0929_综合研究5研究报告

> 用 debug 对一下程序进行分析，记录每一条c语句运行后，相关内存单元的值

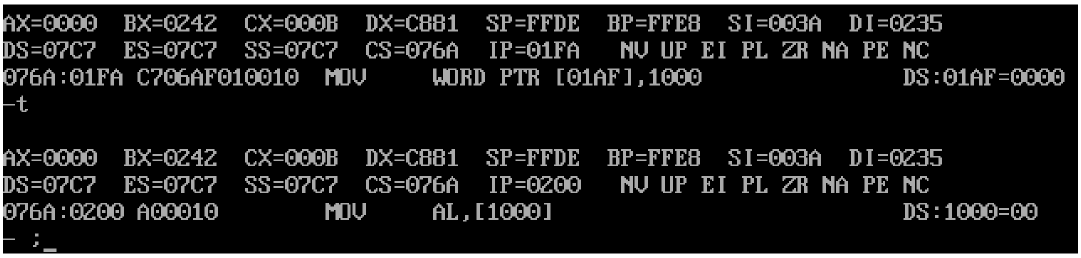
a.c

> 注意理解指针的机制，“**” 和 “&” 运算的意义

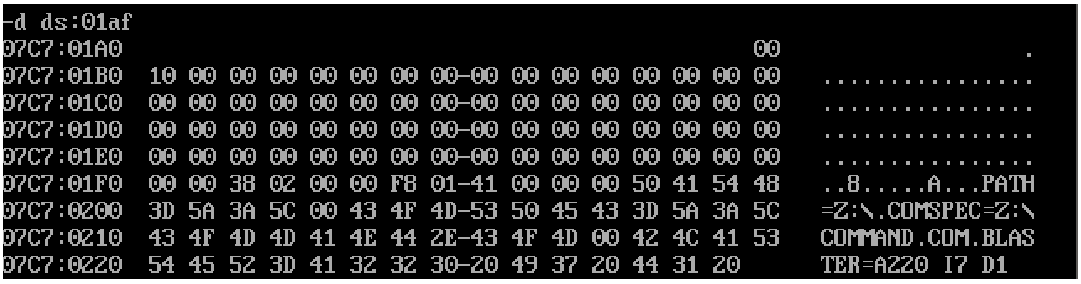
```c
char ch;
char *p;
char **pa;
char far *pf;
int n;

main() {
    p = (unsigned char *)0x1000;
    ch = *(unsigned char *)0x1000 + *p + *(unsigned char far *)0x200;

    p = &ch;

    *p = *p + 1;

    pa = &p;
    **pa = **pa + 1;

    pf = (char far *)&ch;
    *pf = *pf + 1;

    n = (int)&ch;
    *(char *)n = *(char *)n + 1;
}
```

- 第一句 `p = (unsigned char *)0x1000;` p在数据段中偏移地址为 `01af` 然后看内存中的值

  ```c
          /*mov    word ptr DGROUP:_p,4096*/
          p = (unsigned char *)0x1000;
  ```

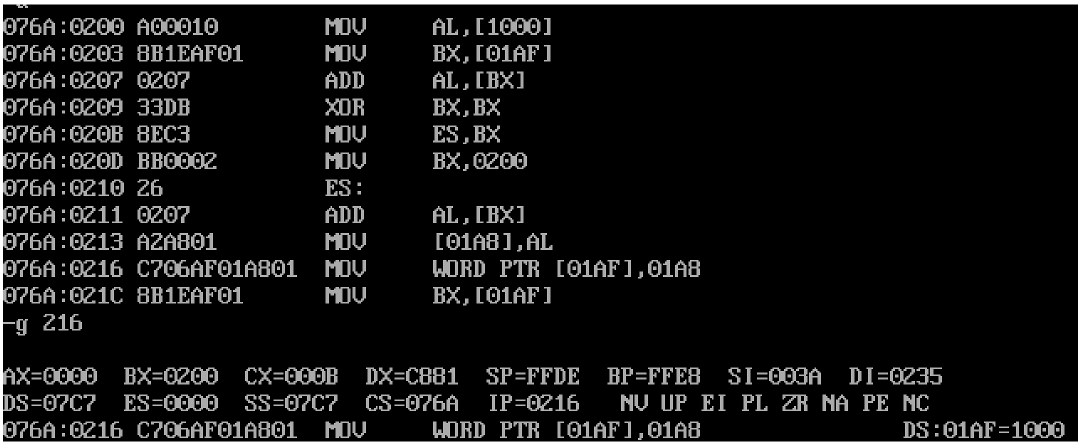
  ```
  AX=0000  BX=0242  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
  DS=07C7  ES=07C7  SS=07C7  CS=076A  IP=01FA   NV UP EI PL ZR NA PE NC
  076A:01FA C706AF010010  MOV     WORD PTR [01AF],1000               DS:01AF=0000
  -t

  AX=0000  BX=0242  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
  DS=07C7  ES=07C7  SS=07C7  CS=076A  IP=0200   NV UP EI PL ZR NA PE NC
  076A:0200 A00010        MOV     AL,[1000]                          DS:1000=00
  - ;_
  ```



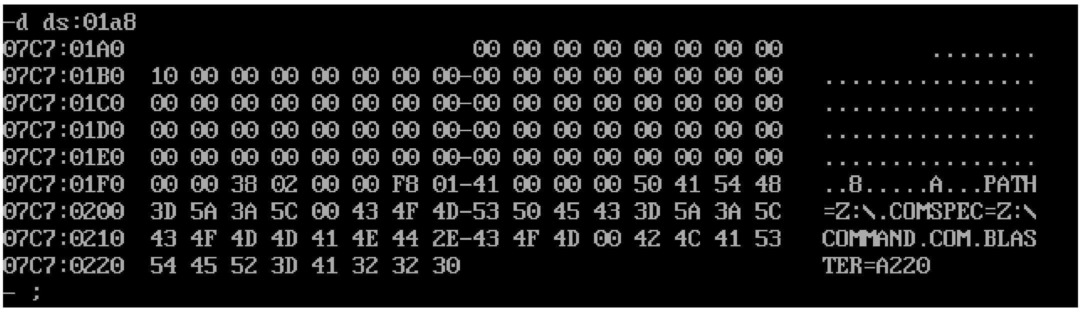
```
-d ds:01af
07C7:01A0                                                00    .
07C7:01B0  10 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30-20 49 37 20 44 31 20      TER=A220 I7 D1
```

- 第二句 `ch = *(unsigned char *)0x1000 + *p + *(unsigned char far *)0x200;`

```
    /*
        mov al,byte ptr [4096]
        mov bx,word ptr DGROUP:_p
        add al,byte ptr [bx]
        xor bx,bx
        mov es,bx
        mov bx,512
        add al,byte ptr es:[bx]
        mov byte ptr DGROUP:_ch,al
    */
    ch = *(unsigned char *)0x1000 + *p + *(unsigned char far *)0x200;
```



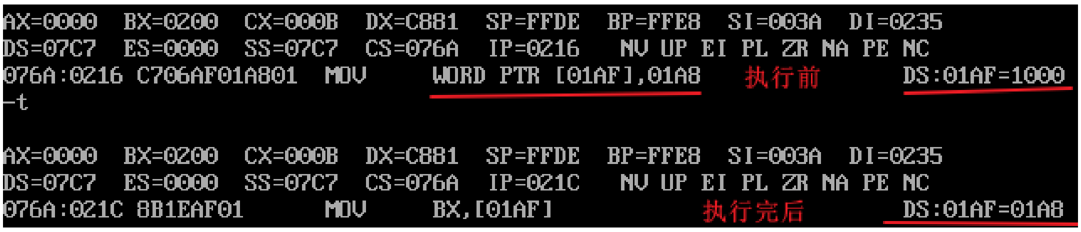
```
076A:0200 A00010        MOV     AL,[1000]
076A:0203 8B1EAF01      MOV     BX,[01AF]
076A:0207 0207          ADD     AL,[BX]
076A:0209 33DB          XOR     BX,BX
076A:020B 8EC3          MOV     ES,BX
076A:020D BB0002        MOV     BX,0200
076A:0210 26            ES:
076A:0211 0207          ADD     AL,[BX]
076A:0213 A2A801        MOV     [01A8],AL
076A:0216 C706AF01A801  MOV     WORD PTR [01AF],01A8
076A:021C 8B1EAF01      MOV     BX,[01AF]
-g 216

AX=0000  BX=0200  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=0216   NV UP EI PL ZR NA PE NC
076A:0216 C706AF01A801  MOV     WORD PTR [01AF],01A8               DS:01AF=1000
```



```
-d ds:01a8
07C7:01A0                          00 00 00 00 00 00 00 00           ........
07C7:01B0  10 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30                           TER=A220
- ;
```

- 第三句 `p = &ch;`

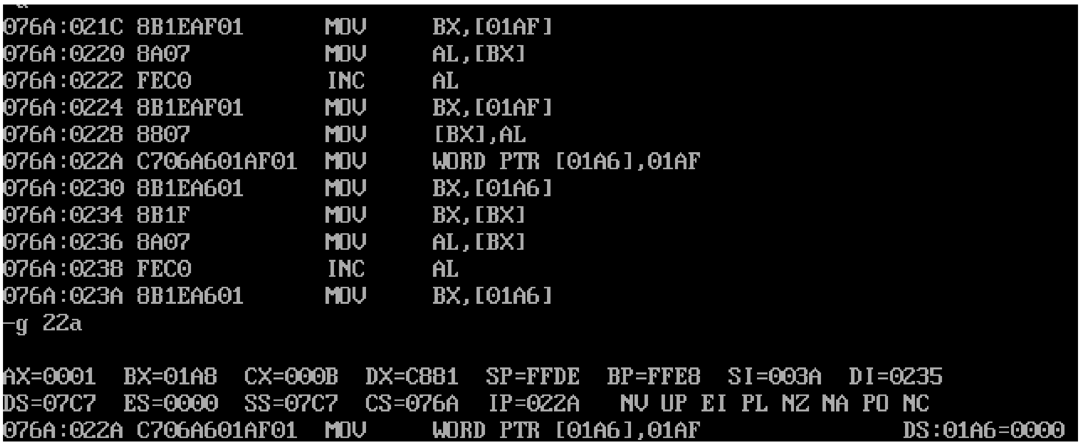
```
    /*                       [01af]             01a8*/
    /*mov   word ptr DGROUP:_p,offset DGROUP:_ch*/
    p = &ch;
```



```
AX=0000  BX=0200  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=0216   NV UP EI PL ZR NA PE NC
076A:0216 C706AF01A801  MOV     WORD PTR [01AF],01A8  执行前       DS:01AF=1000
-t

AX=0000  BX=0200  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=021C   NV UP EI PL ZR NA PE NC
076A:021C 8B1EAF01      MOV     BX,[01AF]          执行完后        DS:01AF=01A8
```

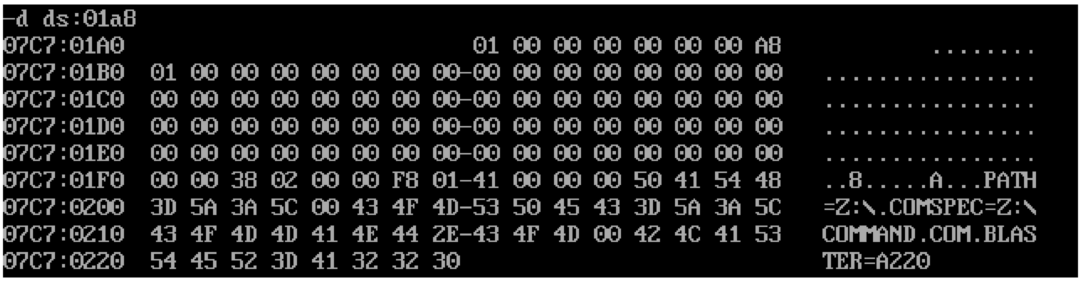
- 第四句 `*p = *p + 1;`

```
/*
    mov bx,word ptr DGROUP:_p
    mov al,byte ptr [bx]
    inc al
    mov bx,word ptr DGROUP:_p
    mov byte ptr [bx],al
*/
*p = *p + 1;
```



```
076A:021C 8B1EAF01      MOV     BX,[01AF]
076A:0220 8A07          MOV     AL,[BX]
076A:0222 FEC0          INC     AL
076A:0224 8B1EAF01      MOV     BX,[01AF]
076A:0228 8807          MOV     [BX],AL
076A:022A C706A601AF01  MOV     WORD PTR [01A6],01AF
076A:0230 8B1EA601      MOV     BX,[01A6]
076A:0234 8B1F          MOV     BX,[BX]
076A:0236 8A07          MOV     AL,[BX]
076A:0238 FEC0          INC     AL
076A:023A 8B1EA601      MOV     BX,[01A6]
-g 22a

AX=0001  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=022A   NV UP EI PL NZ NA PO NC
076A:022A C706A601AF01  MOV     WORD PTR [01A6],01AF               DS:01A6=0000
```

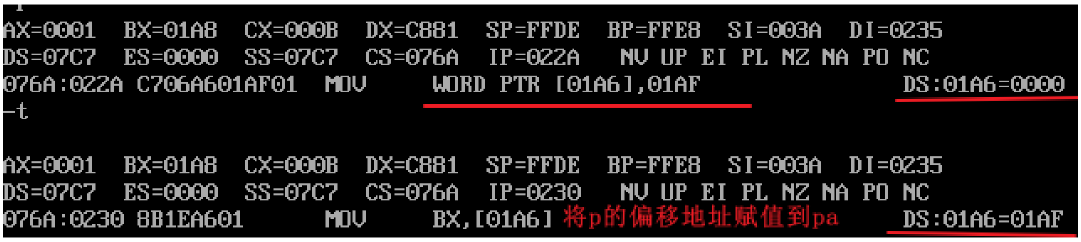
可以看到p指向的内存中的值增加一



```
-d ds:01a8
07C7:01A0                           01 00 00 00 00 00 00 A8           ........
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30                           TER=A220
```

- 第五句 `pa = &p;`

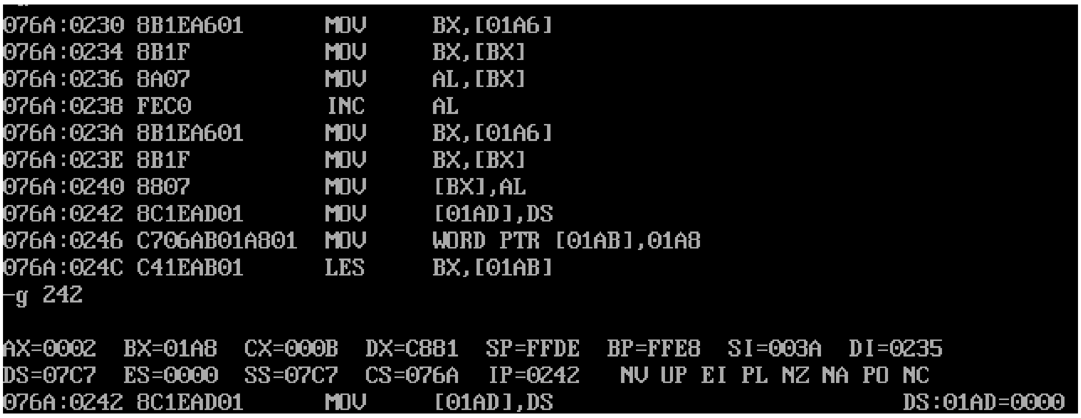
```
/*              ds:[01a6]        01af
mov word ptr DGROUP:_pa,offset DGROUP:_p
*/
pa = &p;
```



```
AX=0001  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=022A   NV UP EI PL NZ NA PO NC
076A:022A C706A601AF01  MOV     WORD PTR [01A6],01AF               DS:01A6=0000
-t

AX=0001  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=0230   NV UP EI PL NZ NA PO NC
076A:0230 8B1EA601      MOV     BX,[01A6] 将p的偏移地址赋值到pa     DS:01A6=01AF
```

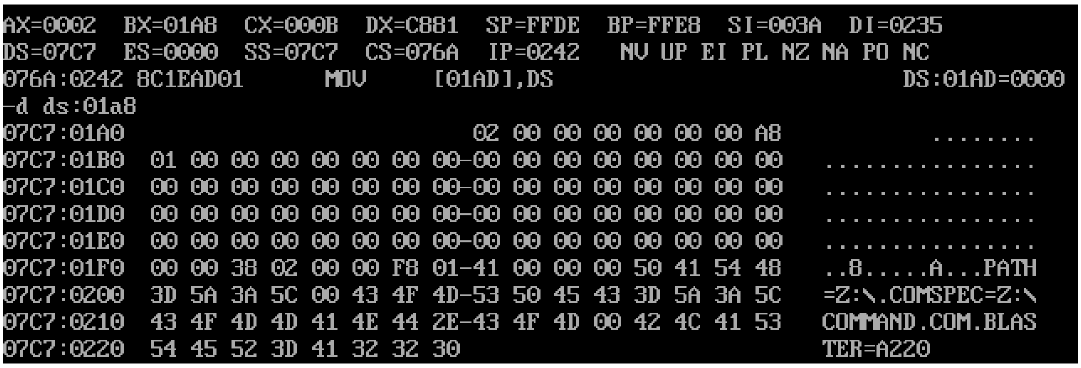
- 第六句 `**pa = **pa + 1;`

```
/*
    mov bx,word ptr DGROUP:_pa
    mov bx,word ptr [bx]
    mov al,byte ptr [bx]
    inc al
    mov bx,word ptr DGROUP:_pa
    mov bx,word ptr [bx] bx=01a6
    mov byte ptr [bx],al
*/
**pa = **pa + 1;
```



```
076A:0230 8B1EA601      MOV     BX,[01A6]
076A:0234 8B1F          MOV     BX,[BX]
076A:0236 8A07          MOV     AL,[BX]
076A:0238 FEC0          INC     AL
076A:023A 8B1EA601      MOV     BX,[01A6]
076A:023E 8B1F          MOV     BX,[BX]
076A:0240 8807          MOV     [BX],AL
076A:0242 8C1EAD01      MOV     [01AD],DS
076A:0246 C706AB01A801  MOV     WORD PTR [01AB],01A8
076A:024C C41EAB01      LES     BX,[01AB]
-g 242

AX=0002  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=0242   NV UP EI PL NZ NA PO NC
076A:0242 8C1EAD01      MOV     [01AD],DS                          DS:01AD=0000
```

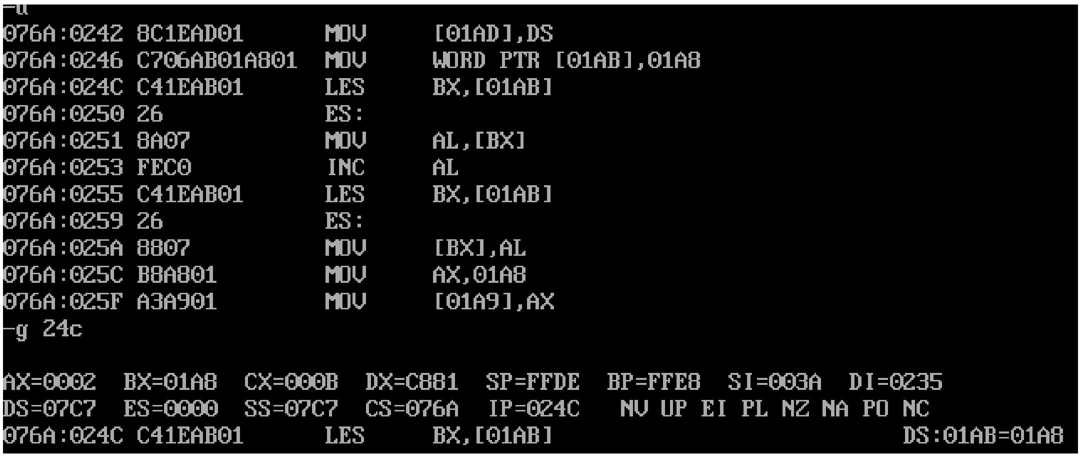
可以看到p指向的内存中的值增加一



```
AX=0002  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=0242   NV UP EI PL NZ NA PO NC
076A:0242 8C1EAD01      MOV     [01AD],DS                          DS:01AD=0000
-d ds:01a8
07C7:01A0                           02 00 00 00 00 00 00 A8           ........
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30                           TER=A220
```

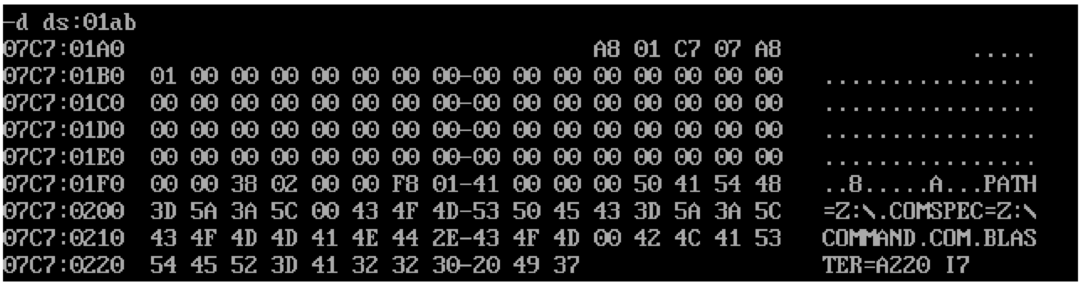
- 第七句 `pf = (**char** far *)&ch;`

```
/*                 01ad
    mov word ptr DGROUP:_pf+2,ds
                   [01a8]              01a8
    mov word ptr DGROUP:_pf,offset DGROUP:_ch
*/
pf = (char far *)&ch;
```



```
076A:0242 8C1EAD01      MOV     [01AD],DS
076A:0246 C706AB01A801  MOV     WORD PTR [01AB],01A8
076A:024C C41EAB01      LES     BX,[01AB]
076A:0250 26            ES:
076A:0251 8A07          MOV     AL,[BX]
076A:0253 FEC0          INC     AL
076A:0255 C41EAB01      LES     BX,[01AB]
076A:0259 26            ES:
076A:025A 8807          MOV     [BX],AL
076A:025C B8A801        MOV     AX,01A8
076A:025F A3A901        MOV     [01A9],AX
-g 24c

AX=0002  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=0000  SS=07C7  CS=076A  IP=024C   NV UP EI PL NZ NA PO NC
076A:024C C41EAB01      LES     BX,[01AB]                          DS:01AB=01A8
```

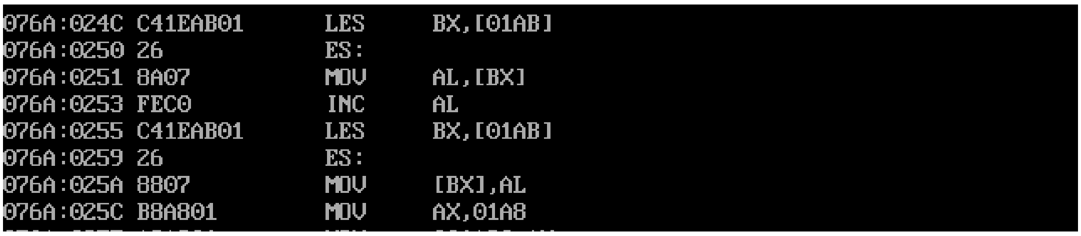
可以看到pf存在的是 `ch` 的地址



```
-d ds:01ab
07C7:01A0                                    A8 01 C7 07 A8              .....
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30-20 49 37                  TER=A220 I7
```

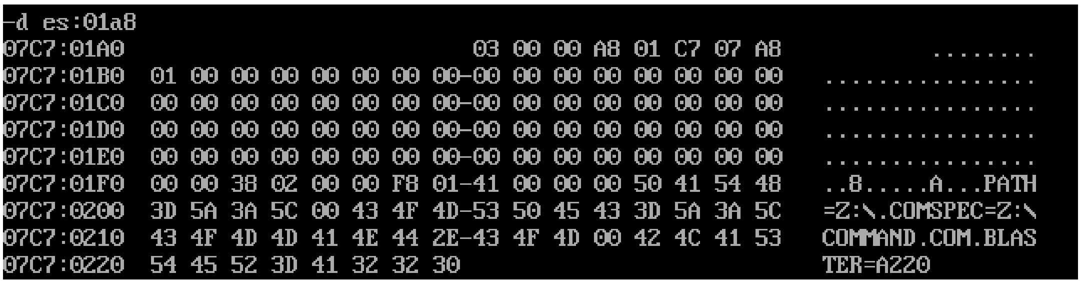
- 第八句 `*pf = *pf + 1;`

```
    /*
        les bx,dword ptr DGROUP:_pf
        mov al,byte ptr es:[bx]
        inc al
        les bx,dword ptr DGROUP:_pf
        mov byte ptr es:[bx],al
    */
    *pf = *pf + 1;
```



```
076A:024C C41EAB01      LES     BX,[01AB]
076A:0250 26            ES:
076A:0251 8A07          MOV     AL,[BX]
076A:0253 FEC0          INC     AL
076A:0255 C41EAB01      LES     BX,[01AB]
076A:0259 26            ES:
076A:025A 8807          MOV     [BX],AL
076A:025C B8A801        MOV     AX,01A8
```

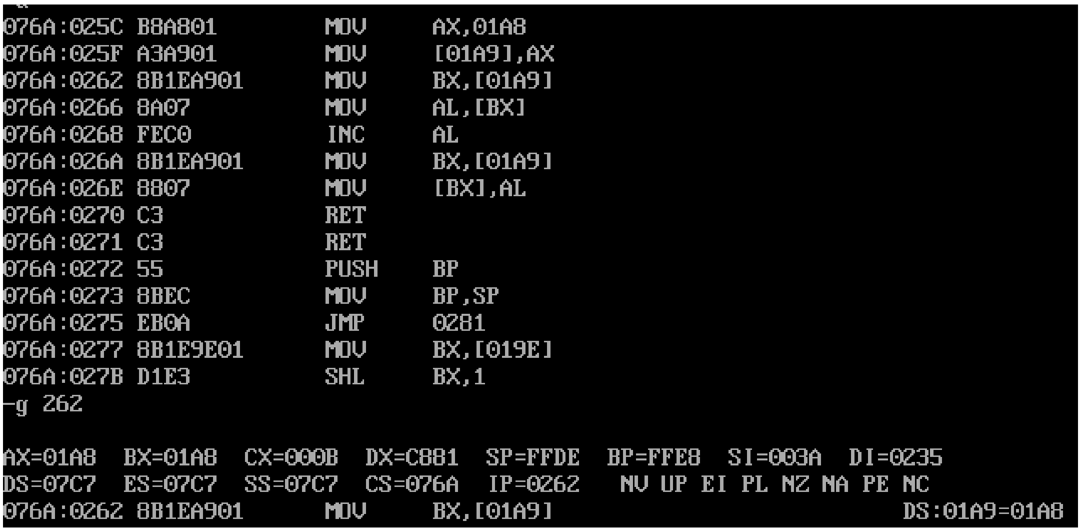
可以看到p指向的内存中的值增加一



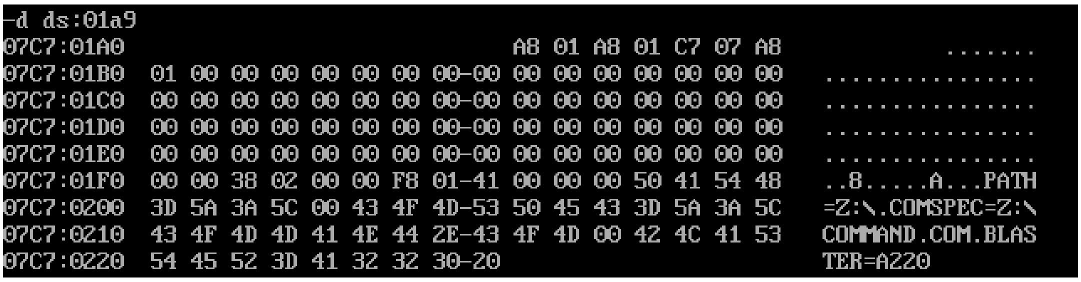
```
-d es:01a8
07C7:01A0                           03 00 00 A8 01 C7 07 A8           ........
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30                           TER=A220
```

- 第九句 `n = (**int**)&ch;`

```
    /*                        01a8
        mov ax,offset DGROUP:_ch
                            01a9
        mov word ptr DGROUP:_n,ax
    */
    n = (int)&ch;
```



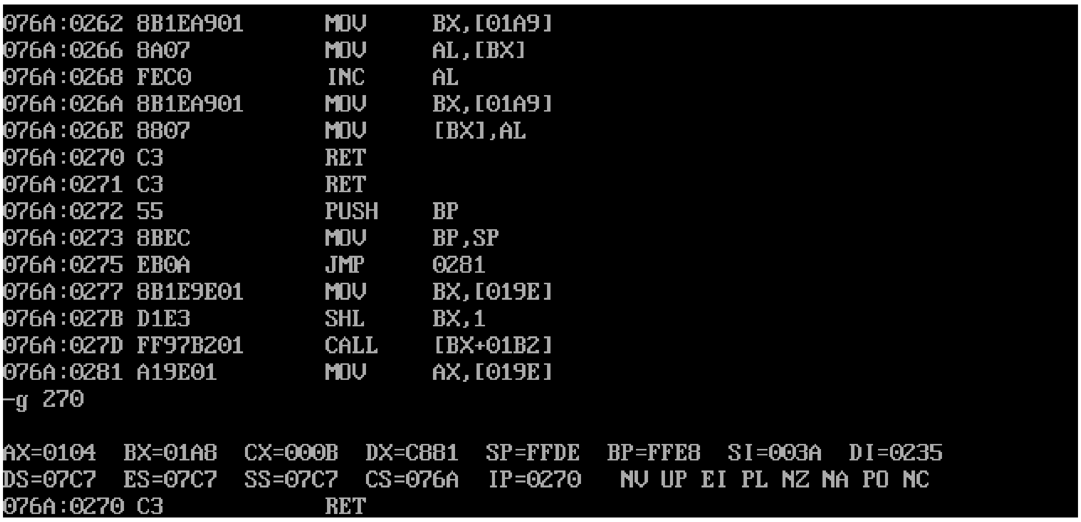
```
076A:025C B8A801        MOV     AX,01A8
076A:025F A3A901        MOV     [01A9],AX
076A:0262 8B1EA901      MOV     BX,[01A9]
076A:0266 8A07          MOV     AL,[BX]
076A:0268 FEC0          INC     AL
076A:026A 8B1EA901      MOV     BX,[01A9]
076A:026E 8807          MOV     [BX],AL
076A:0270 C3            RET
076A:0271 C3            RET
076A:0272 55            PUSH    BP
076A:0273 8BEC          MOV     BP,SP
076A:0275 EB0A          JMP     0281
076A:0277 8B1E9E01      MOV     BX,[019E]
076A:027B D1E3          SHL     BX,1
-g 262

AX=01A8  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=07C7  SS=07C7  CS=076A  IP=0262   NV UP EI PL NZ NA PE NC
076A:0262 8B1EA901      MOV     BX,[01A9]                          DS:01A9=01A8
```



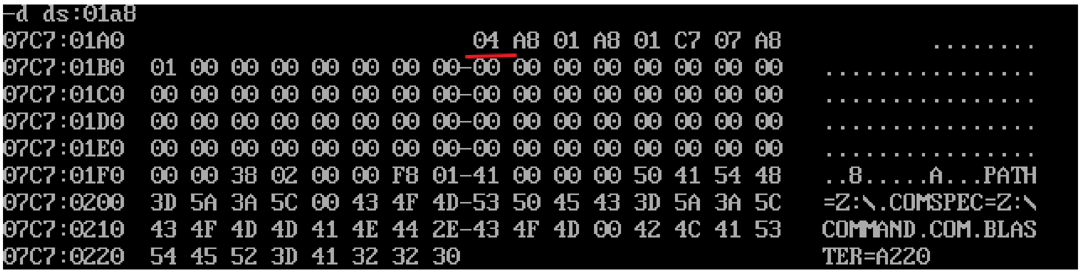
```
-d ds:01a9
07C7:01A0                              A8 01 A8 01 C7 07 A8            .......
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30-20                        TER=A220
```

- 第十句 `*(**char** *)n = *(**char** *)n + 1;`

```
    /*
        mov bx,word ptr DGROUP:_n
        mov al,byte ptr [bx]
        inc al
        mov bx,word ptr DGROUP:_n
        mov byte ptr [bx],al
    */
    *(char *)n = *(char *)n + 1;
```



```
076A:0262 8B1EA901      MOV     BX,[01A9]
076A:0266 8A07          MOV     AL,[BX]
076A:0268 FEC0          INC     AL
076A:026A 8B1EA901      MOV     BX,[01A9]
076A:026E 8807          MOV     [BX],AL
076A:0270 C3            RET
076A:0271 C3            RET
076A:0272 55            PUSH    BP
076A:0273 8BEC          MOV     BP,SP
076A:0275 EB0A          JMP     0281
076A:0277 8B1E9E01      MOV     BX,[019E]
076A:027B D1E3          SHL     BX,1
076A:027D FF97B201      CALL    [BX+01B2]
076A:0281 A19E01        MOV     AX,[019E]
-g 270

AX=0104  BX=01A8  CX=000B  DX=C881  SP=FFDE  BP=FFE8  SI=003A  DI=0235
DS=07C7  ES=07C7  SS=07C7  CS=076A  IP=0270   NV UP EI PL NZ NA PO NC
076A:0270 C3            RET
```



```
-d ds:01a8
07C7:01A0                           04 A8 01 A8 01 C7 07 A8          ........
07C7:01B0  01 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C7:01F0  00 00 38 02 00 00 F8 01-41 00 00 00 50 41 54 48   ..8.....A...PATH
07C7:0200  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07C7:0210  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07C7:0220  54 45 52 3D 41 32 32 30                           TER=A220
```

综上可以看出 `*p` 的功能是取出以 `p` 中数据作为偏移地址的内存中的值， `&p` 的功能就是取出 `p` 的偏移地址

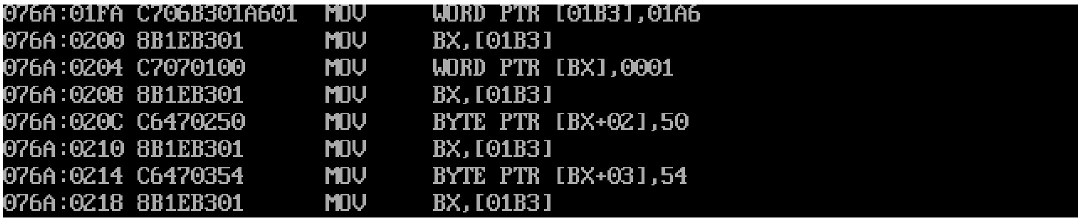
b.c

> 注意理解 struct指针的用法，指针“ + ”运算的意义。

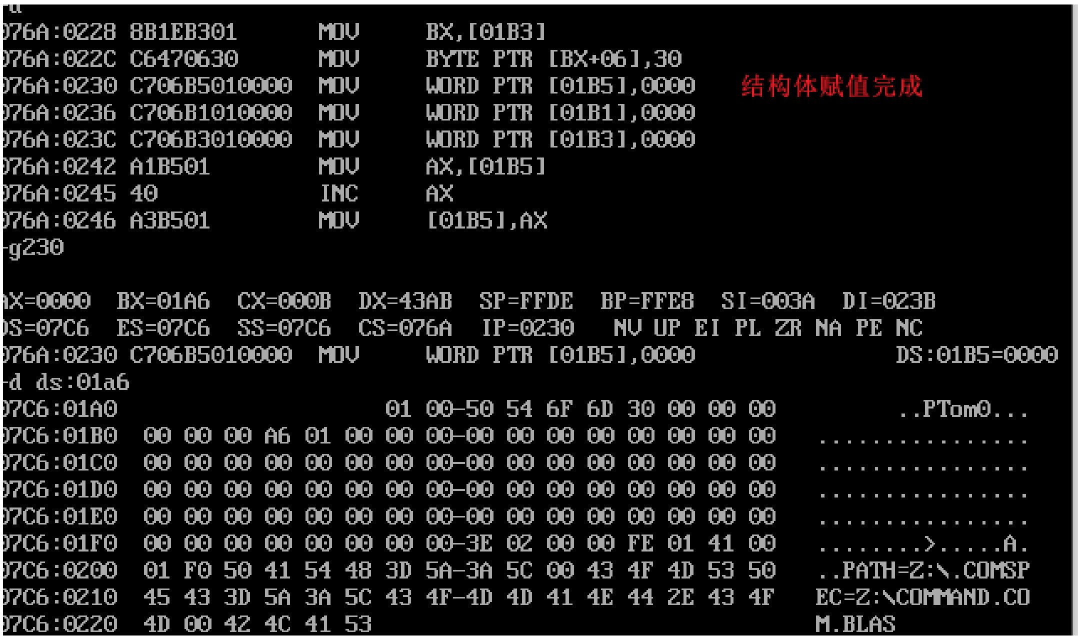
```
typedef struct {
    int number;
    char c;
    char name[8];
} stu;

stu a;

char *pchar;
int *pint;
stu *pstu;

```

```c
main() {
    pstu = &a;

    pstu->number = 1;
    (*pstu).c = 80;
    pstu->name[0] = 'T';
    pstu->name[1] = 'o';
    (*pstu).name[2] = 'm';
    (*pstu).name[3] = '0';

    pchar = 0;
    pint = 0;
    pstu = 0;

    pchar = pchar + 1;
    pint = pint + 1;
    pstu = pstu + 1;
}
```

- 通过汇编代码可以得出不论是指针的 `->` 运算和 `.` 运算最后翻译成的汇编都是把 `stu` 的首地址传给 `bx` 然后通过 `bx` 加上偏移来访问结构体变量的真正内存地址

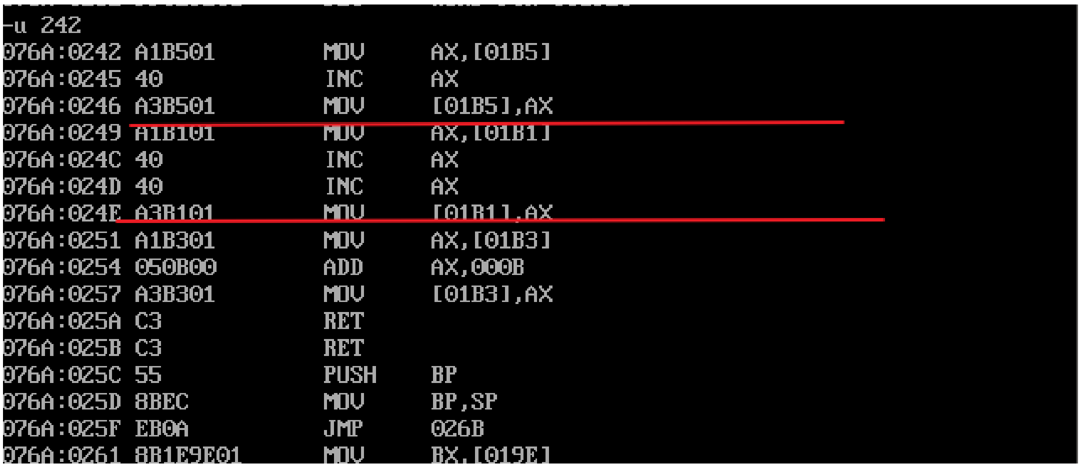


```
076A:01FA C706B301A601   MOV     WORD PTR [01B3],01A6
076A:0200 8B1EB301       MOV     BX,[01B3]
076A:0204 C7070100       MOV     WORD PTR [BX],0001
076A:0208 8B1EB301       MOV     BX,[01B3]
076A:020C C6470250       MOV     BYTE PTR [BX+02],50
076A:0210 8B1EB301       MOV     BX,[01B3]
076A:0214 C6470354       MOV     BYTE PTR [BX+03],54
076A:0218 8B1EB301       MOV     BX,[01B3]
```



```
-u
076A:0228 8B1EB301       MOV     BX,[01B3]
076A:022C C6470630       MOV     BYTE PTR [BX+06],30
076A:0230 C706B5010000   MOV     WORD PTR [01B5],0000
076A:0236 C706B1010000   MOV     WORD PTR [01B1],0000
076A:023C C706B3010000   MOV     WORD PTR [01B3],0000
076A:0242 A1B501         MOV     AX,[01B5]
076A:0245 40             INC     AX
076A:0246 A3B501         MOV     [01B5],AX
-g230

AX=0000  BX=01A6  CX=000B  DX=43AB  SP=FFDE  BP=FFE8  SI=003A  DI=023B
DS=07C6  ES=07C6  SS=07C6  CS=076A  IP=0230   NV UP EI PL ZR NA PE NC
076A:0230 C706B5010000   MOV     WORD PTR [01B5],0000               DS:01B5=0000
-d ds:01a6
07C6:01A0                    01 00-50 54 6F 6D 30 00 00 00         ..PTom0...
07C6:01B0  00 00 00 A6 01 00 00 00-00 00 00 00 00 00 00 00   ................
07C6:01C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C6:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C6:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07C6:01F0  00 00 00 00 00 00 00 00-3E 02 00 00 FE 01 41 00   ........>.....A.
07C6:0200  01 F0 50 41 54 48 3D 5A-3A 5C 00 43 4F 4D 53 50   ..PATH=Z:\.COMSP
07C6:0210  45 43 3D 5A 3A 5C 43 4F-4D 4D 41 4E 44 2E 43 4F   EC=Z:\COMMAND.CO
07C6:0220  4D 00 42 4C 41 53                                 M.BLAS
```

- 字符型指针加一就把指针内存中的数据增加一，整形指针加一就把指针内存中的数据增加二，结构体指针加一就是把指针内存中的数据增加结构体中各个变量长度总和。



```
-u 242
076A:0242 A1B501        MOV     AX,[01B5]
076A:0245 40            INC     AX
076A:0246 A3B501        MOV     [01B5],AX
076A:0249 A1B101        MOV     AX,[01B1]
076A:024C 40            INC     AX
076A:024D 40            INC     AX
076A:024E A3B101        MOV     [01B1],AX
076A:0251 A1B301        MOV     AX,[01B3]
076A:0254 050B00        ADD     AX,000B
076A:0257 A3B301        MOV     [01B3],AX
076A:025A C3            RET
076A:025B C3            RET
076A:025C 55            PUSH    BP
076A:025D 8BEC          MOV     BP,SP
076A:025F EB0A          JMP     026B
076A:0261 8B1E9E01      MOV     BX,[019E]
```

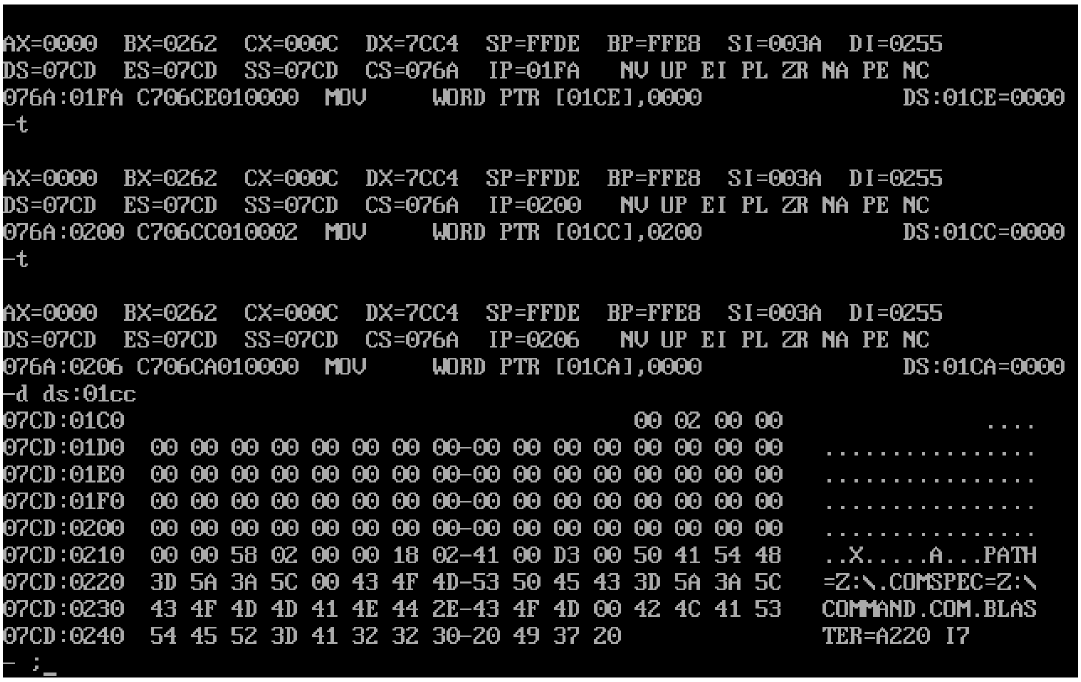
C.C

> 将字符串“hello world！”分别拷贝到从0:200、:210起始的内存中:将数组a分别拷贝到0:220、0:230起始的内存中。
>
> 注意理解“ [] ”运算的意义及数组名与指针的关系。
>
> 假设p是一个指针，p［n］的意义等同于＊（p＋n）

```c
char *p;
char far *pf;
char str[20] = "hello world!";
int a[8] = {11, 22, 33, 44, 55, 66, 77, 88};
int n;

main() {
    pf = (char far *)0x200;
    for (n = 0; str[n]; n++)
        *(pf + n) = str[n];

    p = str;
    pf = (char far *)0x210;
    for (n = 0; p[n]; n++)
        pf[n] = *(str + n);

    for (n = 0; n < 8; n++)
        ((int far *)0x220)[n] = *(a + n);
    for (n = 0; n < 8; n++)
        *(int far *)(0x230 + n * 2) = *(&a[0] + n);
}
```

- 1

```c
/*
    mov word ptr DGROUP:_pf+2,0
    mov word ptr DGROUP:_pf,512
*/
pf = (char far *)0x200;
```

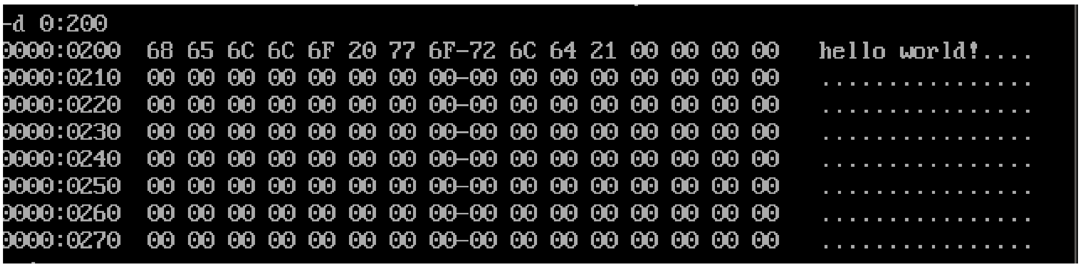


```
AX=0000  BX=0262  CX=000C  DX=7CC4  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=07CD  SS=07CD  CS=076A  IP=01FA   NV UP EI PL ZR NA PE NC
076A:01FA C706CE010000  MOV     WORD PTR [01CE],0000              DS:01CE=0000
-t

AX=0000  BX=0262  CX=000C  DX=7CC4  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=07CD  SS=07CD  CS=076A  IP=0200   NV UP EI PL ZR NA PE NC
076A:0200 C706CC010002  MOV     WORD PTR [01CC],0200              DS:01CC=0000
-t

AX=0000  BX=0262  CX=000C  DX=7CC4  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=07CD  SS=07CD  CS=076A  IP=0206   NV UP EI PL ZR NA PE NC
076A:0206 C706CA010000  MOV     WORD PTR [01CA],0000              DS:01CA=0000
-d ds:01cc
07CD:01C0                                      00 02 00 00               ....
07CD:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0200  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0210  00 00 58 02 00 00 18 02-41 00 D3 00 50 41 54 48   ..X.....A...PATH
07CD:0220  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07CD:0230  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07CD:0240  54 45 52 3D 41 32 32 30-20 49 37 20               TER=A220 I7
- ;_
```

- 2

```
/*
;	?debug	L 9	[01ca]
	mov	word ptr DGROUP:_n,0
	jmp	short @5
@4:
;	?debug	L 10
	mov	bx,word ptr DGROUP:_n ;[01ca]
	mov	al,byte ptr DGROUP:_str[bx] ;[bx+0194]
	les	bx,dword ptr DGROUP:_pf
	add	bx,word ptr DGROUP:_n
	mov	byte ptr es:[bx],al
@3:
	inc	word ptr DGROUP:_n
@5:			[01ca]
	mov	bx,word ptr DGROUP:_n
	cmp	byte ptr DGROUP:_str[bx],0
	jne	@4
*/
for (n = 0; str[n]; n++)
	*(pf + n) = str[n];
```

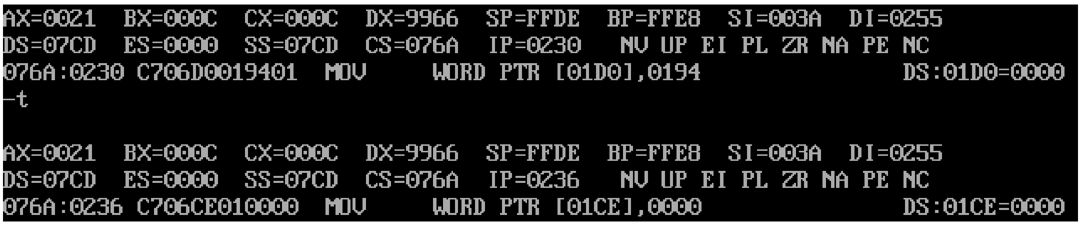
初始化后



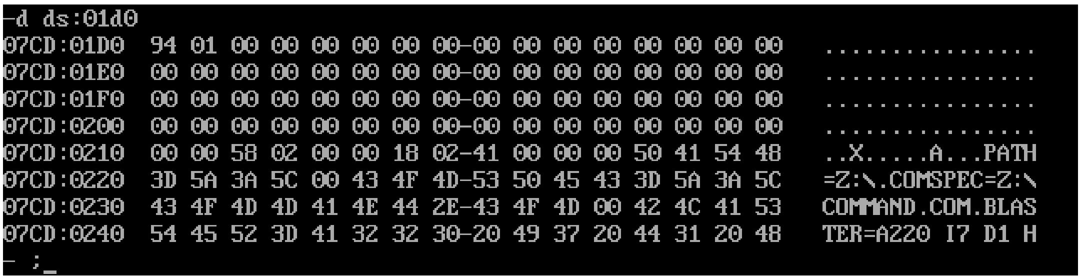
```
-d 0:200
0000:0200  68 65 6C 6C 6F 20 77 6F-72 6C 64 21 00 00 00 00   hello world!....
0000:0210  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0220  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

- 3

```
/*                  01d0                0194
    mov word ptr DGROUP:_p,offset DGROUP:_str
*/
p = str;
```



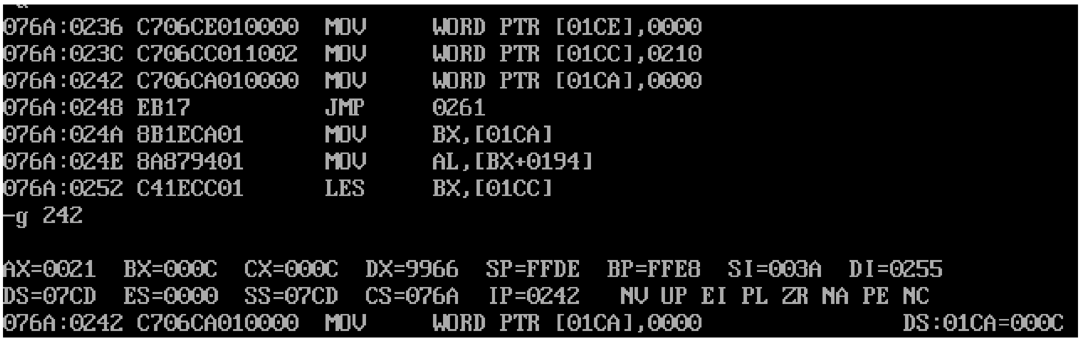
```
AX=0021  BX=000C  CX=000C  DX=9966  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=0000  SS=07CD  CS=076A  IP=0230   NV UP EI PL ZR NA PE NC
076A:0230 C706D0019401  MOV     WORD PTR [01D0],0194               DS:01D0=0000
-t

AX=0021  BX=000C  CX=000C  DX=9966  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=0000  SS=07CD  CS=076A  IP=0236   NV UP EI PL ZR NA PE NC
076A:0236 C706CE010000  MOV     WORD PTR [01CE],0000               DS:01CE=0000
```



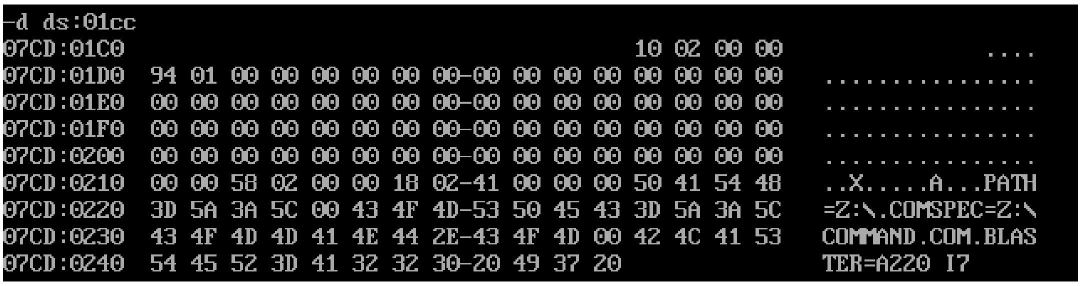
```
-d ds:01d0
07CD:01D0  94 01 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0200  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0210  00 00 58 02 00 00 18 02-41 00 00 00 50 41 54 48   ..X.....A...PATH
07CD:0220  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07CD:0230  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07CD:0240  54 45 52 3D 41 32 32 30-20 49 37 20 44 31 20 48   TER=A220 I7 D1 H
- ;_
```

- 4

```
/*
    mov word ptr DGROUP:_pf+2,0
    mov word ptr DGROUP:_pf,528
*/

pf = (char far *)0x210;
```



```
076A:0236 C706CE010000  MOV     WORD PTR [01CE],0000
076A:023C C706CC011002  MOV     WORD PTR [01CC],0210
076A:0242 C706CA010000  MOV     WORD PTR [01CA],0000
076A:0248 EB17          JMP     0261
076A:024A 8B1ECA01      MOV     BX,[01CA]
076A:024E 8A879401      MOV     AL,[BX+0194]
076A:0252 C41ECC01      LES     BX,[01CC]
-g 242

AX=0021  BX=000C  CX=000C  DX=9966  SP=FFDE  BP=FFE8  SI=003A  DI=0255
DS=07CD  ES=0000  SS=07CD  CS=076A  IP=0242   NV UP EI PL ZR NA PE NC
076A:0242 C706CA010000  MOV     WORD PTR [01CA],0000               DS:01CA=000C
```



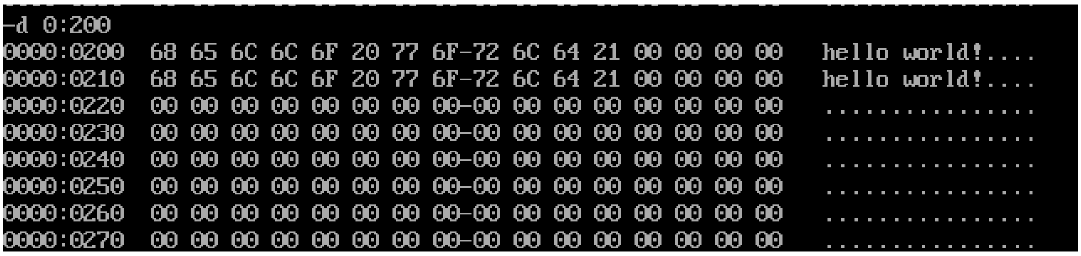
```
-d ds:01cc
07CD:01C0                                      10 02 00 00               ....
07CD:01D0  94 01 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0200  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
07CD:0210  00 00 58 02 00 00 18 02-41 00 00 00 50 41 54 48   ..X.....A...PATH
07CD:0220  3D 5A 3A 5C 00 43 4F 4D-53 50 45 43 3D 5A 3A 5C   =Z:\.COMSPEC=Z:\
07CD:0230  43 4F 4D 4D 41 4E 44 2E-43 4F 4D 00 42 4C 41 53   COMMAND.COM.BLAS
07CD:0240  54 45 52 3D 41 32 32 30-20 49 37 20               TER=A220 I7
```

- 5

```
/*
;    ?debug  L 14
     mov word ptr DGROUP:_n,0
     jmp short @9
@8:
;    ?debug  L 15
     mov bx,word ptr DGROUP:_n
     mov al,byte ptr DGROUP:_str[bx]
     les bx,dword ptr DGROUP:_pf
     add bx,word ptr DGROUP:_n
     mov byte ptr es:[bx],al
@7:
     inc word ptr DGROUP:_n
@9:
     mov bx,word ptr DGROUP:_p
```

```
    add bx,word ptr DGROUP:_n
    cmp byte ptr [bx],0
    jne @8
*/
for (n = 0; p[n]; n++)
        pf[n] = *(str + n);
```

拷贝hello world从0:200 -> 0:210

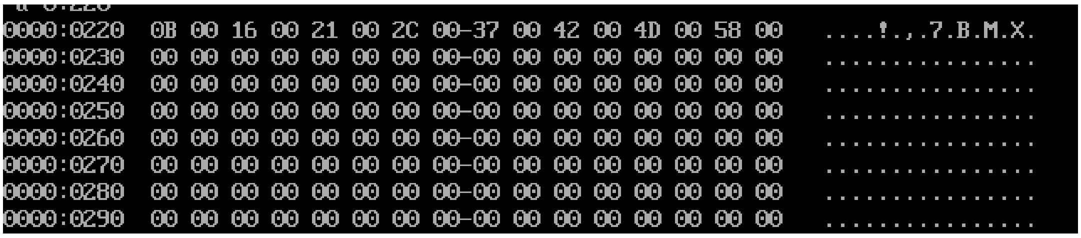


```
-d 0:200
0000:0200  68 65 6C 6C 6F 20 77 6F-72 6C 64 21 00 00 00 00   hello world!....
0000:0210  68 65 6C 6C 6F 20 77 6F-72 6C 64 21 00 00 00 00   hello world!....
0000:0220  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

- 6

```
/*
@6:
;   ?debug  L 17
    mov word ptr DGROUP:_n,0
    jmp short @13
@12:
;   ?debug  L 18
    mov bx,word ptr DGROUP:_n
    shl bx,1
    mov ax,word ptr DGROUP:_a[bx]
    mov dx,word ptr DGROUP:_n
    shl dx,1
    xor bx,bx
    mov es,bx
    mov bx,544
    add bx,dx
    mov word ptr es:[bx],ax
@11:
    inc word ptr DGROUP:_n
@13:
    cmp word ptr DGROUP:_n,8
    jl  @12
*/
for (n = 0; n < 8; n++)
        ((int far *)0x220)[n] = *(a + n);
```

初始化

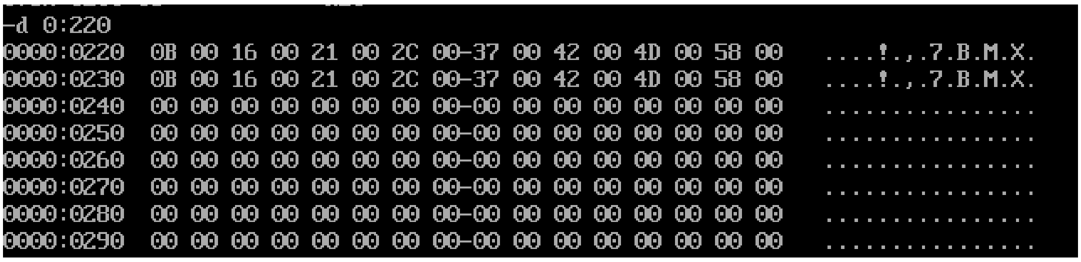


```
0000:0220  0B 00 16 00 21 00 2C 00-37 00 42 00 4D 00 58 00   ....!.,.7.B.M.X.
0000:0230  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0280  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0290  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

- 7

```
/*
@10:
;   ?debug  L 19
    mov word ptr DGROUP:_n,0
    jmp short @17
@16:
;   ?debug  L 20
    mov bx,word ptr DGROUP:_n
    shl bx,1
    mov ax,word ptr DGROUP:_a[bx]
    push    ax
    mov ax,word ptr DGROUP:_n
    shl ax,1
    add ax,560
    cwd
    mov bx,ax
    mov es,dx
    pop ax
    mov word ptr es:[bx],ax
@15:
    inc word ptr DGROUP:_n
@17:
    cmp word ptr DGROUP:_n,8
    jl  @16
*/
for (n = 0; n < 8; n++)
        *(int far *)(0x230 + n * 2) = *(&a[0] + n);
```

拷贝a从0:220->0:230



```
-d 0:220
0000:0220  0B 00 16 00 21 00 2C 00-37 00 42 00 4D 00 58 00   ....!.,.7.B.M.X.
0000:0230  0B 00 16 00 21 00 2C 00-37 00 42 00 4D 00 58 00   ....!.,.7.B.M.X.
0000:0240  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0250  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0260  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0270  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0280  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0000:0290  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

综上 `p[n]` 的意思是访问以 `p` 为基地址 `n` 为偏移地址中的数据，数组名 `p` 和指针 `*p` 存储的都是数据的起始地址