《汇编语言程序设计》教学大纲

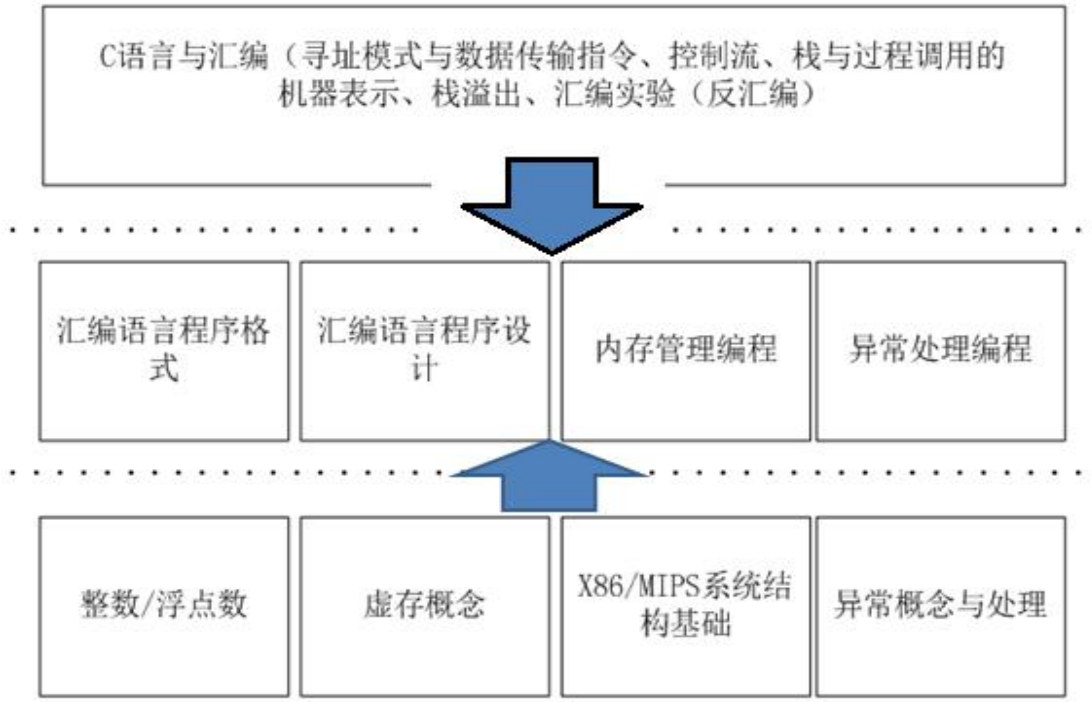
一、教学目的与要求

汇编课主要介绍汇编语言的基础理论、编程工具、编程方法和应用技术。通过课程的学习，使学生掌握利用汇编语言进行程序设计的方法和技巧，获得在计算机底层编程的经验，同时了解程序在机器上运行的基本原理，建立“时间”和“空间”的概念，关注影响程序运行的系统特征，使学生能应用所学的理论和技术编写出高效、可靠的程序，具备一定的软件设计与分析能力。

二、教学任务及其结构

教学任务是：通过课堂教学、自习和上机实验等环节，要求学生了解计算机的编程结构，掌握汇编语言程序设计的基本概念、方法和技巧，学会上机调试、运行程序的基本方法，具备一定的软件设计的能力。

主体的内容结构如下图。即以整数/浮点数/体系结构基础知识/虚存、异常概念为基本，从 C 语言与汇编的对应关系入手，讲授汇编语言自身的格式、语法、程序设计等内容。



三、单元教学目标与任务

第一章　汇编语言基础知识　　6 学时

1．1 汇编语言与计算机系统结构　　1.5 学时

1．2 各类指令集简介　　1.5 学时

1．3 整数的计算机表示与运算　　1.5 学时

1．4 浮点数的计算机表示　　1.5 学时

第二章　80x86 计算机组织　　2 学时

2．1 计算机系统构成与保护模式

第三章　C 语言与汇编　　12.5 学时

3．1 寻址模式与数据传输指令等　　3 学时

3．2 控制流 3 学时
3．3 栈与过程调用的机器表示 3 学时
3．4 汇编实验 2 学时
3．5 数据的机器表示 1.5 学时
第四章 汇编语言程序格式 6 学时
4．1 程序格式与伪操作等 3 学时
4．2 上机过程等 3 学时
第五章 循环与分支程序 6 学时
5．1 循环程序设计 3 学时
5．2 分支程序设计 3 学时
第六章 子程序设计 6 学时
6．1 子程序设计-1 3 学时
6．2 子程序设计-2 3 学时
第七章 MIPS 汇编 7.5 学时
7．1 MIPS 汇编初步 2 学时
7．2 MIPS 指令集与汇编程序设计 2 学时
7．3 MIPS 异常处理 1.5 学时
7．4 虚存管理初步 1 学时
7．5 MIPS 内存管理 1 学时
第八章 补充内容 2 学时
8．1 可定制处理器指令集

四、实验内容

（1）用递归子程序计算 Fibonacci 数。

要求：1. 用汇编和 C（或 C++）两种语言实现

2. 以此为例分析：

- 堆栈的使用情况
- 两种语言的优缺点（空间和时间）

3. 此实验要求提交实验报告

（2）建立一个宏（数学）库，扩展已有的指令系统。

要求：

1. 宏库包含 **n 的阶乘**、**n 的开方**、**n 的平方**、**n 的绝对值**、**以 2 为底 n 的对数**、**以 10 为底 n 的对数**、**2 的 n 次幂**、**10 的 n 次幂**的宏定义，运算结果仅取整数部分，不考虑溢出（字长 32 位），但要考虑 n 的正负；
2. 采用 EAX 等 32 位寄存器；
3. 尽量使用条件或重复等高级汇编技术；
4. 代码段中要有相应的宏调用，以检验宏定义的正确性；
5. 适当的输入输出提示；
6. 此实验要求提交实验报告

（3）软件炸弹拆除实验

要求：

1. 通过远程服务器登录后下载每个人的实验包，内含多个 X86 32 位目标程序；
2. 通过反汇编以及 gdb 等工具，进行目标程序语义分析与试运行，了解其含义并给出正确的输入以拆除“炸弹”。
3. 上传实验结果。

（4） 栈溢出攻击实验

要求：

1. 通过远程服务器登录后下载每个人的实验包，内含多个 X86 32 位目标程序；
2. 通过反汇编以及 gdb 等工具，进行目标程序语义分析与试运行，实施栈溢出操作，并完成预定的程序行为；
3. 上传实验结果。

（5） MIPS 异常处理实验

要求：

1. 在 SPIM 模拟器上，通过汇编程序产生地址错误异常；
2. 编写异常处理程序，要求正确处理该异常并返回（打出相关异常信息）；
3. 能够支持发生在 delay slot 内的异常。

五、重点难点与解决办法

学生遇到的难点主要分两种：一是基本概念方面的，对这些概念和基础知识正确而清晰的理解非常重要；二是应用和编程方面的（主要体现在程序格式或结构上），而基本概念和编程技术又是本课程的重点，所以必须帮助学生解决这些难点。课程的重点和难点主要在以下几个方面：

- 80x86 计算机组织

这儿我们开始提到寄存器、内存单元的地址和内容、存储器的逻辑分段、物理地址、逻辑地址（段地址和偏移地址）等概念，学生往往不理解什么是“段”？为什么要“分段”？“物理”和“逻辑”有什么区别？“实模式”与“保护模式”是怎么回事？等等。

此时课程刚刚开始，学生还没有接触编程，没有感性认识，我们通过类比、图示、举例等通俗形式尽可能讲清楚这些问题，比如：8086/8088 之所以只能工作在“实模式”，是因为它的地址线是 20 根，而一个寄存器只有 16 位长，容纳不下 20 位的物理地址，必须使用两个寄存器；这两个寄存器一个放段地址，一个放偏移地址（二者构成逻辑地址），此时顺便讲解每个寄存器的专门用途，给出实模式下物理地址的计算公式；同时结合存储器中存放数据的不同，引出“逻辑分段”的概念。处理器发展到 80386 之后，寄存器达到 32 位，已经能够放下 32 位物理地址。但对于新增的保护工作模式而言，并不是简单地把物理地址放到寄存器中，此时存储器中的代码和数据为适应多任务的需求包含各自的“保护”信息，这些信息存放在“描述符”中，而“描述符”又构成“描述符表”保存在内存的某块区域。此时的逻辑地址由 16 位选择器（原来的段寄存器充当）和 32 位寄存器构成，代码和数据共用 4GB 的内存空间，已经不再有“段”的概念。

- 循环、分支与子程序结构的程序设计

从这儿开始涉及编程方法与技巧，尽管所举例子可能在高级语言中也出现过，学生并不

陌生，但学生的困难在于对这种表现形式的不习惯，特别是转移指令、循环指令的用法。我们会告诉学生，高级语言的 for while if else 翻译过来也是一大堆这样的形式（在“C 语言与汇编”中详述），我们直接用汇编这样写，会比高级语言效率更高；另外对于处理一些特定的问题如地址表、逻辑尺，汇编有其独特的优势。我们也会涉及一些经典算法如冒泡排序、折半查找，重点让学生感受汇编与高级语言在表现形式上的相同与不同。

- 栈与过程调用的机器表示

用栈传递参数或参数地址是学生比较难掌握的，尤其是 EBP 寄存器的使用和带立即数的 RET 指令的使用。更进一步地，嵌套和递归子程序设计对学生而言也有一定难度。我们在平时练习和上机中加重了这方面的训练，在实验中也特别设计了这方面的内容，并要求学生在实验报告中分析子程序功能、参数传递机制、栈变化情况等。

- 异常处理程序设计

通过 SPIM 模拟器的实际单步运行，一步步的演示 MIPS 下异常处理的流程与相关寄存器的变化，给出直观的异常处理概念。

- MIPS 内存管理

学生基本上对于系统虚存没有概念。因此必须首先通过通俗的原理讲解与虚存访问流程示例来赋予基本概念，然后才能针对特定的 TLB 处理流程、相关指令进行详细介绍。