第1章

操作系统的几个抽象概念： 进程 虚拟内存 文件

文件是I/O设备的抽象表示，虚拟内存是对主存和磁盘I/O设备的抽象表示，进程是对处理器、主存和I/O设备的抽象表示。

进程是操作系统对一个正在运行的程序的抽象。

虚拟机是对整个计算机的抽象，包含操作系统 处理器 和程序。

操作系统保持跟踪进程运行所需的所有状态信息。该状态，称为上下文。

在linux中地址空间最上面的区域保留给操作系统的代码和数据，对所有进程都一样。地址空间的底部区域存放用户进程定义的代码和数据。从低地址向上具体如下：

程序代码和数据： 对所有进程，代码都从同一固定地址开始。进程开始时固定了大小。

堆：代码和数据后的为运行时堆，可动态扩展和收缩

栈：用户虚拟地址空间顶部为用户栈。用来实现函数调用。

内核虚拟内存：地址空间顶部区域是为内核保留的。不允许应用程序读写该区域或直接调用内核代码定义的函数。必须通过内核执行这些操作。

小头端和大头端：

小头端：低地址保存低有效位，高地址保存高有效位，与数学相似。

大头端：按照书写的方式，从低到高依次表示。

如i=0x01234567 小头端：67 45 23 01

大头端：01 23 45 67

计算机使用补码表示数字

无符号加法的溢出：x +y=s，其中s<x或s<y，则发生了溢出

有符号加法的溢出：x<0,y<0，x+y=s,s>=0时，发生了负溢出

X>0,y>0,x+y=s,s<=0,发生了正溢出

以寄存器为目标，对于生成小于8字节结果的指令，寄存器中剩余的字节的结果的规定：生成1 2字节数字的指令会保持剩下的字节不变;生成4字节数字的指令会把高位4个字节置为0

数据传输指令

Movb movw movl movq, 源操作数与目的操作数位数相同

立即数中 0xff,十六进制表示的是十六进制的字面值，$-1表示的是-1的补码表示

movq指令只能以表示为32位补码数字的立即数作为源操作数，然后把这个值符号扩展得到64位的值;

movabsq 指令能够以任意64位立即数作为源操作数，并只能以寄存器作为目的。

Movz,movs :源操作数位数小于目的操作数,把数据从源（寄存器或内存中）复制到目的寄存器。

Movz指令进行零扩展， movzbw movzbl 将1字节数字零扩展到2字节，4字节。

Movs指令进行符号扩展， movsbw movsbl 将1字节数字符号扩展到2字节，4字节

条件码

CF：进位标志。最近的操作使最高位产生进位。可用来检查无符号操作的溢出，结果正确

ZF：零标志。最近的操作得出的结果为0

SF：符号标志。最近的操作得到的结果为负数。

OF：溢出标志。最近的操作导致一个补码溢出-正溢出或负溢出（结果错误），用于有符号数字

Leaq指令不改变条件码，因为它是用来进行地址计算的。

指令

Cmp s1,s2 基于 s2-s1 设置条件码，不改变其他寄存器。

Test s1,s2 基于s1&s2

Cmp和test都有后缀，b w l q，表示值的位数

条件码的访问：

条件码无法直接访问。方法：

1. 可以根据条件码的某种组合，将一个字节设置为0或1，指令 set，该类指令的后缀表示不同条件。一条set指令的目的操作数是低位单字节寄存器元素之一，或是一个字节的内存位置，指令将这个字节设置为0或1

所有set指令都适用的情况为：执行比较指令，根据计算t=a-b设置条件码。更具体点，假设a b t分别表示变量a b t的补码形式表示的整数，t=a-b，

比较无符号时，当a-b<0时，cmp设置进位标志，无符号比较使用的是进位标志和零标志。

2）可以条件跳转到程序的某个其他部分

3）可以有条件的传送数据

跳转指令 jmp 无条件跳转 目的地通常为label指明，产生目标代码文件时，汇编器会确定所有带label-标号指令的地址，并经跳转目标（目的指令的地址）编码为跳转指令的一部分。

指令jmp存在直接跳转和间接跳转，直接跳转是给出一个标号（label)作为跳转目标，间接跳转的写法是 ‘\*’后面跟一个操作数指示符。具体实例如下：

Jmp .L1 跳转到 .L1

Jmp \*%rax 用寄存器%rax中的值作为跳转目标

Jmp \*(%rax) 以寄存器%rax中的值作为读地址，从内存中读出跳转目标。

跳转指令还存在其他跳转指令（有条件跳转）

跳转指令的编码方式：

1. 相对, 将目标指令的地址与紧跟在跳转指令后面那条指令的地址之间的差作为编码。这些地址偏移量可以编码为1 2 4个字节。
2. 给出“绝对”地址，用4个字节直接指定目标。汇编器和连接器选择合适的跳转目标编码。

注意：以上的相对寻址表明，程序计数器的值是跳转指令后面的那条指令的地址，而不是跳转指令本身的地址。