指针及字符串操作分析

源程序：

```
#include <stdio.h>

int main (void) {
    int array[5] = {0, 1, 2, 3, 4};
    int *p = array;
    char *str = "Hello, world!";
    int cnt;

    for (cnt = 0; cnt < 5; cnt ++) {
        p++;
    }

    for (cnt = 0; cnt < 13; cnt ++) {
        str ++;
    }

    return 0;
}
```

反汇编结果：

```
3:    int main (void) {
00401010   push        ebp
00401011   mov         ebp,esp
00401013   sub         esp,60h
00401016   push        ebx
00401017   push        esi
00401018   push        edi
00401019   lea         edi,[ebp-60h]
0040101C   mov         ecx,18h
00401021   mov         eax,0CCCCCCCCh
00401026   rep stos    dword ptr [edi]
4:        int array[5] = {0, 1, 2, 3, 4};
00401028   mov         dword ptr [ebp-14h],0
0040102F   mov         dword ptr [ebp-10h],1
00401036   mov         dword ptr [ebp-0Ch],2
0040103D   mov         dword ptr [ebp-8],3
00401044   mov         dword ptr [ebp-4],4
5:        int *p = array;
0040104B   lea         eax,[ebp-14h]
0040104E   mov         dword ptr [ebp-18h],eax
6:        char *str = "Hello, world!";
00401051   mov         dword ptr [ebp-1Ch],offset string "Hello, world!" (0042201c)
7:        int cnt;
8:
```

```
9:        for (cnt = 0; cnt < 5; cnt ++) {
00401058   mov         dword ptr [ebp-20h],0
0040105F   jmp         main+5Ah (0040106a)
00401061   mov         ecx,dword ptr [ebp-20h]
00401064   add         ecx,1
00401067   mov         dword ptr [ebp-20h],ecx
0040106A   cmp         dword ptr [ebp-20h],5
0040106E   jge         main+6Bh (0040107b)
10:           p++;
00401070   mov         edx,dword ptr [ebp-18h]
00401073   add         edx,4
00401076   mov         dword ptr [ebp-18h],edx
11:       }
00401079   jmp         main+51h (00401061)
12:
13:       for (cnt = 0; cnt < 13; cnt ++) {
0040107B   mov         dword ptr [ebp-20h],0
00401082   jmp         main+7Dh (0040108d)
00401084   mov         eax,dword ptr [ebp-20h]
00401087   add         eax,1
0040108A   mov         dword ptr [ebp-20h],eax
0040108D   cmp         dword ptr [ebp-20h],0Dh
00401091   jge         main+8Eh (0040109e)
14:           str ++;
00401093   mov         ecx,dword ptr [ebp-1Ch]
00401096   add         ecx,1
00401099   mov         dword ptr [ebp-1Ch],ecx
15:       }
0040109C   jmp         main+74h (00401084)
16:
17:       return 0;
0040109E   xor         eax,eax
18:   }
004010A0   pop         edi
004010A1   pop         esi
004010A2   pop         ebx
004010A3   mov         esp,ebp
004010A5   pop         ebp
004010A6   ret
```

分析：

- 元素的地址表示

  数组中个元素的地址表示通过寄存器BP间接寻址，当调用main函数时，传送指令MOV将SP的值传给BP，而后对栈的元素取用通过相对BP的偏移间接完成。

- 指针的赋值

  首先在栈中开辟一个字存指针，通过相对于BP的偏移保存地址。

- 指针的移动

  通过保存指针的栈里的值的算术运算来移动指针。