Chen (schen@wcupa.edu)



Portable Executable (PE) file

- A Portable Executable (**PE**) **file** is the standard binary **file** format for an **Executable (.exe) or DLL** under Windows NT, Windows 95, and Win32.
- Derived from COFF (Common Object File Format) in UNIX platform, and it is not really “portable”.



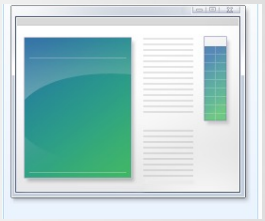
Now here is the kicker. Even though this specification is spelled out by Microsoft, compilers/linkers chose to ignore some parts of it.

To make things even worse, the Microsoft loader doesn't enforce a good portion of this specification and instead makes assumptions if things start getting weird.

So even though the spec outlined here says a particular field is supposed to hold a certain value, the compiler/linker or **even a malicious actor could put whatever they want in there and the program will likely still run**...

Portable Executable (PE) file

- PE formatted files include:
  - .exe, .scr (executable)
  - .dll, .ocx, .cpl, drv (library)
  - .sys, .vxd (driver files)
  - .obj (objective file)

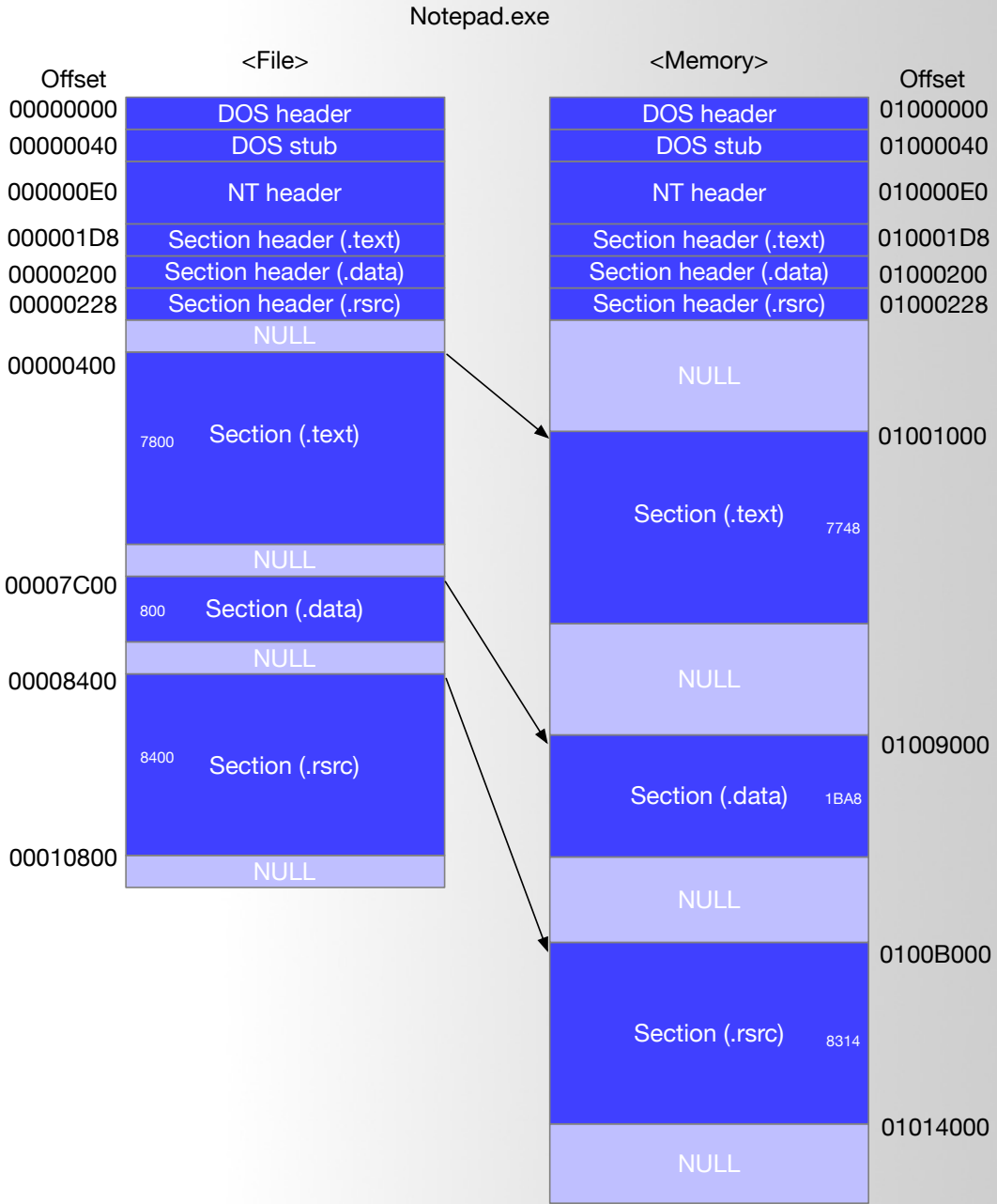


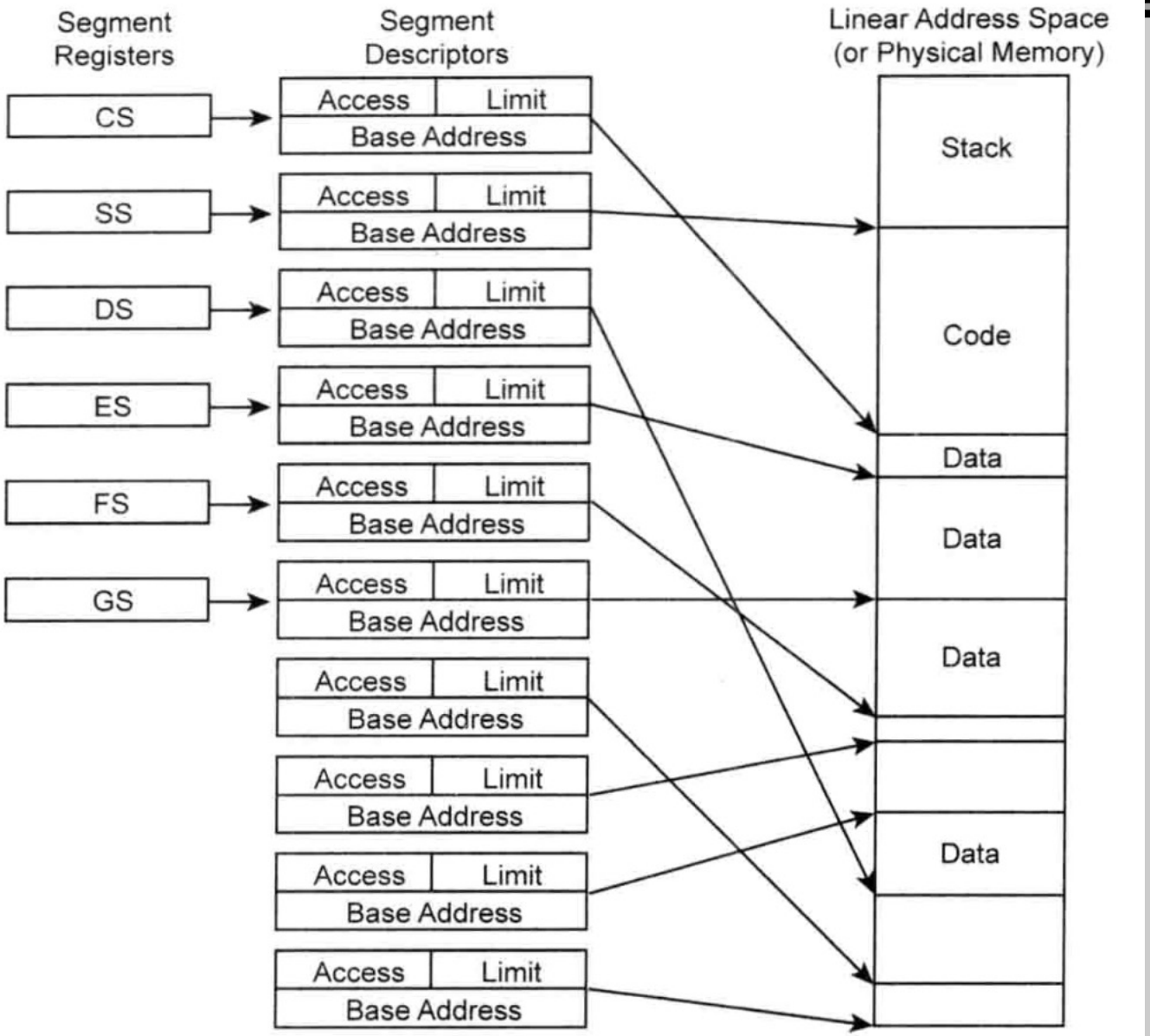
- All PE formatted files can be executed, except obj file.
  - .exe, .scr can be directly executed inside Shell (explorer.exe)
  - others can be executed by other program/service
- **PE refers to 32 bit** executable file, or **PE32**. **64 bit** executable file is named as **PE+ or PE32+.** (Note that it is not PE64).

PE Example – Notepad.exe

```
00000000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZÉ.........  ..
00000010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ╕.......@.......
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00  00 00 00 00 E8 00 00 00  ............Φ...
00000040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ..╢..┤.=!╕.L=!Th
00000050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is.program.canno
00000060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t.be.run.in.DOS.
00000070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
00000080  A5 6D 16 9B E1 0C 78 C8  E1 0C 78 C8 E1 0C 78 C8  Ñm.¢ß.x╚ß.x╚ß.x╚
00000090  1B 2F 38 C8 E0 0C 78 C8  E1 0C 78 C8 E0 0C 78 C8  ./8╚α.x╚ß.x╚α.x╚
000000A0  1B 2F 61 C8 F2 0C 78 C8  E1 0C 79 C8 23 0C 78 C8  ./a╚≥.x╚ß.y╚#.x╚
000000B0  76 2F 3D C8 E0 0C 78 C8  3B 2F 64 C8 F2 0C 78 C8  v/=╚α.x╚;/d╚≥.x╚
000000C0  1B 2F 45 C8 E0 0C 78 C8  52 69 63 68 E1 0C 78 C8  ./E╚α.x╚Richß.x╚
000000D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000000E0  00 00 00 00 00 00 00 00  50 45 00 00 4C 01 03 00  ........PE..L...
000000F0  0D 84 7D 3B 00 00 00 00  00 00 00 00 E0 00 0F 01  .ä};........α...
00000100  0B 01 07 00 00 6E 00 00  00 A6 00 00 00 00 00 00  .....n...ª......
00000110  E0 6A 00 00 00 10 00 00  00 80 00 00 00 00 00 01  αj.......Ç......
00000120  00 10 00 00 00 02 00 00  05 00 01 00 05 00 01 00  ................
00000130  04 00 00 00 00 00 00 00  00 30 01 00 00 04 00 00  .........0......
00000140  55 D8 01 00 02 00 00 80  00 00 04 00 00 10 01 00  U╪.....Ç........
00000150  00 00 10 00 00 10 00 00  00 00 00 00 10 00 00 00  ................
00000160  00 00 00 00 00 00 00 00  20 6D 00 00 C8 00 00 00  .........m..╚...
00000170  00 A0 00 00 48 89 00 00  00 00 00 00 00 00 00 00  .á..Hë..........
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
00000190  40 13 00 00 1C 00 00 00  00 00 00 00 00 00 00 00  @...............
```

Load PE file (Notepad.exe) into Memory




Segment Registers
Segment Descriptors
Linear Address Space (or Physical Memory)
CS
SS
DS
ES
FS
GS
Access
Limit
Base Address
Stack
Code
Data
Data
Data
Data

VA & RVA

- VA (Virtual Address): The address is called a “VA” because **Windows creates a distinct** VA space for each process, **independent** of physical memory. For almost all purposes, a VA should be considered just an address. A VA is not as predictable as an RVA because the loader might not load the image at its preferred location.

- RVA (Relative Virtual Address): The address of an item after it is loaded into memory, with the base address of the image file subtracted from it. The RVA of an item almost always differs from its position within the file on disk (file pointer).

RVA + ImageBase = VA

In 32bit Windows OS, each process has 4GB virtual memory which means the range of VA is: **00000000 - FFFFFFFF**

DOS Header

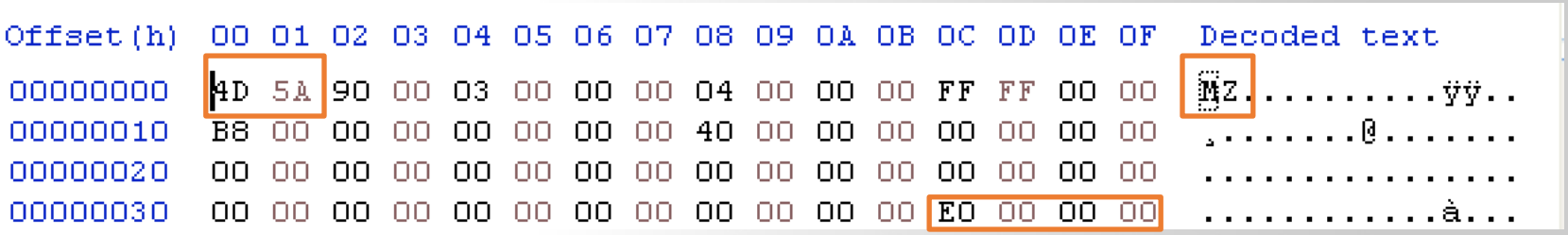
```
struct DOS_Header
{
// short is 2 bytes, long is 4 bytes
    char signature[2] = { 'M', 'Z' };
    short lastsize;
    short nblocks;
    short nreloc;
    short hdrsize;
    short minalloc;
    short maxalloc;
    void *ss; // 2 byte value
    void *sp; // 2 byte value
    short checksum;
    void *ip; // 2 byte value
    void *cs; // 2 byte value
    short relocpos;
    short noverlay;
    short reserved1[4];
    short oem_id;
    short oem_info;
    short reserved2[10];
    long  e_lfanew; // Offset to the 'PE\0\0' signature relative to the beginning of the file
}
```

The first 2 letters are **always** the letters "**MZ**", the initials of Mark Zbikowski, who created the first linker for DOS. To some people, the first few bytes in a file that determine the type of file are called the "**magic number**,"

DOS Header

```
long  e_lfanew;
```

long → 32 bit → ? Byte

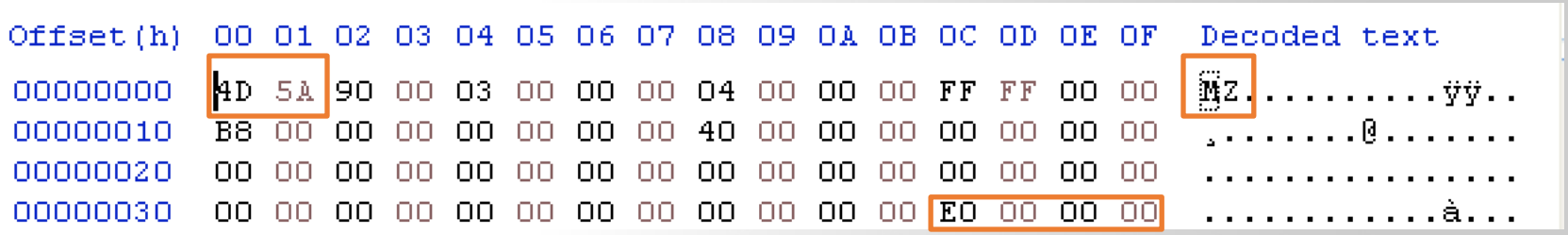


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E0	00	00	00	.....à...

E0 00 00 00

value for e_lfanew → ?

DOS Header



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	E0	00	00	00	.....à...

e_lfanew → 000000E0

DOS stub

```
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..°..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
00000070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00000080  EC 85 5B A1 A8 E4 35 F2 A8 E4 35 F2 A8 E4 35 F2  ì…[¡¨ä5ò¨ä5ò¨ä5ò
00000090  6B EB 3A F2 A9 E4 35 F2 6B EB 55 F2 A9 E4 35 F2  kë:ò©ä5òkëUò©ä5ò
000000A0  6B EB 68 F2 BB E4 35 F2 A8 E4 34 F2 63 E4 35 F2  këhò»ä5ò¨ä4òcä5ò
000000B0  6B EB 6B F2 A9 E4 35 F2 6B EB 6A F2 BF E4 35 F2  këkò©ä5òkëjò¿ä5ò
000000C0  6B EB 6F F2 A9 E4 35 F2 52 69 63 68 A8 E4 35 F2  këoò©ä5òRich¨ä5ò
000000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

https://virtualconsoles.com/online-emulators/dos/

```
C:\>notepad.exe
This program cannot be run in DOS mode.
```

Q & A