

《汇编语言程序设计》教学大纲

课程英文名称: Assembly Language programming

课程代码: G2231330

学 时 数 : 32

学 分 数: 2

课程类型: 专业选修课程

适用学科专业: 信息安全

先修课程: 数字逻辑设计及应用, 计算机导论

执 笔 者: 赵洋

编写日期: 2018.09

审 核 人: 汤羽

一、课程简介 (Course Introduction)

《汇编语言程序设计》是一门软件工程专业基础课。汇编语言是高级语言和计算机系统间的桥梁, 其特点是把“程序”与“机器”相结合, 在了解计算机结构的基础上学习程序设计, 学习汇编语言编程对学生理解影响程序运行的机器特性, 并建立起完整的系统概念有着重要的作用。

Assembly Language Programming is a basic course for software engineering. Assembly language is a bridge between advanced language and computer system. Its characteristic is to combine "program" with "machine", learn program design on the basis of understanding computer structure, and learn assembly language programming plays an important role in students' understanding of the machine characteristics affecting program operation and establishing a complete system concept.

二、课程目标 (Course Objectives)

汇编语言程序设计是计算机及相关专业的核心专业基础课程, 它不仅是计算机组成原理、操作系统、微机原理与接口等核心课程的必要先修课, 而且对于训练学生掌握程序设计基本方法, 熟悉上机操作和程序调试技术有着重要作用。

通过本课程的学习, 使学生熟悉微型计算机指令系统和寻址方式; 掌握汇编语言程序的基本格式; 具有编写顺序程序、分支程序、循环程序和子程序等基本结构程序的能力; 掌握常用的应用程序的设计方法。

Assembly language programming is a core professional basic course for computer and related specialties. It is not only a necessary prerequisite course for core courses such as computer composition principle, operating system, computer principle and interface, but also plays an important role in training students to master the basic methods of programming and to be familiar with computer operation and program debugging technology.

Through the study of this course, students will be familiar with the instruction system and addressing mode of micro-computer; master the basic format of assembly language program; have the ability to write basic structural programs such as sequential program, branch program, loop program and subroutine; and master the design methods of commonly used application programs.

课程采用工程教育的基于学习成果教育模式（Outcomes Based Education, OBE）实施教学，其课程目标、课程模块、培养要求之间的关系见表 1 所示。

表 1 课程目标、课程模块、专业培养要求之间关系

课程目标 (CO)	CO1: 要求学生掌握利用汇编指令正确编写汇编程序的方法；培养学生优选汇编指令对一个算法提高实现高速性能的能力。
课程模块 (CM)	<p>CM1: 基础知识 对硬件系统结构的问题进行一部分的探讨，以使后续的课程可在一个好的基础上进行。本章是全课和计算机知识的基础，要求通过本章学习，学生对以下方面的知识有较全面的认识。</p> <p>CM2: 寄存器(CPU 工作原理) 对一个汇编程序员来说，CPU 中的主要部件是寄存器。寄存器是 CPU 中程序员可以用指令读写的部件。程序员通过改变各种寄存器中的内容实现对 CPU 的控制，因此要求通过本章学习了解寄存器基础知识。</p> <p>CM3: 寄存器（内存访问） 从访问内存的角度继续学习几个寄存器。因此要求通过本章学习使用寄存器访问内存和建立内存分段使用的概念</p> <p>CM4: 第 1 个程序 开始编写完整的汇编语言程序，用编译器将它们编译成为可执行文件（如：*.exe 文件），在操作系统中运行。</p> <p>CM5: [bx]和 loop 指令 介绍是使用 BX 寄存器和 LOOP 指令，编写循环程序，并引入标号和段前缀的使用。</p>

本课程满足毕业要求指标点的达成度评价方式见表 3 所示。

表 3 毕业要求指标点达成度评价

指标点	*权重 系数	考核方式					考核模块
		考试	测验	作业	实验	其他	
达成度 评价方式							

三、课程内容和安排和要求

（一）教学内容、要求及教学方法

本课程共 32 学时，课堂讲授 24 学时，上机实践 8 学时。课程采用“案例+翻转课堂教学”方式实施课堂教学。课程内容如下章节组成：

第 1 章 基础知识（CM1）2 学时

教学内容：机器语言、汇编语言的产生、汇编语言的组成、存储器、指令和数据、存储单元、CPU 对存储器的读写、地址总线、数据总线、控制总线、内存地址空间（概述）、主板、接口卡、各类存储器芯片、内存地址空间等汇编语言的一般概念。

重点：机器语言、汇编语言的产生、汇编语言的组成、存储器、指令和数据、存储单元、CPU 对存储器的读写、地址总线、数据总线、控制总线、内存地址空间（概述）、主板、接口卡、各类存储器芯片、内存地址空间等汇编语言的一般概念。

难点：对内存地址空间的正确理解。

教学要求：学习和掌握使用汇编语言的目的；汇编语言的组成，存储器构成，指令和数据，CPU 对存储器的读写，总线构成和内存地址空间。

教学方法：采用翻转课堂教学+案例方法，在课堂内，针对重点及难点进行探讨与讲解。首先引出学习汇编语言的目的问题，分析为什么需要掌握汇编语言。同时提出一些需要使用汇编语言解决的问题，启发学生进一步思考。在讲汇编基础知识过程中，同时还采用案例教学方法，以计算机存储器构成为例，使学生能直观理解内存地址空间的概念。

第2章 寄存器(CPU工作原理) (CM2) 2学时

教学内容：通用寄存器，字在寄存器中的存储，几条汇编指令，物理地址，16位结构的CPU，8086CPU给出物理地址的方法，“段地址 $\times 16$ +偏移地址=物理地址”的本质含义，段的概念，段寄存器CS和IP，代码段。

重点：8086CPU物理地址的表示方法，段的概念和CS、IP寄存器的工作原理。

难点：逻辑地址和物理地址的转换，CS、IP寄存器的修改方法。

教学要求：学习和掌握与汇编语言学习相关的基础硬件知识。理解8086CPU中对内存地址空间的逻辑和物理表达方式，CPU在指令执行过程中的基本原理，以及使用指令改变CS、IP寄存器内容的基本方法和使用要求。

教学方法：采用翻转课堂教学+案例方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。采用案例教学方式，以现实中位置导航为例，分析8086CPU采用逻辑地址表达物理地址的原理和原因。通过课后练习让学生验证debug环境下mov指令修改CS的执行结果，总结mov指令使用方式，并加深对CS寄存器使用的特殊性理解。

实践能力培养：本章教学内容开始接触汇编语言基本指令的使用，通过汇编环境搭建、验证指令使用方式等过程，可以使学生对汇编的开发环境建立初步的了解，拓展学生底层编程能力。

第3章 寄存器(内存访问) (CM3) 3学时

教学内容：内存中字的存储，DS和[address]，字的传送，mov、add、sub指令，数据段，栈，CPU提供的栈机制，栈顶超界的问题，push、pop指令，栈段；

重点：内存中如何定位存储单元，栈的定义和使用。

难点：栈的定义和初始化过程。

教学要求：学习和理解8086CPU对内存地址单位的访问方法，掌握使用DS和[address]方式访问普通内存单位的方式，掌握栈的初始化和访问方法。

教学方法：采用翻转课堂教学+案例方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。采用实例教学方法，结合学习的数据传送指令，栈操作指令，给出例程介绍内存访问的过程。

实践能力培养：本章内容主要建立学生对 8086CUP 内存访问理解，建立学生对计算机底层运行机制的了解，可以培养学士底层思维能力。

第 4 章 第 1 个程序（CM4）2 学时

教学内容：一个源程序从写出到执行的过程，源程序，编辑源程序，编译，连接，以简化的方式进行编译和连接，1.exe 的执行，可执行文件中的程序装入内存并运行的原理，程序执行过程的跟踪。

重点：汇编语言程序编写到执行的完整流程，汇编语言程序的调试过程。

难点：理解汇编语言编译和连接的作用，掌握 debug 的调试命令。

教学要求：学习理解编制汇编语言程序的完整流程，对基本的 debug 命令能正确使用。

教学方法：采用翻转课堂教学+案例方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。采用实例教学方法，结合实际例程展示程序 debug 调试过程，并在课后联系中留出问题思考，要求学生总结在 Masm 方式和 Debug 方式编制和执行汇编指令的差异性。

实践能力培养：培养学生对不同环境下汇编语言使用和调试的基本动手能力。

第 5 章 [BX]和 loop 指令使用（CM5）2 学时

教学内容： [bx]，Loop 指令，在 Debug 中跟踪用 loop 指令实现的循环程序，Debug 和汇编编译器 Masm 对指令的不同处理，loop 和[bx]的联合应用，段前缀，一段安全的空间，段前缀的使用。

重点：以 bx 寄存器作为偏移地址的内存单元访问方式，loop 指令的使用，循环程序的设计框架。

难点：如何正确设置循环程序的初始条件，理解通过修改寄存器内容访问连续的内存单元。

教学要求：学习和理解汇编语言中循环程序的设计方法，掌握使用 `bx` 寄存器和 `loop` 指令编制简单的循环程序。

教学方法：采用翻转课堂教学+案例方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。以实际的循环程序案例给出 `bx` 寄存器、`loop` 指令的具体应用说明。总结上次课程课后练习中的问题，使学生理解编译和调试环境中，对指令书写，数据类型等存在的差异。

实践能力培养：通过教学过程培养学生采用 CPU 视角进行程序逻辑设计的思考能力，对与未来面对底层开发基础能力有良好的启示作用。

第 6 章 包含多个段的程序（CM6）2 学时

教学内容：在代码段中使用数据，在代码段中使用栈，将数据、代码、栈放入不同的段。

重点：学习如何合理安排程序中数据、代码和栈，掌握段的定义和使用。

难点：理解段定义的灵活性，段的重叠问题。

教学要求：学生应掌握拥有多个段的汇编语言程序的编写和调试能力，学会合理安排程序对内存空间的使用。

教学方法：采用翻转课堂教学+对比教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。采用对比教学方法，按照不分段和分段方式完成相同功能的程序编制，对比分段模式的优越性。定义具有重叠特征的段，让学生思考段重叠后如何确定代码执行的入口位置？如何指定，不指定会产生什么样的结果？

实践能力培养：本章内容是较为完整地给出了真实汇编程序的结构。学生可以看到在汇编语言中，程序员可以直接安排自己的程序，实现面向硬件的编程，这将对理解计算机内部工作机制有极大的帮助，也有利于建立学生对计算机硬件知识的学习信心。

第 7 章 灵活定位内存地址的方法（CM7）3 学时

教学内容：`and` 和 `or` 指令，关于 ASCII 码，以字符形式给出的数据，大小写转换的问题，`[bx+idata]`，用`[bx+idata]`的方式进行数组的处理，`SI` 和 `DI`，`[bx+si]`和`[bx+di]`，`[bx+si+idata]`和`[bx+di+idata]`，不同的寻址方式的灵活应用。

重点：了解不同的寻址方式，掌握应用不同寻址方式解决实际问题方法。

难点：不同寻址方式中物理地址的计算方法，访问非连续内存单元时如何设置寄存器内容。

教学要求：学生应熟练掌握直接、寄存器间接、基址寻址、基址变址寻址等不同内存寻址方式，并理解不同寻址方式的应用方法和编制应用程序。

教学方法：采用翻转课堂教学+案例教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。结合循环程序设计基础，设计相对复杂的处理逻辑，使学生初步建立综合使用寻址方式的能力。

实践能力培养：本章教学引入了多种寻址方式，并通过案例介绍了与循环程序的综合使用，可以培养学生对相对复杂程序逻辑的编写能力。

第 8 章 数据处理的两个基本问题（CM8）2 学时

教学内容：bx、si、di、bp，机器指令处理的数据所在位置，汇编语言中数据位置的表达，寻址方式，指令要处理的数据有多长？，寻址方式的综合应用，div 指令，伪指令 dd，dup。

重点：了解 8086CUP 的数据类型，掌握数据的定义方法。

难点：dup 伪指令定义连续数据，定义后的数据在内存中的存放。

教学要求：学生应能了解 8086CUP 中支持的数据类型，并掌握数据定义的伪指令使用方式。

教学方法：采用翻转课堂教学+案例教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。综合运用已经学习到的循环程序设计、多种寻址方式设计对采用复杂结构体的数据处理的例程，巩固学习的知识。

实践能力培养：通过复杂的数据对象处理，让学生掌握汇编语言对结构体数据的定义和使用能力，基本具备使用汇编语言程序解决普通实际问题的能力。

第 9 章 转移指令的原理（CM9）2 学时

教学内容：操作符 offset, jmp 指令，依据位移进行转移的 jmp 指令，转移的目的地址在指令中的 jmp 指令，转移地址在寄存器中的 jmp 指令，转移地址在内存中的 jmp 指令，jcxz 指令，loop 指令，根据位移进行转移的意义，编译器对转移位移超界的检测。

重点：了解根据位移进行转移的工作原理，掌握 jmp, jcxz, loop 指令的应用场景。

难点：位移的计算，转移过程中对不同转移地址的具体要求。

教学要求：学生能够正确理解位移与转移地址的关系，并根据不同转移要求选择正确的转移指令设计程序逻辑。

教学方法：用翻转课堂教学+案例教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。以实例讲解转移位移在使用过程中的限制，使学生理解位移超界的问题。

实践能力培养：培养学生面向硬件编程时的正确思维习惯，增强底层编程时考虑问题的严谨能力。

第 10 章 Call 和 Ret 指令（CM10）2 学时

教学内容：ret 和 retf, call 指令，依据位移进行转移的 call 指令，转移的目的地址在指令中的 call 指令，转移地址在寄存器中的 call 指令，转移地址在内存中的 call 指令，call 和 ret 的配合使用，mul 指令，模块化程序设计，参数和结果传递的问题，批量数据的传递，寄存器冲突的问题。

重点：了解子程序设计的思路，掌握使用 call、ret 和 retf 指令编写子程序

难点：模块化程序设计思想，主程序与子程序之间的参数和结果传递，以及冲突情况下对寄存器的保护方式。

教学要求：学生能够掌握子程序的设计方法，使用 call、ret、retf 等指令完成子程序的编制，并熟悉子程序编写过程中参数结果传递，寄存器保护等问题的解决方式。

教学方法：用翻转课堂教学+案例教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。以实例讲解子程序编制过程，使学生理解子程序的通用结构，并掌握参数和结果传递、寄存器保护的方法。

实践能力培养：培养学生在逻辑上抽象共性问题的能力，提高解决复杂工程问题中的解构能力。

第 11 章 标志寄存器（CM11）2 学时

教学内容：ZF 标志，PF 标志，SF 标志，CF 标志，OF 标志，adc 指令，sbb 指令，cmp 指令，检测比较结果的条件转移指令，DF 标志和串传送指令，pushf 和 popf，标志寄存器在 Debug 中的表示。

重点：8086CPU 中标志寄存器中标志位的构成，使用标志位实现对程序逻辑的控制。

难点：指令执行对标志位的影响。

教学要求：学生能够区分标志寄存器中不同标志位的含义和标志位在寄存器中的具体为准，并能够使用 cmp、test 等指令和条件转移指令完成对程序的逻辑控制。

教学方法：用翻转课堂教学+案例教学方法，在课堂内，针对重点及难点进行探讨与讲解。通过设置的测试点，引导学生思考和验证学生对基础知识的掌握程度。以实例讲解指令对标志位的影响，使学生理解标志寄存器的作用，并掌握利用比较指令和条件转移指令进行程序逻辑控制的方法。

实践能力培养：培养学生通过综合运用条件转移指令编制程序的动手能力，为具备较全面的硬件编程能力和计算机运行机制的理解能力奠定基础。

（二）自学内容和要求

1. 自学内容

第 12-15 章 内中断，int 指令，端口，外中断，直接定址表。

2. 应达到的要求

了解 I/O 端口编址的方法及其指令的寻址方式；几种输入输出控制方式的工作机制及他们之间的差异。了解中断基本原理及中断的处理过程。

（三）实践性教学环节和要求

1. 主要内容和学时分配

总上机学时数 24 学时，其中课内 8 学时，课外 16 学时。学时分配如下：

汇编源程序的上机调试操作基础训练（2 学时）；数制、码制和子程序的编程与调试（2 学时）；分支与循环程序的调试与运行（2 学时）；串操作指令及其应用程序的设计与调试运行（2 学时）；综合应用程序的设计与调试（16 学时课外）。

2. 应达到的要求

掌握文本编辑器的基本用法、汇编语言源程序的两种书写格式，汇编语言程序的汇编、连接和运行方法，DEBUG 调试程序的常用命令和用法。掌握汇编语言中各种制数和码制的含义用途及互相变换、8086 系列 CPU 主要寄存器的结构、功能和用法；掌握子程序的编程与调试方法。掌握用 DOS 功能调用，实现字符的输入和显示。掌握分支程序的结构，分支程序的设计与调试方法；掌握循环程序设计与调试方法；通过编写包含有分支、循环和子程序汇编源程序，汇编、连接和调试，对程序中的错误进行查找定位和修正，检查程序运行过程中某些关键寄存器的值，从而掌握正确分配与使用寄存器。熟悉串操作指令的基本格式和使用方法，掌握常用的串操作程序的设计方法；熟悉串操作程序的调试运行过程。熟悉一个综合应用程序的设计与调试运行过程。

三、考核方式

本课程考核评价环节组成：课程作业评价（占 20%）、课程实验评价（占 20%）和期末考试评价（占 60%）三个环节。

课程作业考核评价依据学生完成课程作业的正确性、合理性、针对性、规范性情况给出成绩，学生缺勤、缺课程讨论可以在课程作业成绩中扣负分（缺 1 次扣 1 分）。课程作业评价内容针对本单元所属课程目标设计。评价标准见本课程作业评分细则。

课程实验评价依据由实验报告给出实验成绩。主要依据实验报告评价学生报告规范性。课程实验评价内容针对课程目标 CO1 设计。评价标准见本课程上机测评评分细则。

期末考试评价依据学生回答期末试题的正确性，给出期末成绩。期末考试评价内容针对课程目标 CO1 设计。评价标准见本课程期末试卷评分细则。

四、建议教材及参考资料

教 材：

《汇编语言》（第三版）王爽主编 清华大学出版社 2013.09

参 考 书：

1.《8086/8088 宏汇编语言程序设计教程》（第二版）王正智 编著电子工业出版社 2002.3

2. 《IBM-PC 汇编语言程序设计》（第 2 版）沈美明等编著 清华大学出版社 2001.8