Ppt1：

用指令助记符和符号地址来编写程序

汇编程序讲源代码翻译成二进制编码

包括：指令助记符，数据和存放数据的变量，标号，伪指令，以及相应的使用规则

源程序中一条汇编指令或者微指令均占一行

Ppt2：

Ppt3：

讲的是汇编编辑连接操作

Ppt4：

讲的是连接和运行调试

Ppt6：

源程序的基本框架结构

汇编语言对内存都是分段使用的，一般包括，代码段（CS指向当前运行的代码），数据段（DS指向当前的数据段），堆栈段（SS指向当前的堆栈段）

但是少了堆栈段也没啥关系，因为操作系统可以分配

系统分配的：

CS

SS

SP

代码在上部，堆栈在下部

Ppt7：

一闪而过

Ppt8：

\_STACK SEGMENT STACK ‘STACK’

DB 32766 DUP(0)

TOS DW 0

\_STACK ENDS

\_DATA SEGMENT ‘DATA’

MSG DB ‘HELLO,WORLD!’, 13, 10, ‘$’

\_DATA ENDS

\_TEXT SEGMENT ‘CODE’

ASSUME CS: \_TEXT, DS: \_DATA, SS: \_STACK

START: MOV AX, \_DATA

MOV DS, AX

MOV AX, 4C00H

INT 21H

\_TEXT ENDS

END

Ppt9:

源程序总是以end作为结束标志

End后面可以附带一个在源程序中已经定义的符号，用以说明程序中的在调入时开始执行的第一条指令所在的位置。这个标号就是程序的执行入口

——源程序是主代码块，伪指令end后面必须有标号

——…………非………………………………………………不能…………

PPT10:

汇编语言程序两方面内容：

与数据相关的常量和变量

与转移地址相关的符号和过程

标号名，变量名，过程名，段名等称为标识符

组成标识符的字符，字母a~z，数字0~9，以及？。@$之类

Ppt11：

变量定义：

数据定义伪指令的格是是：

[变量名] 数据定义伪指令 初值表 [;注释]

分配一个或者多个指定类型的内存单元，并且可用变量表示该内存单元

如果有变量名那么它仅仅代表所定义的数据存储区的第一个单元地址

用 DB,DW,DD,DF,DQ,DT定义并且分配一个或者多个1B,2B,4B,6B,8B,10B类型的存储单元。

比如：

Msg DB ‘HELLO’ ;定义变量，分配5个1B空间，置初值

DB 13,10,’$’ ；分配3geB空间，置初值

Cnt DW 5\*20, -2 ;定义变量，分配两个2B空间，置初值

S1 DB ?,?,?,? ；定义变量，分配4个1B空间，未置初值

DD 123456H,? ；分配2个4B空间

定义变量，赋值的时候，可以用十进制10，可以用十六进制0Ah,（这个0不能省，否则编程ah这个标识符）

可以用二进制00001010b来写

Ppt12：

可以用“？”形式

——分配一个或者多个存储单元，但是内容不确定

——一般情况下，程序在汇编时以0填充

符号地址和地址表达式：

——符号地址对应的地址编号

——是无符号整型常数

Msg DB ‘HELLO’, 13, 10, ‘$’

P1 DW Msg，Msg+5

可以用DUP把某项重复n次，格是为：

N DUP（数据项）

S1 DB “你好“，2 DUP(‘!’), 2 DUP(‘A’,’B’), 3 DUP(1,2,2 DUP(‘$’))

Ppt13：

变量的使用

对于内存操作数，除了寄存器间接寻址方式不适用变量名之外，其余的各种寻址方式均可以用变量名来表示

用变量名表示的内存操作数寻址为：

直接寻址 MOV AX VAR

寄存器寻址 MOV AX VAR [BX]

相对基址变址寻址： MOV AX VAR [BX][DI]

Ppt14：

地址属性

Ppt16：

使用万丈的段定义伪指令来制定一个段，可以具体控制汇编程序MASM和连接程序link在内存中组织代码和数据的方式

格是就是

[类型] Segment [定位类型][组合方式][地址模式][‘分类名’]

DATA SEGMENT USE16(在dos下)/USE32(vs下调试) PUBLIC

1. 段指定伪指令 ASSUME

ASSUME 段寄存器名字：段名[段寄存器名，……]

1. 结束用end
2. 符号定义指令

等值伪指令EQU

格式：名字EQU符号串

CONST EQU 50 ;定义常数

VAR EQU WORD PTR X

LAB EQU START

M EQU MOV

等于号伪指令=

‘5. 地址计数器与对准伪指令

ARRAY DW 1,2,$,3,4,$

如果array偏移地址为0074，那么最后存储区内容为：

01 00 02 00 78 00 03 00 04 00 7E 00