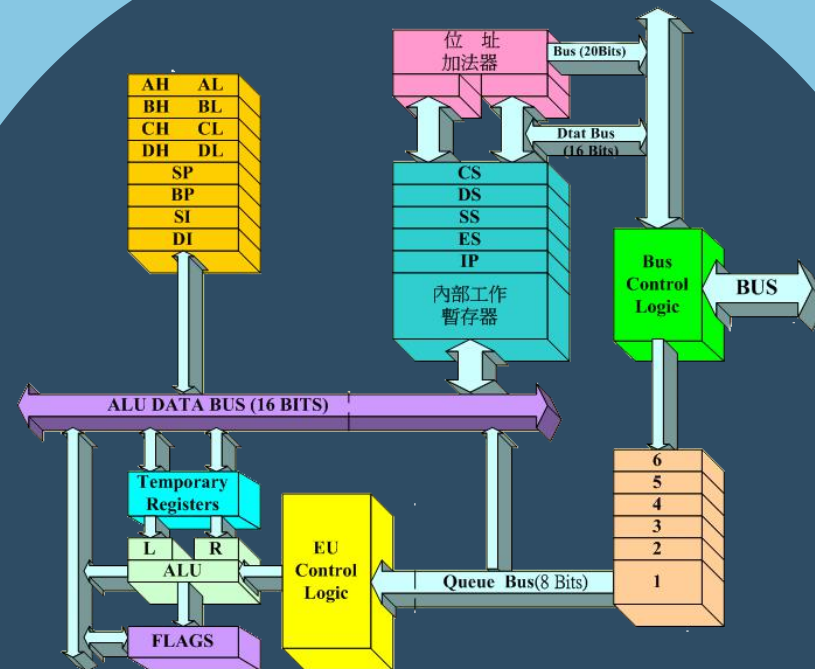


高级语言的指令级调试

贺利坚 主讲



汇编语言程序设计
Assembly Language

汇编语言与机器语言、高级语言的关系

汇编语言与机器语言

- 📁 一对一关系，每一条汇编语言指令对应一条机器语言指令

```
C:\>debug ptest.exe
-u
076A:0000 B86D07      MOV     AX,076D
076A:0003 8ED8      MOV     DS,AX
076A:0005 B80000      MOV     AX,0000
076A:0008 BD360000     LEA     SI,[0000]
076A:000C B90500      MOV     CX,0005
```

高级语言与汇编语言

- 📁 一对多关系，C语言的一条语句会扩展为多条汇编语言指令或机器指令

C:

```
int Y;
int X = (Y+4)*3;
```

Assembly:

```
mov eax, Y
add eax, 4
mov ebx, 3
imul ebx
mov X, eax
```

VS中的反汇编

```
反汇编 c001.cpp
(全局范围)

#include <stdio.h>
int main()
{
    int Y=9;
    int X = (Y+4)*3;
    return 0;
}
```

```
反汇编 c001.cpp
地址: main(void)

#include <stdio.h>
int main()
{
    00321370  push    ebp
    00321371  mov     ebp, esp
    00321373  sub     esp, 0D8h
    00321379  push    ebx
    0032137A  push    esi
    0032137B  push    edi
    0032137C  lea     edi, [ebp-0D8h]
    00321382  mov     ecx, 36h
    00321387  mov     eax, 0CCCCCCCCh
    0032138C  rep stos dword ptr es:[edi]

    int Y=9;
    0032138E  mov     dword ptr [Y], 9

    int X = (Y+4)*3;
    00321395  mov     eax, dword ptr [Y]
    00321398  add     eax, 4
    0032139B  imul    eax, eax, 3
    0032139E  mov     dword ptr [X], eax

    return 0;
    003213A1  xor     eax, eax
}
```

反汇编再例

```
反汇编 c001.cpp
(全局范围)
#include <stdio.h>
int main()
{
    int X =3, Y;
    Y=(X++) +1;
    Y=(++X) +1;
    return 0;
}
```

```
Y=(X++) +1;
001217F5  mov     eax, dword ptr [X]
001217F8  add     eax, 1
001217FB  mov     dword ptr [Y], eax
001217FE  mov     ecx, dword ptr [X]
00121801  add     ecx, 1
00121804  mov     dword ptr [X], ecx
Y=(++X) +1;
00121807  mov     eax, dword ptr [X]
0012180A  add     eax, 1
0012180D  mov     dword ptr [X], eax
00121810  mov     ecx, dword ptr [X]
00121813  add     ecx, 1
00121816  mov     dword ptr [Y], ecx
return 0;
00121819  xor     eax, eax
}
```

在Codeblocks中观察、调试

Disassembly

Function: main (D:\CB\C\main.c:5)
Frame start: 0x28ff20

0x401334	push	%ebp
0x401335	mov	%esp,%ebp
0x401337	and	\$0xffffffff0,%esp
0x40133a	sub	\$0x10,%esp
0x40133d	call	0x401920 <__main>
0x401342	movl	\$0x9,0xc(%esp)
0x40134a	mov	0xc(%esp),%edx
0x40134e	mov	%edx,%eax
0x401350	shl	%eax
0x401352	add	%edx,%eax
0x401354	add	\$0xc,%eax
0x401357	mov	%eax,0x8(%esp)
0x40135b	mov	\$0x0,%eax
0x401360	leave	
0x401361	ret	

main.c

```
1  #include <stdio.h>
2  int main()
3  {
4      int Y=9;
5      int X = (Y+4)*3;
6      return 0;
7  }
```

Memor