微机原理及接口技术

Hardware Principles and Interfacing of Modern Computer

Lecture 7: Assembly Engineering

陈启军，张 伟

Email: zhang_wi@mail.tongji.edu.cn

Dept. Of Control Science and Engineering, TongJi University

Content

- 汇编语言的学习方法
- 示例程序剖析和编程技巧学习
- 汇编程序开发的实验环境设置

Reference

- The Intel Microprocessors: 8086/8088…: Architecture, Programming, and Interfacing, [美]Barry B. Brey（巴里 B.布雷）著, 机械工业出版社，2005，ISBN 7-111-16052-5
- 沈美明， 温冬婵, 80X86汇编语言程序设计,清华大学出版社, 2001.09, ISBN 7-302-04540-2
- IBM PC Assembler Language Programming

How to Learning Well ?

- 多看——看别人写的范例程序，并且进行深入的分析，体会其编程技巧
- 多练——根据实际中的功能要求，进行实战模拟
- 在体悟原理的同时，要注重技能的锻炼
  - 注重大规模程序的开发和质量掌控
    - 单个项目 > 10000行，本科4年累计应 > 3万行
  - 注重常见编程技巧的掌握：例如快速查找和排序，查表法，跳转表法
  - 注意有效编程模式（编程套路）的总结，如状态机模式
  - 调试 (debugging) 技巧
  - 版本管理技巧
- 在实践中特别是走弯路的过程中总结经验，把每次犯错误都当作学习的机会

尽管在原理学习阶段，每个人领悟的快慢深浅会受到天资的影响，但技能的培训和锻炼如果不花下去足够的时间，是不会有层次上的提高的。

First Full Sample Program For DOS

- **Section 4.7 Assembler Detail**
  - Example 4-20, P. 143(Sixth Ed.)

First Full Sample Program For DOS

- 要点：尽管一个汇编程序在本质上就是instruction的序列，但是，一个可实际应用的汇编程序在开发时必须充分考虑其运行环境已经提供的功能和约束，这里主要是硬件BIOS和操作系统提供的功能和约束，例如
  - 我们开发的实验程序一般都可认为是操作系统软件调用的一个子程序，所以我们的入口过程经常需要声明为FAR
  - DOS汇编程序退出时不宜用halt指令，而应该调用4CH号INT21中断退出程序并返回DOS
  - 在需要读取键盘输入和向显示器输出时，可以调用BIOS中断程序和DOS中断程序实现

First Full Sample Program For DOS

- 说明：
  - BIOS中断和DOS中断的概念：可被认为是由BIOS固件和DOS操作系统提供的一系列功能子函数的集合，可在程序中使用。这与中断服务子程序ISR是不同的，只是借用了“中断”这个名次而已
  - DOS缺省堆栈段：100 byte。为避免以后函数调用时不必要的堆栈溢出可能，简易显式定义堆栈段并用dup(?)开辟足够的空间
  - DOS中断06H: 读取键盘
  - DOS中断4CH: 退出到DOS提示符状态

First Full Sample Program For DOS

- 说明：简化的语法格式，采用汇编器的指示符号**(assembler directives)**书写
  - Text Book Section 4.7
    - Example 4-21 (P.145, Sixth Ed.)

First Full Sample Program For DOS

- 思考：神秘的BIOS中断和DOS中断看来功能强大，它们是如何实现的？我们能否自己开发之？

  - Note: The instruction set is **self-contained**, which enable us to do anything with assembly language, Including developing BIOS routines, DOS routine, or even virus.
  - Recruitment in future: BIOS Engineer, Device Driver Software Developers, …

Sample Two

- **Example 5.37**
- **Tips**
  - 如何从一个字符串中搜索指定字符：利用CMPSB指令
  - 学习跳转指令的使用
  - 学习如何通过读写显示缓冲区操纵文本屏幕显示
    - Video display buffer begins at address B800:000.

    It's the second method to manipulate monitor display in addition to DOS INT 21H calls.

Sample Three

- Example 6-21 (P.215)
- Tips
  - 利用堆栈来传递函数参数和返回值

Sample Four

- **Example 6-22 (P 217)**
- **Tips:**
  - 工具函数：向monitor输出一个以NULL结束的字符串

```
; a procedure that displays the character string
; addressed by SI in the data segment. the string must
; end with NULL
; this procedure changes AX, DX, SI
STRING PROC FAR
   LODSB                ; get string character
   CMP AL, 0            ; test for null
   JE STRING1           ; if null
   MOV DL, AL           ; move ascii code to DL
   MOV AH, 2            ; select function 02h
   INT 21H              ; access dos
   JMP STRING           ; repeat until null
STRING1:
   RET                  ; return from procedure
```

Sample Five

- **Example 7.5**
- **Tips**
  - Public修饰符和FAR修饰符
  - 利用LIB实用程序创建库(library)。创建库是开发大工程和复用以前代码的良好手段之一。

Sample Six

- **Example 7.15-7.17**
- **Tips:**
  - 几种在程序中读取键盘的方法

Sample Seven

- **Example 7.18-7.22**
- **Tips:**
  - 使用DOS中断程序向显示器输出信息
  - 使用BIOS中断程序向显示器输出信息并且控制光标位置

    注：以上仅限于文本方式(text mode)，图形方式(graphics mode)下完全不同。

Sample Eight

- **Example 7.27-7.34, 7.45, 7.46**
- **Tips:**
  - Converting from binary to ascii
  - Converting from ascii to binary
  - Displaying and reading hexadecimal
  - Using lookup table for data conversions

Sample Nine

- **Example 7.48-7.51**
- **Tips:**
  - Interrupt Vector的更改
  - Interrupt Hook技术的思路
  - TSR (Terminated and stay resident) Program
  - DOS下Hotkey的实现

Sample Ten

- **Example 8.1-8.9（P300)**
- **Tips:**
  - 在C语言中嵌入汇编的开发模式，特别是32bit编程环境下
  - 在C语言中用汇编直接操纵I/O端口

Sample Eleven

- **Example 8.10 – 8.15**
- **Tips:**
  - 使用独立汇编模块嵌入C项目的开发模式
  - 在C程序中使用新增CPU指令

ASM Environment Installation

- 方法1：多重引导方案
  - 在计算机上装一套MS DOS或者支持DOS的Win95 / 98系统，并配合Borland的TASM或Microsoft的MASM
    - 我们的实验室环境即是采用该方案
    - 该方案可以实现对硬件的完全操纵，功能上最为完备

ASM Environment Installation

- 方案2：虚拟机方案
  - 在Windows2000/XP或者Linux操作系统中，安装VMware等虚拟机软件，然后在该虚拟机上安装MSDOS / Win95 / Win98 操作系统，并配合TASM 或者MASM进行学习和实验
    - 虚拟机的硬件是仿真出来的，某些涉及硬件读写的程序可能会有问题，但绝大部分都可用

ASM Environment Installation

- **方案3：CPU仿真**
  - 安装仿真软件 emu8086
    - 遵照该软件的要求编写汇编程序进行实验
    - 可查阅该软件安装目录下的sample
    - 操作界面直观，非常适于学习，但程序功能受限制
- **方案4：直接在Win2000 / XP系统下使用assembler和linker学习汇编**
  - 出于protected mode的原因，很多操作特别是涉及直接操作I/O的指令很可能无法执行，会被系统认为是非法操作。

ASM Environment Installation

- 方案5：独立的汇编语言模块
  - 选取Borland C++ Builder / Microsoft Visual Studio等开发工具开发主程序，然后与汇编模块链接进行实验
    - 该方法在工程实践中应用很广，值得掌握
    - 汇编模块的开发要受到一些汇编语法上的限制

ASM Environment Installation

- 方案6：嵌入式汇编
  - 选取Borland C++ Builder / Microsoft Visual Studio等开发工具开发主程序，然后在其函数实现中嵌入汇编代码进行实验
    - 工程中经常采用，值得掌握
    - 使用方便，但汇编语句受到的限制更多