**OS Lab2**

1. 请你谈谈对多层语言模型的理解，即为什么需要有机器语言、汇编语言和高级语言三层？

机器语言是01序列，是数字电路可以直接理解的语言，可以直接执行，速度最快。但机器语言不便于人类阅读与理解，不便于编程与调试。且对CPU类型的依赖强（不同CPU的机器语言不同），通用性差。

汇编语言的出现是为了方便程序员对程序的理解与编写。汇编语言将各种指令的01序列抽象成一个个助记符，如减法指令的01序列抽象成sub，这就使得人们能更好地理解与编写程序。除此之外，汇编语言屏蔽了硬件与指令集的差异：如X86与RISCV可能都有mul指令，但是这两种指令集的mul指令对应的二进制串可能不一样，但在汇编语言层面都可以用mul来表示，这就屏蔽了硬件与指令集的差异。

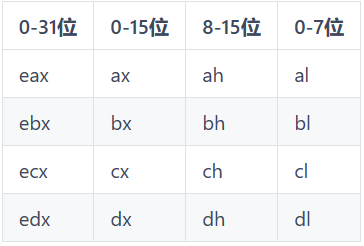
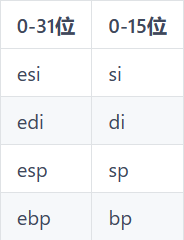
即使汇编语言隐藏了机器语言的内容，但过程调用、条件分支等代码段的汇编语言表达仍然十分复杂。而且一个程序，在不同指令集下的汇编语言表示可能也有所不同。高级语言的出现简化了过程调用，条件分支的代码，而且屏蔽了指令集的差异，由编译程序实现高级语言到汇编语言的翻译。通过这层抽象，人们可以忽略硬件和指令集的差异，更好地专注于程序本身，更利于对程序的理解于编写。

1. 请你描述下IA-32处理器的种类和用法，例如eax又可以分为哪几个寄存器来访问？esp的用途是什么？

分类：通用寄存器，段寄存器，指令寄存器，状态寄存器

**通用寄存器：**

有eax,ebx,ecx,edx,ebp,esp,esi,edi八个，都是32位的，用于算术运算，数据传输，32位寄存器用于保护模式，为了兼容16位的实模式，每一个32位寄存器又可以拆分成16位寄存器和8位寄存器来访问。例如ax是eax的低16位，ah是ax高8位，al是ax的低8位。ebx，ecx，edx也有相同的访问模式。如下所示。

e表示extend的意思。eax,ebx,ecx,edx可以按32、16、8位访问，esi,edi,esp,ebp只可以按32、16位访问。

有eax,ebx,ecx,edx,ebp,esp,esi,edi八个，都是32位的，用于算术运算，数据传输，32位寄存器用于保护模式，为了兼容16位的实模式，每一个32位寄存器又可以拆分成16位寄存器和8位寄存器来访问。例如ax是eax的低16位，ah是ax高8位，al是ax的低8位。ebx，ecx，edx也有相同的访问模式。如下所示。

**段寄存器：**

有cs,ss,ds,es,fs,gs六个，用于存放段的基地址。cs表示code segment，ss表示stack segment，ds表示data segment，es表示extend segment。fs,gs指向其他数据段的基地址。

**指令寄存器：**

指令寄存器eip存放下一条指令的地址，程序正常执行时每条指令执行后eip的内容加一，但有些指令可以直接改变eip的内容，如branch指令，jump指令，ret指令等

**状态寄存器：**

状态寄存器eflags存放CPU的一些状态标志位。下面提到的标志位实际上是eflags的某一个位。常用的标志位如下。

* 进位标志(CF)。在无符号算术运算的结果无法容纳于目的操作数时被置1。
* 溢出标志(OF)。在有符号算术运算的结果无法容纳于目的操作数时被置1。
* 符号标志(SF)。在算术或逻辑运算产生的结果为负时被置1。
* 零标志(ZF)。在算术或逻辑运算产生的结果为0时被置1。

1. 请查阅相关资料，说说eflags的各个位有什么含义？
2. 什么是线性地址？实模式的寻址模式是什么？地址空间大小如何？
3. nasm汇编中的内存寻址方式有哪些？语法是什么？请分别描述。
4. 在什么情况下会使用默认寄存器cs，ds，ss？如何避免CPU在计算线性地址时使用默认寄存器？