

Chapter 4 Machine Language

- *Make everything as simple as possible , but not simpler.
- 让每件事尽可能简单，而又不简单过度
- Introduction
 - 计算机的描述：结构化
 - 图灵：固定硬件通过加载软件来完成功能
 - 机器语言是一种约定的形式，用来对底层程序进行编码，从而形成机器指令
 - 是硬件与软件相接的中间线
 - Hack机器语言包括二进制版本与符号汇编版本
- Background
 - 为了对机器语言做一般性描述，只需要集中讨论三个主要的抽象体：处理器，内存，寄存器
 - 机器
 - 语言
 - 命令
- Specification
 - Hack computer:
 - A指令 (Addressing)
 - C指令
 - $dest = comp ; jump$
 - 指令代码的描述：计算什么？存储到哪？下一步做什么？
 - Computation：计算规范
 - Destination：目的地规范
 - Jump：跳转规范
 - jump通知计算机下一步将执行什么命令
 - 默认情况将执行下一条指令
 - 获取其他地址的一条指令（A寄存器会记录这个地址）
 - jump的比较规范都是相对于0而言

j1 (out < 0)	j2 (out = 0)	j3 (out > 0)	助记符	作 用
0	0	0	null	No jump
0	0	1	JGT	If out > 0 jump
0	1	0	JEQ	If out = 0 jump
0	1	1	JGE	If out ≥ 0 jump
1	0	0	JLT	If out < 0 jump
1	0	1	JNE	If out ≠ 0 jump
1	1	0	JLE	If out ≤ 0 jump
1	1	1	JMP	Jump

- A寄存器的冲突

使用 A 寄存器的冲突 A 寄存器使用上的冲突正如刚才所讲，程序员可以使用 A 寄存器为“包含 M 的 C-指令”指定数据内存中的地址，也可以为“包含 jump 的 C-指令”指定指令内存中的地址。因此，为了避免 A 寄存器在使用上的冲突，在编写良好的程序中，在可能引发 jump（即有一些非零的 j 位）的 C-指令中不能引用 M，反之亦然（即在“引用 M 的 C-指令”中也不准引发 jump）。

- Hack程序

- 符号

- 输入/输出

- 语法规约/文件格式

- Perspective

- Hack programing

- Project

- 机器语言顶层编程

- tool：汇编编译器，CPU仿真器

- CPU仿真器将汇编程序翻译为二进制代码，加载到硬件平台

- 乘法程序

- I/O处理程序

以上内容整理于 [幕布文档](#)