

实验内容

- **实验1：** 输入输出实验（第11~12周）
- **实验2：** 分支程序设计实验（第12周）
- **实验3：** 排序程序设计实验（第13周）
- **实验4：** 子程序设计实验（第14周）
- **实验5：** 整数加减运算实验（第15周）
- **实验6：** 浮点运算实验（第16~17周）

- **实验7：** **8255**接口程序设计实验（暂无安排）

实验1： 输入输出实验

- **实验目的：**

1. 掌握汇编程序的基本编写方式
2. 学习汇编语言的基础I/O操作，为后续实验做准备

- **实验内容：**

1. 创建一个名为Input1. txt的文件
2. 使用键盘输入一个包含大写字母、小写字母和数字的字符串，并将这个字符串写入文件中
3. 读取这个文件，将小写字母转换成大写形式，大写字母和数字保持不变，最后整个字符串输出到屏幕，并写入Output1. txt文件

实验2：分支程序设计实验

- 实验目的：

1. 掌握汇编语言中比较和跳转命令
2. 掌握汇编语言中循环命令的使用

- 实验内容：

1. 键盘输入一个数字N ($0 < N < 10$)
2. 把 $1 \sim N^2$ 的自然数按行顺序存入二维数组
3. 在屏幕上打印出该数组的左下半三角

***例：** N=6时，应打印成类似下图形式：

```
1
7 8
13 14 15
19 20 21 22
25 26 27 28 29
31 32 33 34 35 36
```

实验3：排序程序设计实验

- **实验目的：**

1. 综合运用汇编中的I/O、比较、跳转、循环等指令
2. 掌握用汇编语言实现数字排序程序的方法

- **实验内容：**

1. 从名为“Input3.txt”的文本文件中读取一组数字
2. 将这些数字从小到大进行排序
3. 将这些数字按照排序后的次序打印在屏幕上
4. 数字范围 $[-1024, 1023]$ ，排序数字不超过100个
5. 排序算法不限

实验4：子程序设计实验

- 实验目的：

1. 掌握汇编语言中子程序的编写与调用方法
2. 理解汇编语言中的递归调用

- 实验内容：

1. 编写一个程序，使用子程序调用的方式计算 $n!$
2. n 的值通过键盘输入
3. n 的范围(0, 20)
4. 程序设计中请注意运算结果的范围

实验5： 整数加减计算实验

- 实验目的：

1. 掌握汇编语言的整数加减运算
2. 掌握汇编语言的堆栈操作

- 实验内容：

1. 编程计算任一整数加减运算表达式
2. 表达式从键盘输入，可带括号
3. 表达式的长度不超过1024个字节
4. 表达式示例： $15 - (4 - 2)$

实验6：浮点运算设计实验

- 实验目的：

1. 理解浮点数的存储和运算原理
2. 掌握x87算术协处理器的运算指令
3. 掌握汇编语言中浮点数与字符串的转换方法

- 实验内容：

1. 从键盘输入 x 和 a_1 、 a_2 、 a_3
2. 计算 $a_1\sqrt{x} + a_2 \log_2 x + a_3 \sin(x)$ 并将结果打印到屏幕
3. 当 x 小于0时，输出信息 “Error: $x < 0!$ ”

***注：**因为EMU8086不支持FPU指令，所以此实验无法在EMU8086中完成

内容

- 时间安排
- 实验环境
- 实验内容
- **实验教程**
 - **MASMv5.0+DOSBOX**
 - **EMU8086**
 - **32位汇编环境**

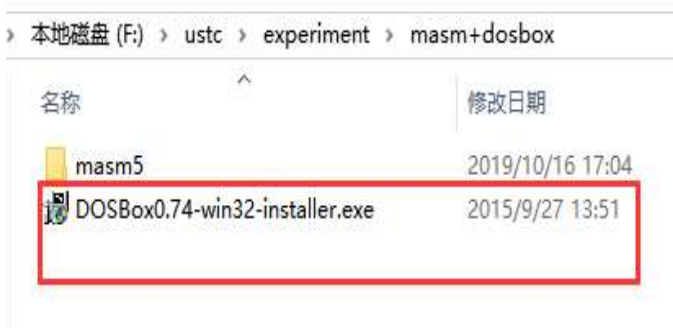
MASMv5.0+DOSBOX配置(1)

1. 在非中文目录下新建文件夹，将MASMv5.0套件解压至此

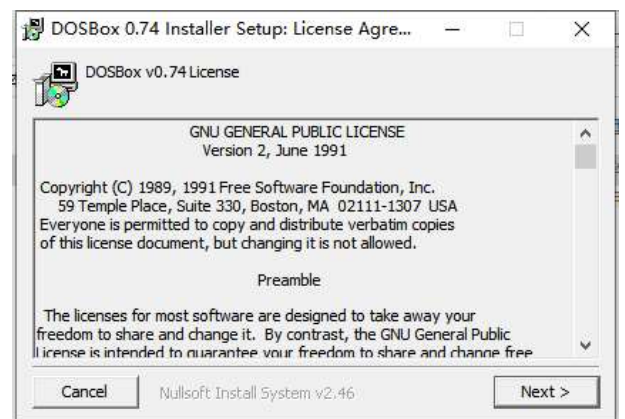


名称	修改日期	类型	大小
cmd.exe	2008/4/14 20:00	应用程序	460 KB
debug.exe	2015/9/27 13:51	应用程序	21 KB
edit.com	2008/4/14 20:00	MS-DOS 应用程序	69 KB
LINK.EXE	1996/5/12 16:28	应用程序	39 KB
MAKE.EXE	1996/5/12 16:28	应用程序	24 KB
MASM.EXE	1996/5/12 16:28	应用程序	65 KB
test.asm	2008/12/12 10:00	ASM 文件	1 KB

2. 安装DOSBOX

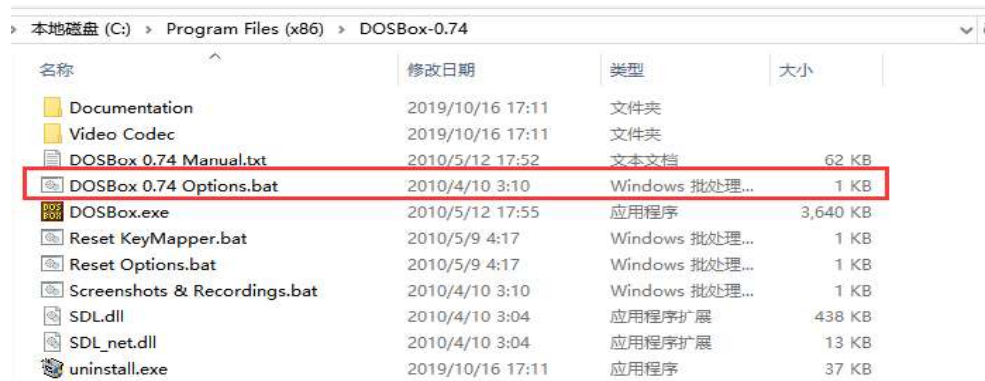


名称	修改日期
masm5	2019/10/16 17:04
DOSBox0.74-win32-installer.exe	2015/9/27 13:51



MASMv5.0+DOSBOX配置(2)

3. 在DOSBOX的安装目录下运行DOSBox 0.74-2 Options.bat 来进行DOSBOX的启动参数配置



名称	修改日期	类型	大小
Documentation	2019/10/16 17:11	文件夹	
Video Codec	2019/10/16 17:11	文件夹	
DOSBox 0.74 Manual.txt	2010/5/12 17:52	文本文档	62 KB
DOSBox 0.74 Options.bat	2010/4/10 3:10	Windows 批处理...	1 KB
DOSBox.exe	2010/5/12 17:55	应用程序	3,640 KB
Reset KeyMapper.bat	2010/5/9 4:17	Windows 批处理...	1 KB
Reset Options.bat	2010/5/9 4:17	Windows 批处理...	1 KB
Screenshots & Recordings.bat	2010/4/10 3:10	Windows 批处理...	1 KB
SDL.dll	2010/4/10 3:04	应用程序扩展	438 KB
SDL_net.dll	2010/4/10 3:04	应用程序扩展	13 KB
uninstall.exe	2019/10/16 17:11	应用程序	37 KB

4. 在弹出的配置文件中找到 [autoexec], 在其下根据个人实际情况填写MASM的解压目录, Ctrl + S保存之后退出

```
[autoexec]
```

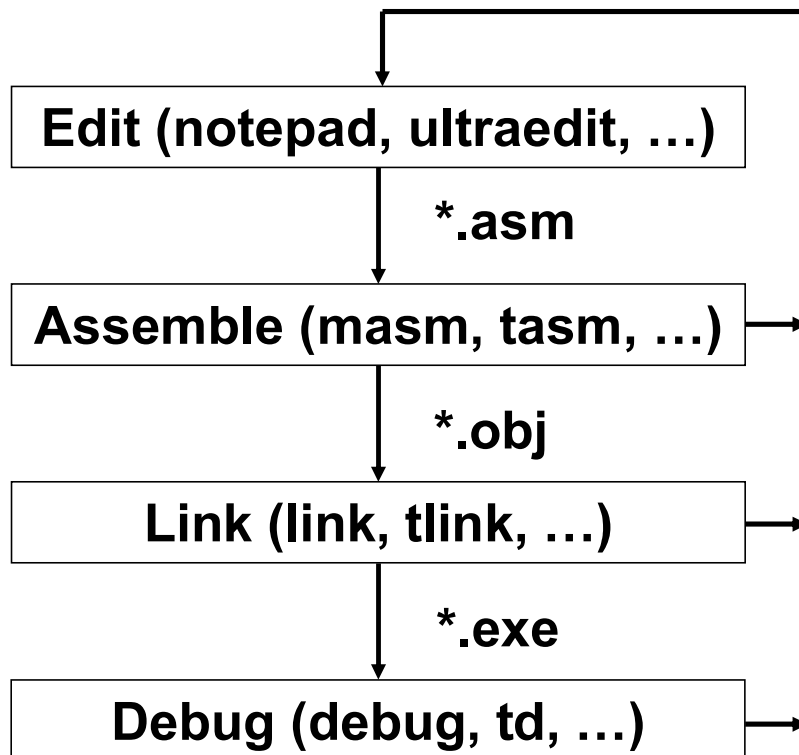
```
# Lines in this section will be run at startup.
```

```
# You can put your MOUNT lines here.
```

```
MOUNT C F:\ustc\experiment\masm+dosbox\masm5
```

MASMv5.0+DOSBOX使用(1)

- 程序设计流程



MASMv5.0+DOSBOX使用(2)

1. 使用Notepad++、VS Code等编写.asm文件，并将其放入之前配置好的masm文件夹, 这里命名为“hello.asm”



名称	修改日期	类型	大小
cmd.exe	2008/4/14 20:00	应用程序	460 KB
debug.exe	2015/9/27 13:51	应用程序	21 KB
edit.com	2008/4/14 20:00	MS-DOS 应用程序	69 KB
hello.asm	2019/10/16 20:23	ASM 文件	1 KB
HELLO.EXE	2019/10/16 20:23	应用程序	1 KB
HELLO.OBJ	2019/10/16 20:23	3D 对象	1 KB
LINK.EXE	1996/5/12 16:28	应用程序	39 KB
MAKE.EXE	1996/5/12 16:28	应用程序	24 KB
MASM.EXE	1996/5/12 16:28	应用程序	65 KB
test.asm	2008/12/12 10:00	ASM 文件	1 KB
TEST.OBJ	2019/10/16 20:19	3D 对象	1 KB

```
DATAS SEGMENT
    STRING DB 'Hello World!$'

DATAS ENDS

CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS

START:
    MOV AX,DATAS
    MOV DS,AX

    LEA DX,STRING

    MOV AH,9
    INT 21H

    MOV AX,4C00H
    INT 21H
CODES ENDS
END START
```

MASMv5.0+DOSBOX使用(3)

2. 启动DOSBOX

首先输入挂载的盘符号，这里是**C:**。

```
HAVE FUN!  
The DOSBox Team http://www.dosbox.com  
Z:\>SET BLASTER=A220 I7 D1 H5 T6  
Z:\>MOUNT C F:\ustc\experiment\masm+dosbox\masm5  
Drive C is mounted as local directory F:\ustc\experiment\masm+dosbox\masm5\  
Z:\>C:  
C:\>
```

3. 运行**MASM**进行编译

输入要编译的.asm文件名(不加后缀名)，按照提示完成编译

```
Z:\>MOUNT C F:\ustc\experiment\masm+dosbox\masm5  
Drive C is mounted as local directory F:\ustc\experiment\masm+dosbox\masm5\  
Z:\>C:  
C:\>masm  
Microsoft (R) Macro Assembler Version 5.00  
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.  
Source filename [I.ASM]: hello  
Object filename [hello.OBJ]:  
Source listing [NUL.LST]:  
Cross-reference [NUL.CRF]:  
  
51758 + 464786 Bytes symbol space free  
0 Warning Errors  
0 Severe Errors  
C:\>
```

MASMv5.0+DOSBOX使用(4)

4. 运行LINK进行链接

输入.obj文件名hello，按照提示完成链接

```
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Source filename [.ASM]: hello
Object filename [hello.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51750 + 464786 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Object Modules [.OBJ]: hello
Run File [HELLO.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
```

5. （可选）运行debug进行调试

6. 执行编译好的hello.exe

显示出 “Hello World!”

```
Object Modules [.OBJ]: hello
Run File [HELLO.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>hello
Hello World!
C:\>
```

基本功能调用：基本操作

- **INT 21H**中断命令（**DOS**功能调用）
 - 根据**AH**的值（功能号）来执行不同的中断命令

AH	功能	调用参数	返回参数	备注
00H	程序终止	CS=程序段前缀		同INT 20
4CH	带返回码的终止	AL=返回码		

- **用法**：先将功能号存入AH，再执行INT 21H
- **示例**：执行以下命令后，程序终止，返回码为0

```
MOV AX, 4C00H  
INT 21H
```

基本功能调用：基础I/O操作

- **INT 21H中断命令（DOS功能调用）**
 - 根据**AH**的值（功能号）来执行不同的中断命令

AH	功能	调用参数	返回参数	备注
01H	键盘输入并回显		AL=输入字符的ASCII码	
02H	屏幕输出字符	DL=输出字符的ASCII码		
07H	键盘输入无回显		AL=输入字符的ASCII码	
09H	屏幕输出字符串	DS:DX=串地址		字符串末尾应为 '\$'
0AH	键盘输入到缓冲区（即键盘输入字符串）	DS:DX=缓冲区地址	[DS:DX+1]=实际输入的字符数	[DS:DX]=缓冲区可容纳的最大字符数； [DS:DX+1]=实际输入字符数； 字符串本身从 [DS:DX+2]开始存储

基本功能调用：基础文件操作

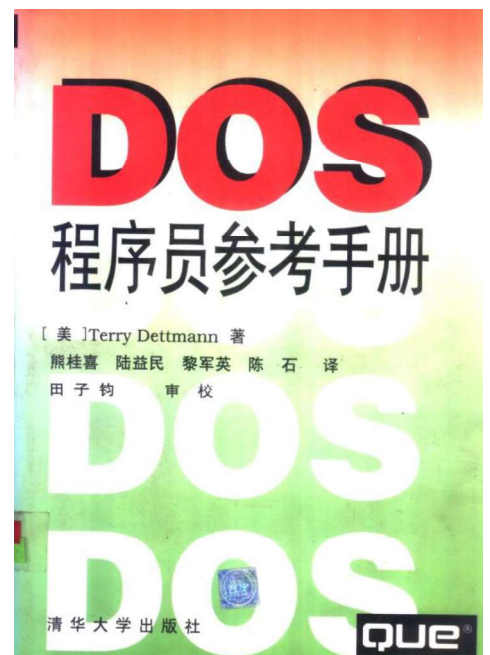
- **INT 21H中断命令（DOS功能调用）**
 - 根据**AH**的值（功能号）来执行不同的中断命令

AH	功能	调用参数	返回参数	备注
3CH	建立文件	DS:DX=文件名串地址; CX=文件类型	成功:AX=文件句柄; 失败:AX=错误码	CX=00H(普通) 02H(隐藏文件) 04H(系统文件)
3DH	打开文件	DS:DX=文件名串地址; AL=0(读)/1(写)/3(读/写)	成功:AX=文件句柄; 失败:AX=错误码	
3EH	关闭文件	BX=文件句柄	失败:AX=错误码	
3FH	读文件/设备	DS:DX=缓冲区地址; BX=文件句柄; CX=读取的字节数	成功:AX=实际读入的字节数; 失败:AX=错误码	若返回AX=0,即已读到文件尾
40H	写文件/设备	DS:DX=缓冲区地址; BX=文件句柄; CX=写入的字节数	成功:AX=实际写入的字节数; 失败:AX=错误码	
41H	删除文件	DS:DX=文件名串地址	成功:AX=00 失败:AX=错误码	

基本功能调用：参考资料

- 《DOS Programmer's Reference》

- 中译《DOS程序员参考手册》
- 图书馆有电子版可供借阅



- 课程主页有其他资料可供参考

- 详情访问：

<http://staff.ustc.edu.cn/~wjluo/mcps/>

EMU8086

GUI比较直观，不再赘述

