



南开大学  
Nankai University

南 开 大 学

计 算 机 学 院

汇编语言与逆向技术实验报告

---

实验 2 dec2hex

---

杨冰雪

年级：2021 级

专业：计算机科学与技术

2023 年 10 月 24 日

## 目录

一、 实验目的	1
二、 实验内容	1
三、 实验代码	2
(一) 十进制字符串转化为十进制数 . . . . .	3
(二) 十六进制字符串的初始化 . . . . .	4
(三) 十进制数转化为十六进制字符串 . . . . .	4
四、 实验步骤	4
(一) 编译 . . . . .	4
(二) 链接 . . . . .	4
五、 实验结果	4
六、 实验总结	5

## 一、 实验目的

1. 熟悉汇编语言的数据传送、寻址和算术运算；
2. 熟悉汇编语言过程的定义和使用；
3. 熟悉十进制和十六进制的数制转换

## 二、 实验内容

编写汇编程序 dec2hex.asm, 编译成 dec2hex.exe。dec2hex.exe 的功能是将 Windows 命令行输入的十进制无符号整数, 转换成对应的十六进制整数, 输出在 Windows 命令行中, 如图 1 所示。

输入的十进制无符号整数的范围是 0 到 4294967295 ( $2^{32}-1$ )。

输出对应的十六进制整数, 对应的范围是 00000000h 到 FFFFFFFFh。

```
D:\>dec2hex.exe
Please input a decimal number( 0~ 4294967295): 100
The hexadecimal number is : 00000064
```

图 1: dec2hex.exe 将十进制 100 转换成十六进制 00000064

1. 使用 StdIn 函数获得用户输入的十进制整数。StdIn 函数的定义在 \masm32\include\masm32.inc, 库文件是 \masm32\lib\masm32.lib。StdIn 函数的定义“StdIn PROTO :DWORD, :DWORD”, 有两个参数, 第一个是内存存储空间的起始地址, 第二个是内存存储空间的大小。函数的例子:

```
.data
buf BYTE 20 DUP(0)
.code
invoke StdIn, addr buf, 20
invoke StdOut, addr buf
```

2. 用户输入的十进制数对应的 ASCII 编码字符串存储在内存中, 编写过程 dec2dw, 将十进制字符串转换成 DWORD 数据。例如, 将字符串“100”转换成 DWORD 数据 00000064h。
3. 编写过程 dw2hex, 将 DWORD 数据转换成十六进制数的 ASCII 字符串。例如, 将 DWORD 数据 00000064h 转换成 ASCII 字符串“00000064”。
4. 使用 StdOut 函数在 Windows 命令行中输出十六进制整数的 ASCII 字符串。StdOut 函数的定义在 \masm32\include\masm32.inc, 库文件是 \masm32\lib\masm32.lib。StdOut 函数的定义“StdOut PROTO :DWORD”, 只有一个参数, 是内存存储空间的起始地址。函数使用的例子同 StdIn 函数的例子。
5. 使用 ml 将 dec2hex.asm 文件汇编到 dec2hex.obj 目标文件, 编译命令:

```
"\masm32\bin\ml /c /coff dec2hex.asm"
```

6. 使用 link 将目标文件 dec2hex.obj 链接成 dec2hex.exe 可执行文件, 链接命令:

```
"\masm32\bin\link /SUBSYSTEM:CONSOLE dec2hex.obj"
```

### 三、 实验代码

dec2hex.asm

```
1 .386
2 .model flat , stdcall
3 option casemap :none
4 include D:\masm32\include\windows.inc
5 include D:\masm32\include\kernel32.inc
6 include D:\masm32\include\masm32.inc
7 includelib D:\masm32\lib\kernel32.lib
8 includelib D:\masm32\lib\masm32.lib
9
10 .data
11     str1 BYTE "please_input_a_dec_number(0-4294967295):", 0
12     str2 BYTE "the_dec_num_to_hex_number_is:", 0
13     buf BYTE 20 DUP(0) ;十进制字符串
14     hexstr BYTE 9 DUP(48) ;十六进制字符串
15     hex_index DWORD 16
16     dec_index DWORD 10
17     decnum DWORD 0 ;十进制数
18     remainder DWORD 0 ;余数
19
20 .code
21 start:
22     invoke StdOut,addr str1
23     invoke StdIn,addr buf,20
24
25     Mov eax,0 ;用来存放十进制数
26     Mov esi,0
27 ;将十进制字符串转化为十进制数
28 str2dec:
29     cmp buf[esi],0
30     jz init_hex
31     Mov bh,buf[esi]
32     Sub bh,48
33     Movzx ebx,bh
34     Mul dec_index ;eax中存放乘法的结果且edx也会被影响
35     Add eax,ebx
36     inc esi
37     jmp str2dec
38 ;初始化十六进制字符串最后一位
39 init_hex:
40     ;给十六进制字符串最后一位加上'\0'
41     Mov decnum,eax
42     Mov esi,0
43     Mov esi,offset hexstr
44     Add esi,8
```

```
45     mov bh,0
46     mov [esi],bh
47 ;十进制数转化为十六进制字符串
48 dec2hex:
49     mov eax,0
50     mov edx,0
51
52     mov eax,decnum
53     div hex_index ;eax中存放除法的商, edx中存放除法的余数
54     mov decnum,eax
55     cmp edx,10 ;当余数大于等于10时, 把余数变为字母
56     jnc toletter
57     add edx,48
58 ;保存余数
59 save_remainder:
60     dec esi
61     Mov remainder,edx
62     Mov dh,BYTE ptr remainder
63     Mov [esi],dh
64
65     cmp eax,0 ;判断是否除完
66     jz display
67     jmp dec2hex
68 ;将余数大于等于10的数转化为字母
69 toletter:
70     add edx,55
71     jmp save_remainder
72 ;展示最终结果
73 display:
74     invoke StdOut,addr str2
75     invoke StdOut,addr hexstr
76
77     invoke ExitProcess,0
78 END start
79 }
```

### (一) 十进制字符串转化为十进制数

因为 masn 库的 StdIn 和 StdOut 都只至此字符串类型, 所以要把输入的十进制字符串转化为十进制的数值。其思路如下:

- 将 eax 寄存器初始化为 0, 用来储存计算出的十进制数
- 遍历输入的字符串, 对于每次取得的字符, 减去 48, 从而得到其数值
- 将 eax 每次乘 10, 再加上字符被转化后的数值
- 循环直至取得的字符为'\0', 表示遍历字符串结束

## (二) 十六进制字符串的初始化

因为十六进制字符串的最后一位必须是'\0'，所以先获取十六进制字符串的地址，再将最后一位地址对应的内容赋值为 0，从而完成字符串的初始化。

## (三) 十进制数转化为十六进制字符串

此过程中融合了十进制数转十六进制，十六进制数转十六进制字符串两个过程。其思路如下：

- 将得到的十进制数除以 16，在 eax 寄存器中会储存商，在 ebx 寄存器中会储存余数
- 把余数与 10 比较，如果大于等于 10，将其转化为对应的字母，否则将其转化为字符
- 得到的余数对应的是低位，所以从十六进制字符串最后一位向第一位方向移动，依次储存得到的余数
- 判断商是否为 0，为 0 表示计算结束，可以展示转换的结果，不为 0 表示还需要继续计算，那就继续循环下去直至商为 0

## 四、 实验步骤

### (一) 编译

使用命令，将汇编文件编译为可重定位的目标文件。

```
E:\npp_code>D:\masm32\bin\ml /c /coff dec2hex.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.

Assembling: dec2hex.asm

*****
ASCII build
*****
```

图 2: 编译过程

### (二) 链接

使用命令，将目标文件链接为可执行文件。

```
E:\npp_code>D:\masm32\bin\link /SUBSYSTEM:CONSOLE dec2hex.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

图 3: 链接过程

## 五、 实验结果

dec2hex.exe 的测试结果如下：

```
E:\npp_code>. \dec2hex.exe
please input a dec number(0-4294967295):0
the dec num to hex number is:00000000
E:\npp_code>. \dec2hex.exe
please input a dec number(0-4294967295):4294967295
the dec num to hex number is:FFFFFFFF
E:\npp_code>. \dec2hex.exe
please input a dec number(0-4294967295):100
the dec num to hex number is:00000064
```

图 4: 测试结果

## 六、 实验总结

通过本次实验, 让对汇编指令有了更加深入的了解, 懂得了 masm 进行汇编的大体逻辑, 了解了各寄存器大体的功能, 学会了大部分伪指令的基本用法。熟悉了汇编语言的数据传送、寻址和算术运算, 明白了汇编语言过程的定义和使用; 也熟悉十进制和十六进制的数制转换。

NIJUB