

## 实验 2 dex2hex

### 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、实验环境

MASM32 编译环境

Windows 命令行窗口

### 三、过程说明（编译、链接、测试过程说明）

#### 1. 编辑：编写汇编程序dec2hex.asm。

程序的功能是将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。

（输入的十进制无符号整数的范围是0到4294967295，输出对应的十六进制整数，对应的范围是00000000h到FFFFFFFFh）。

其中包括：

- StdIn函数：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
- dec2dw过程：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ASCII字符串转换成DWORD数据。
- dw2hex过程：将DWORD数据转换成十六进制数的ASCII字符串。
- StdOut函数：使用StdOut函数在Windows命令函中输出十六进制整数的ASCII字符串。

#### 2. 编译：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 /Zd /coff dec2hex.asm”

```
C:\Users\98712\OneDrive\桌面>\masm32\bin\ml /c /Zd /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
*****
```

### 3. 链接：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”

```
C:\Users\98712\OneDrive\桌面>\masm32\bin\Link /SUBSYSTEM:CONSOLE dec2hex.obj
Microsoft (R) Incremental Linker Version 5.12.8078
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.
```

### 4. 测试：直接执行dec2hex.exe可执行文件。

```
C:\Users\98712\OneDrive\桌面>dec2hex.exe
Please input a decimal number(0~4294967295):100
The hexadecimal number is:00000064
```

## 四、源代码

```
.386
.model flat,stdcall
option casemap :none
include \masm32\include\windows.inc
include \masm32\include\masm32.inc
include \masm32\include\kernel32.inc
includelib \masm32\lib\masm32.lib
includelib \masm32\lib\kernel32.lib

.data
    ask_str BYTE "Please input a decimal number(0~4294967295):",0
    output_str BYTE "The hexadecimal number is:",0
    var BYTE 12 DUP(0)
    confirmed_10d DWORD 10d
    lpHexString BYTE "0123456789ABCDEFF"
    answ BYTE 8 DUP(0)

.code
dec2dw proc
    mov eax,0h
    mov esi,0h
```

L1:

```
sub var[esi],30h
movzx edx,var[esi]
add eax,edx
inc esi
cmp var[esi],0d
je EXIT1
mul confirmed_10d
jmp L1
```

EXIT1:RET

dec2dw endp

dw2hex\_my proc

```
mov ecx,8h
mov edi,0h
```

L3:

```
mov esi,eax
and esi,0F0000000h
shr esi,28
movzx edx,byte ptr[lpHexString+esi]
mov BYTE ptr answ[edi],dl
shl eax,4
inc edi
dec ecx
cmp ecx,0h
jne L3
RET
```

dw2hex\_my endp

start:

```
invoke StdOut, addr ask_str
invoke StdIn,addr var,12
CALL dec2dw
CALL dw2hex_my
invoke StdOut,addr output_str
invoke StdOut,addr answ
invoke ExitProcess,0
```

end start