המרכז
לחינוך סייבר
CYBER EDUCATION CENTER

Assembly for Reverse Engineering

Registers & Flags

Barak Gonen

```
call    2084
jmp     15CA
call    17B4
mov     si,7160
xor     di,di
mov     es,[7606]
mov     bx,000F
xor     cx,cx
mov     bp,0001
mov     dx,ds
```

Registers

- The CPU has some special hardware circuits
- Using them requires zero wait time
- Limited resource
- General purpose registers:
  - Used for all types of calculations
- Special purpose registers
  - IP
  - FLAGS

General Purpose Registers

- 16 bit assembly:

  AX – Accumulator register

  BX – Base address register

  CX – Count register

  DX – Data register

  SI – Source Index

  DI – Destination Index

  BP – Base Pointer

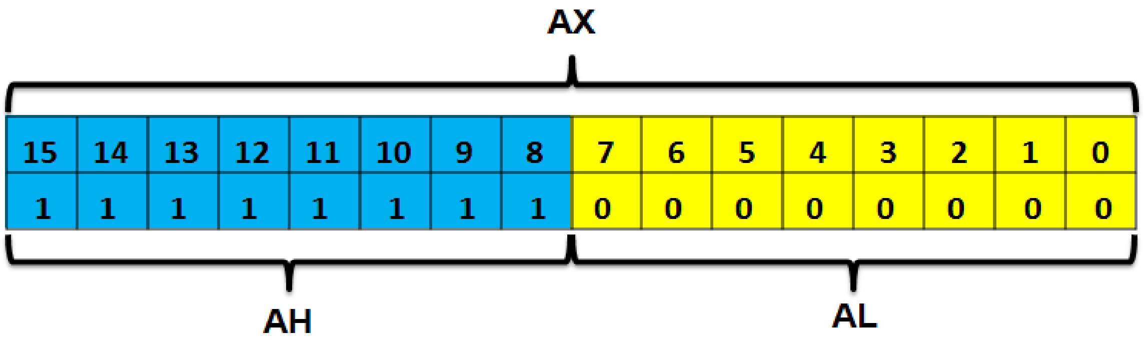
  SP – Stack Pointer

8 Bit Registers

- For byte size operations, use byte registers:

16 bit	8 bit	8 bit
AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

- mov ax, 0xFF00
  - mov ah, 0xFF
  - mov al, 0x00



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

32 Bit Registers

- “E” stands for “Extended”

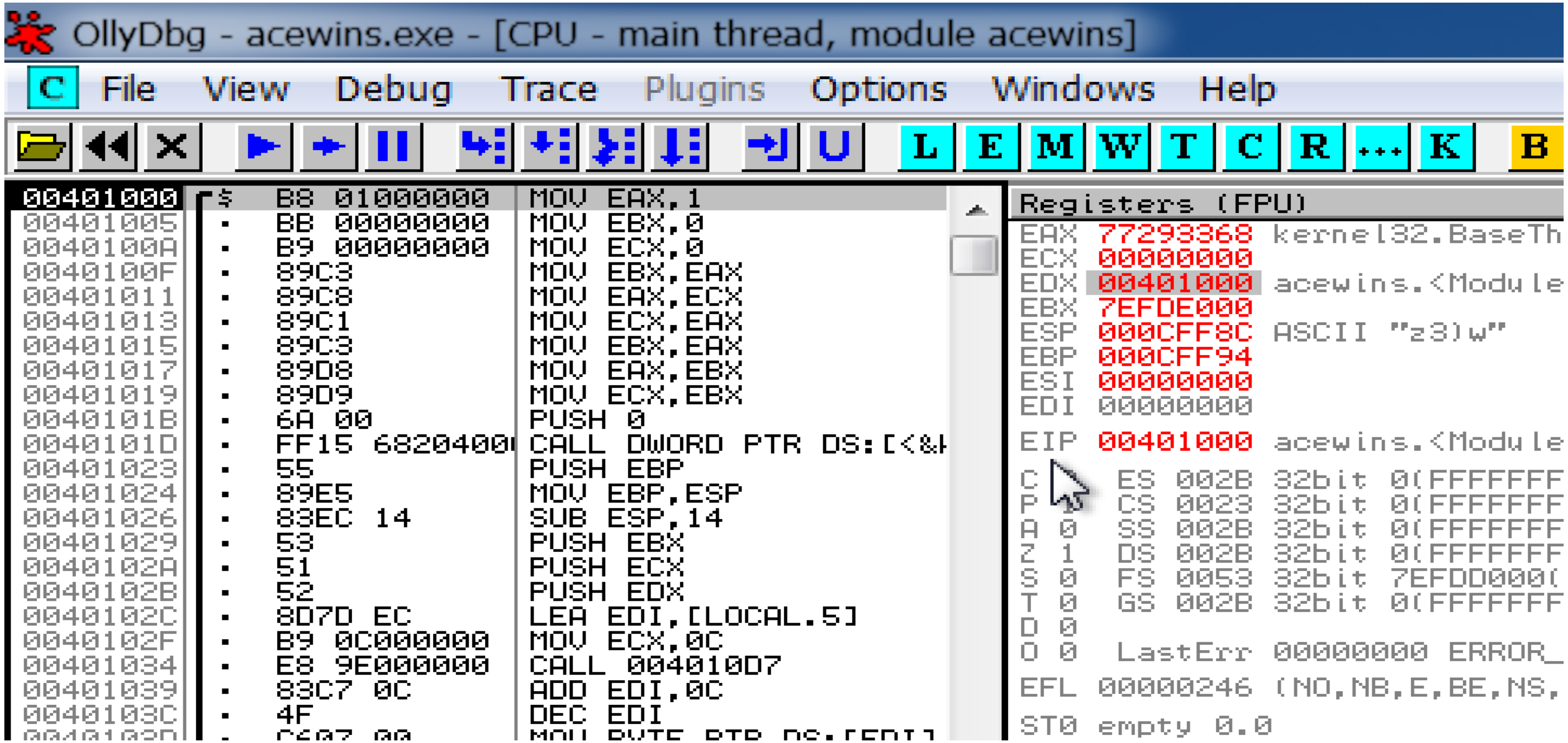
32 Bit	16 Bit	8 Bit
EAX	AX	AL / AH
EBX	BX	BL / BH
ECX	CX	CL / CH
EDX	DX	DL / DH
ESI	SI	
EDI	DI	
EBP	BP	
ESP	SP	

Other General Purpose Regs.

- ESI / EDI – used for copying data from buffer to another
  - String operations
  - Will show up in some exercises
- EBP / ESP
  - Much of the course will be dedicated (stack, procedures)

EIP -Special Purpose Register

- Study hands on :-)


OllyDbg - acewins.exe - [CPU - main thread, module acewins]
C File View Debug Trace Plugins Options Windows Help
U L E M W T C R ... K B

```
00401000 ┌$  B8 01000000   MOV EAX,1
00401005 │.  BB 00000000   MOV EBX,0
0040100A │.  B9 00000000   MOV ECX,0
0040100F │.  89C3          MOV EBX,EAX
00401011 │.  89C8          MOV EAX,ECX
00401013 │.  89C1          MOV ECX,EAX
00401015 │.  89C3          MOV EBX,EAX
00401017 │.  89D8          MOV EAX,EBX
00401019 │.  89D9          MOV ECX,EBX
0040101B │.  6A 00         PUSH 0
0040101D │.  FF15 6820400  CALL DWORD PTR DS:[<&
00401023 │.  55            PUSH EBP
00401024 │.  89E5          MOV EBP,ESP
00401026 │.  83EC 14       SUB ESP,14
00401029 │.  53            PUSH EBX
0040102A │.  51            PUSH ECX
0040102B │.  52            PUSH EDX
0040102C │.  8D7D EC       LEA EDI,[LOCAL.5]
0040102F │.  B9 0C000000   MOV ECX,0C
00401034 │.  E8 9E000000   CALL 004010D7
00401039 │.  83C7 0C       ADD EDI,0C
0040103C │.  4F            DEC EDI
```

```
Registers (FPU)
EAX 77293368 kernel32.BaseTh
ECX 00000000
EDX 00401000 acewins.<Module
EBX 7EFDE000
ESP 000CFF8C ASCII "z3)w"
EBP 000CFF94
ESI 00000000
EDI 00000000

EIP 00401000 acewins.<Module

C      ES 002B 32bit 0(FFFFFFF
P 1    CS 0023 32bit 0(FFFFFFF
A 0    SS 002B 32bit 0(FFFFFFF
Z 1    DS 002B 32bit 0(FFFFFFF
S 0    FS 0053 32bit 7EFDD000(
T 0    GS 002B 32bit 0(FFFFFFF
D 0
O 0    LastErr 00000000 ERROR_

EFL 00000246 (NO,NB,E,BE,NS,

ST0 empty 0.0
```

Summary

- We have learned about:
  - CPU’s general registers – important for reading code
  - EIP - important for tracing code
- By-products:
  - Practiced number representation
  - Practiced Ollydbg

QUESTIONS