实验1

汇编语言程序格式

实验1 实验环境及上机过程

教学重点

汇编语言源程序的建立、汇编、连接、调试及运行

主要内容

I、实验环境

II、汇编语言程序的上机过程

实验环境

计算机语言实验环境就是使用该计算机语言所需要的硬件环境和软件环境

具体来说，就是使用者用到的机型、操作系统，所需要的语言系统软件，以及确定具体编程时源程序、目标程序、可执行程序所存放的路径和文件夹

汇编语言系统软件

汇编语言编译器(MASM.EXE，ML.EXE，CV.EXE等)：作用是将汇编语言源程序（.ASM文件）翻译为目标代码程序（.OBJ文件）

连接器（LINK.EXE)：作用是连接目标代码程序和库函数生成可执行程序文件（.EXE文件）

可执行程序动态调试器（DEBUG.EXE，TR.EXE等）：作用是对可执行程序进行装载情况的静态了解和动态执行调试

将汇编语言系统软件集中在一个文件夹中，E:\MASM

操作系统

汇编语言系统软件需要在PC系列微型计算机的DOS操作系统下运行

PC系列微型计算机都具备该运行环境

Windows操作系统上的“命令提示符”窗口提供了模拟的DOS操作系统环境

进入“命令提示符”窗口方式：

- 执行“开始”→“程序”→“附件”→“命令提示符”命令打开“命令提示符”窗口
- 执行“开始”→“运行”命令打开“运行”对话框→输入“cmd”→“确定”

文件的路径和文件夹

将汇编语言系统软件集中在一个文件夹中，E:\MASM

将源程序、目标程序、可执行程序集中在一个文件夹中，E:\MASM\Xingming

进入E:\MASM文件夹：在“命令提示符”窗口中输入

E：回车　cd MASM 并回车

列出当前目录中的所有文件：输入dir命令并回车

在E:\MASM\Xingming文件夹中汇编源文件：

E:\MASM\MASM EX1.ASM

主要内容

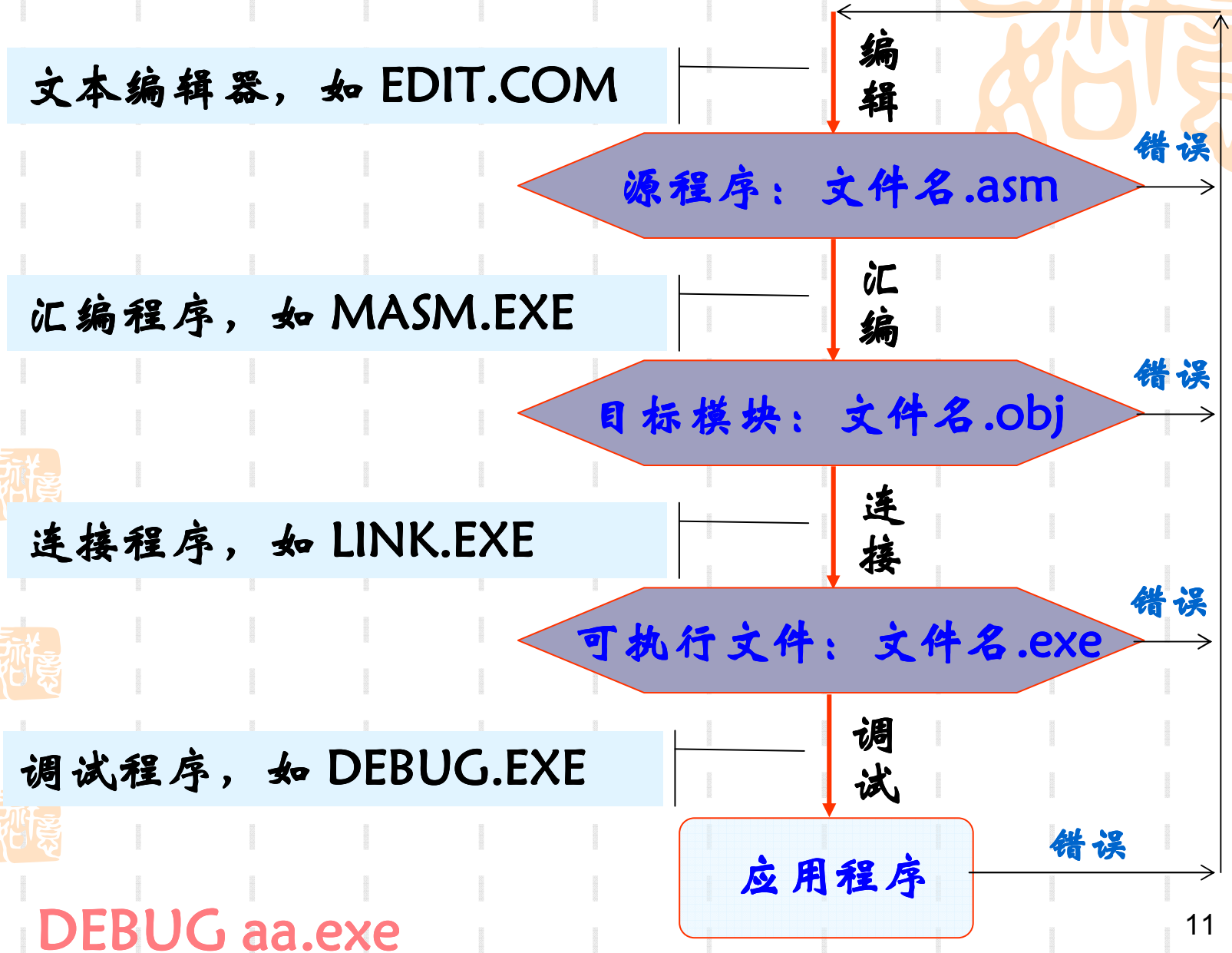
I、实验环境

II、汇编语言程序的上机过程

汇编语言的程序格式

- 完整的汇编语言源程序由段组成
- 一个汇编语言源程序可以包含若干个代码段、数据段、附加段或堆栈段，段与段之间的顺序可随意排列
- 需独立运行的程序必须包含一个代码段，并指示程序执行的起始点，一个程序只有一个起始点
- 所有的可执行性语句必须位于某一个代码段内，说明性语句可根据需要位于任一段内
- 通常，程序还需要一个堆栈段

汇编语言程序的建立及汇编过程



DEBUG aa.exe

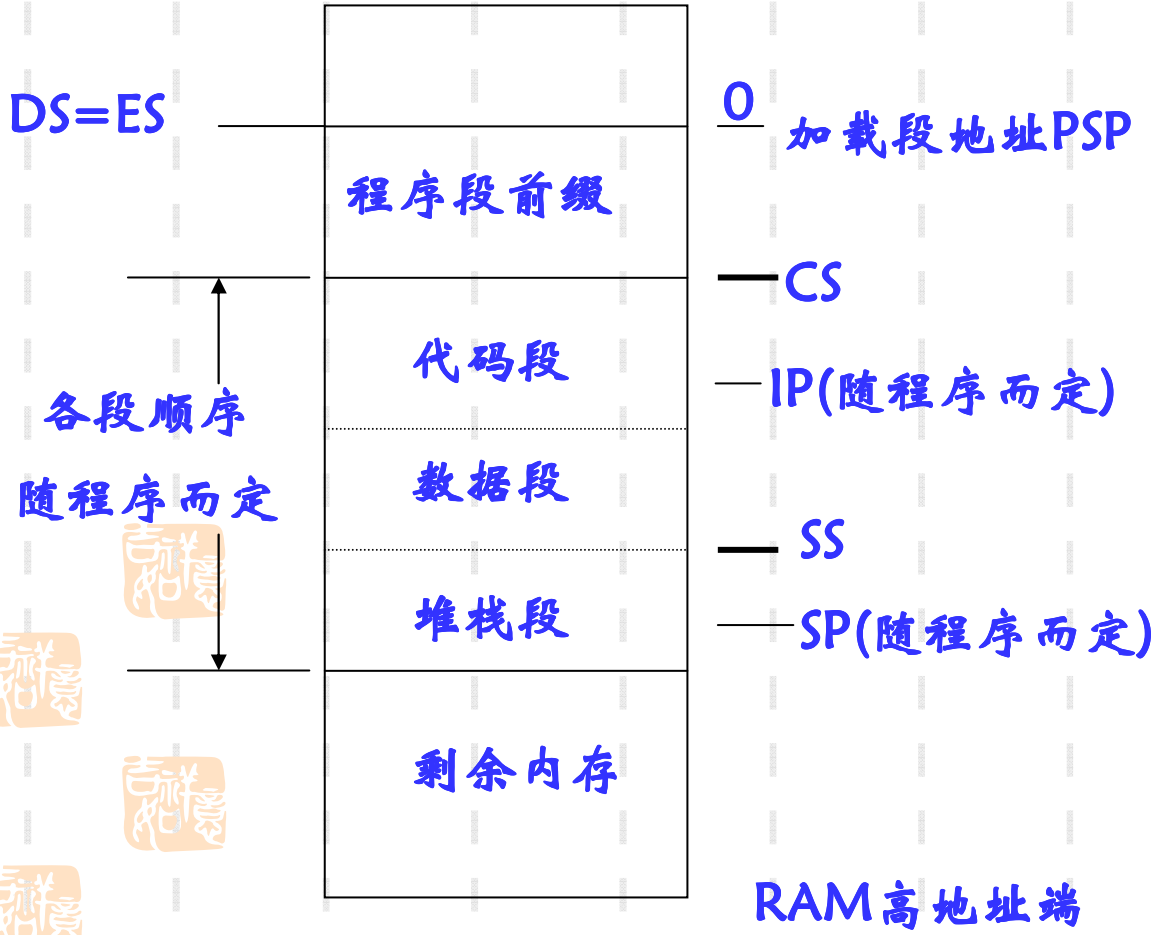
汇编程序的主要功能

- 检查源程序
- 测出源程序中的语法错误，并给出出错信息
- 产生源程序的目标程序，并可给出列表（同时列出汇编语言和机器语言的文件，称为LST文件）
- 展开宏指令

exe程序

- 利用程序开发工具，通常将生成EXE结构的可执行程序（扩展名为.EXE的文件）
- 它可以有独立的代码、数据和堆栈段，还可以有多个代码段或多个数据段，程序长度可以超过64KB，执行起始处可以任意指定
- 当DOS装入或执行一个程序时，DOS确定当时主存最低的可用地址作为该程序的装入起始点。此点以下的区域称为程序段。在程序段内偏移0处，DOS为该程序建立一个程序段前缀控制块PSP（Program Segment Prefix），它占256（=100h）个字节；而在偏移100h处才装入程序本身

exe程序的内存映象图




```
Push  ds
Sub ax,ax
Push ax
Mov ax,data
Mov ds,ax
Mov ax,extra
Mov es,ax


……
ret
```

上机过程

1.建立源程序：利用Windows记事本或EDIT文本编辑器输入汇编语言源程序，保存为扩展名为.ASM的文件-----aa.asm

2. 汇编：进入“命令提示符”窗口，并进入E:\MASM，输入“MASM aa；”并回车------产生aa.obj文件

3.连接：在“命令提示符”窗口输入“LINK aa；”并回车------产生aa.exe文件

4.查看文件清单：输入dir命令并回车

5.利用Debug程序调试与运行可执行程序：输入“debug aa.exe”

调试程序DEBUG

- DEBUG是常用的汇编语言级调试工具，为汇编语言程序员提供了分析指令、跟踪程序的有效手段
- 常用命令：
  - A 输入小汇编程序
  - U 反汇编
  - T 单步执行程序
  - G 执行程序
  - D 显示内存单元
  - R 显示并修改寄存器内容
  - E 修改内存单元内容
  - P 步跟踪程序

一定要采用调试程序DEBUG进行实践

反汇编命令U

格式：（1）U <地址> ↓

（2）U<地址范围> ↓

将指定地址范围内的代码以汇编语句形式显示，同时显示地址及代码.

地址及范围的缺省值是上次U命令后下一地址的值，这样可以连续反汇编.

检查和修改寄存器命令R

格式：（1）R↓ 显示所有寄存器的内容

（2）R<寄存器名>↓ 可进入寄存器修改状态

（3）RF↓ 可显示和修改标志寄存器的状态

标志名		标志为1	标志为0
OF	溢出（是/否）	OV	NV
DF	方向（减量/增量）	DN	UP
IF	中断（允许/关闭）	EI	DI
SF	符号（负/正）	NG	PL
ZF	0（是/否）	ZR	NZ
AF	辅助进位（是/否）	AC	NA
PF	奇偶（偶/奇）	PE	PO
CF	进位（是/否）	CY	NC

运行命令G

格式：（1）G↓

（2）G=<地址>↓

（3）G=<地址><断点>↓

(1)从CS:IP开始执行

(2)从指定地址开始执行

(3)从指定地址开始执行,到断点自动停止

跟踪命令T

格式：　(1) T↓

(2) T=<地址> ↓

(3) T=<地址 条数> ↓

执行指定的一条指令，并显示执行一条指令后的所有寄存器及标志位的状态

(1)从CS:IP开始执行　　(2)从指定地址开始执行

(3)执行从指定地址开始的由条数指定的若干条指令。

步跟踪命令P

格式：P↓

同T命令，但是遇到LOOP类循环指令、INT n类软中断调用指令、CALL类子程序调用指令时不再单步执行，而是连续执行完整个循环程序或者相应的中断程序或子程序，返回LOOP、INT n和CALL等指令的下一条指令

显示内存命令D

格式：（1）D<地址>↓

（2）D<范围>↓

–D100↓显示以DS内容为段地址，以100H为偏移地址的80H个内存单元的内容.

–D100 200↓显示以DS内容为段地址，以100H为偏移地址至200H间的内存单元的内容.

可用段前缀操作.

修改内存命令E

格式：（1）E＜地址＞ ＜内容表＞↓

－E DS:100 F3’XYZ’80↓将内存单元DS：100H至DS:104H共5个单元的内容分别修改为F3H、’X’、’Y’、’Z’和80H.

（2）E＜地址＞↓

执行后显示起始地址为DS:地址单元的内容，此时可输入新值，按空格移到下一单元，或按‘－’回到上一单元修改，修改结束后按回车键.

输入小汇编程序命令A

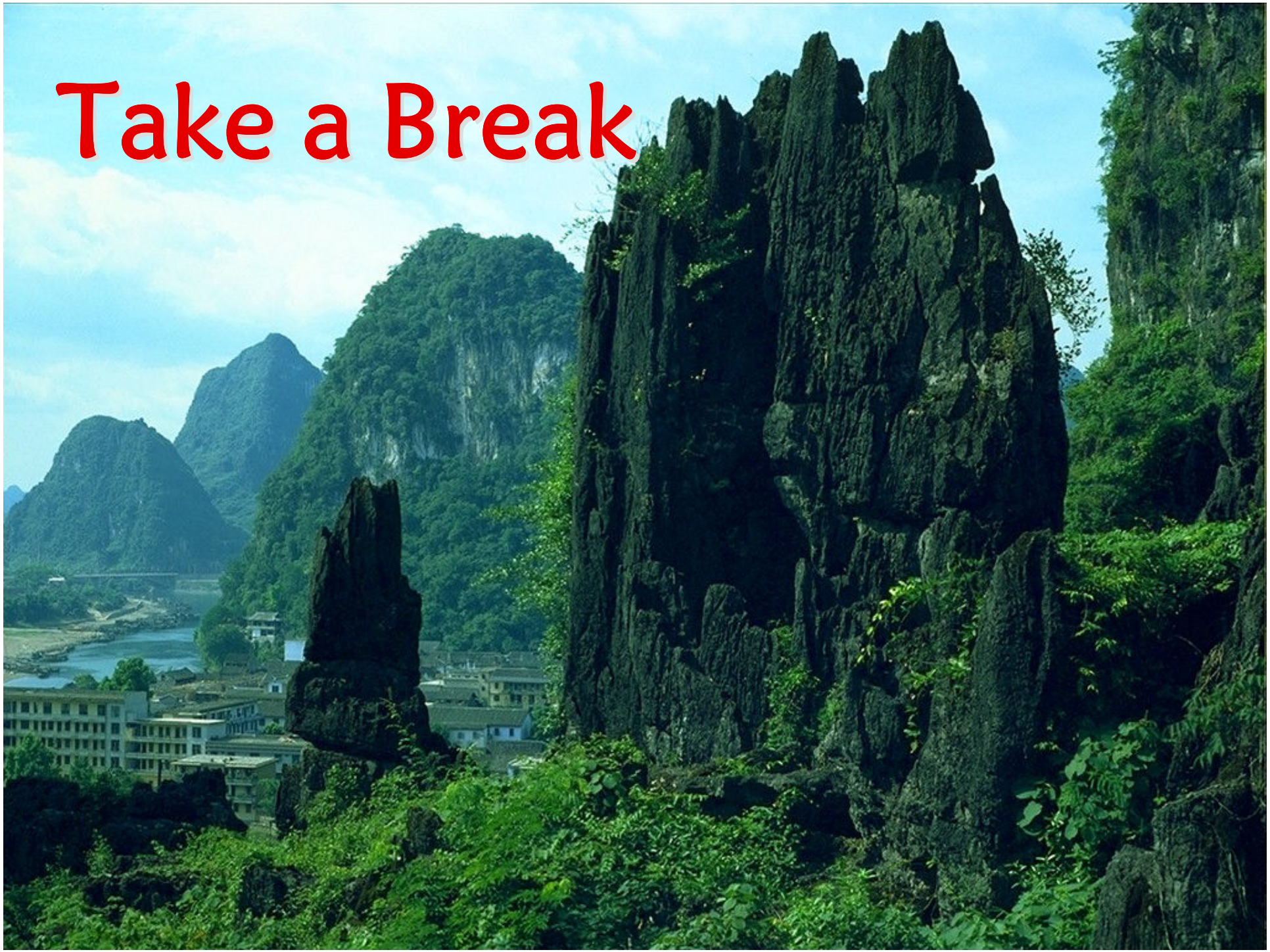
格式： A＜地址＞

从指定地址开始，输入汇编语言语句，由A命令把它们汇编为机器代码，并从指定地址单元开始连续存放

>debug 回车　-A 回车

输入完后，可以用T命令运行程序

A命令的优点是可以对部分程序段进行汇编，然后进行运行和调试，这比放在整个程序中调试要简便得多


Take a Break