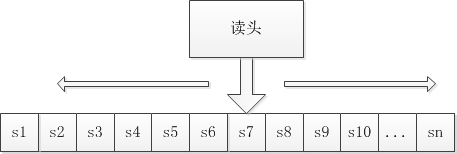
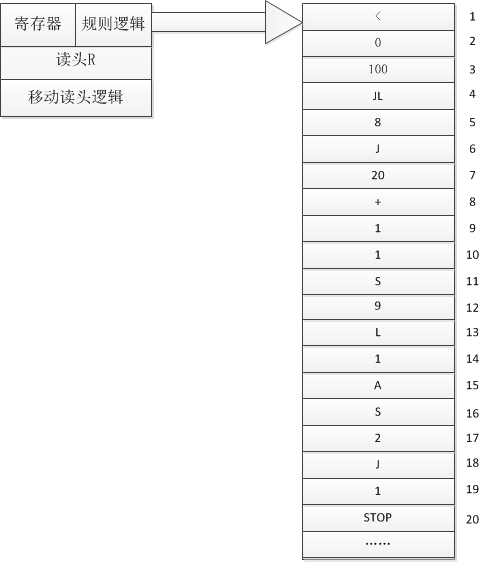
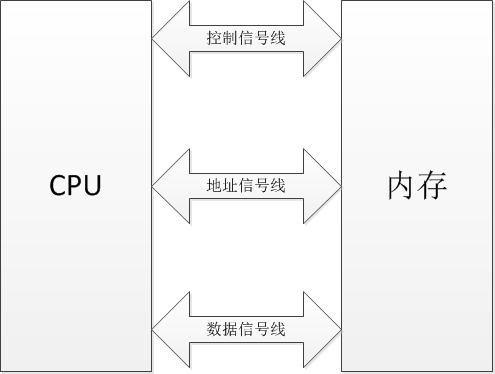
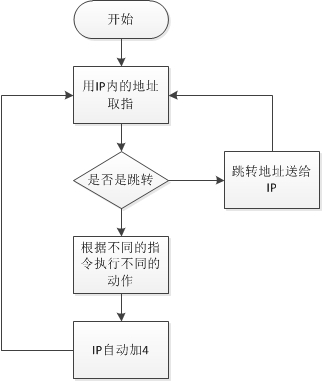
       你或许已经卷起了衣袖，或许在摩拳擦掌，正准备大干一场，打一场硬仗。  
年轻人嘛，行事总是风风火火的。但不是笔者扫你的兴，泼你冷水，在我们写  
代码之前还有很长一段路要走，要静下心来。如果写操作系统是一次旅行的话，  
那么千万不要错过沿途的风景……  
　　我们先来看看本章节都有些什么，我们要解决什么问题。本章节描述了下  
面四个问题：  
       1. 图灵机：从硬件的角度，分析了程序的执行。  
       2. 计算机程序的元素：描述了一段程序的里面究竟都包含了些什么东西。  
       3. 计算机程序的结构：描述了程序两个最基本的结构：顺序和跳转。如  
           果没有这两种结构会是什么结果。  
       4. 一条指令的执行：通过一条指令的执行，描述了一个计算平台的内部  
           组件是如何协同以便完成这条指令的执行的。  
       如果读者对以上四点早已了然于胸，那么读者大可跳过这一章节，大可  
不必在此浪费间，听笔者在这儿唠唠叨叨。  
       操作系统也是软件，也是一大堆程序组成的，所以不要觉得它多么神秘。  
既然是程序，程序又是由一条条各种这台计算机所支持的指令构成的，执行  
程序就是执行这一条条指令。那么我们首先要清楚计算机内部是如何执行这  
些指令的，以及执行一条指令后有什么结果是什么状态，计算机执行完一条  
指令后下一步怎么办……也许你会认为这对写软件的人没什么重要，这难道  
不是硬件工程师该关注的吗？是的，如果你是开发应用软件，可能不需要知  
道这些，但是我们是写操作系统，操作系统是硬件上的第一层软件，所以我  
或多或少的了解要一些硬件相关的东西。要解决这些问题，就不得不从图灵  
机开始……  
       说到图灵机，我们首先要说说图灵这个人。笔者觉得我们这种搞计算机  
的人都应该知道并记得这个人。  
       图灵，1912年6月23日生于英国帕丁顿。是数学家、密码破译专家，当  
然还有很多“家”，我们就先不说了。后来他又到美国普林斯顿大学取得博士学  
位，二战爆发后返回英国，帮助军方破解德国的著名密系统Enigma，帮助盟  
军取得了二战的胜利。啊，看到这里你是不是瞬间觉得他很伟大了。不过笔  
者认为他真正伟大的地方不在这里，而是他提出了两个最重要的东西——图灵  
机和图灵测试。也是因为这两个东西，他后来被人们尊称为计算机之父、人  
工智能之父。人们为了纪念他，专门设置了图灵奖，学计算机的不会不知道  
图灵奖吧。图灵没有恋爱（不知道是否是同性恋），孤独的走完了短短42  
岁的生命，英雄总是孤独的…...关于图灵个人的信息我们就说到这儿吧。下  
面我们去看看我们关注的图灵机。  
       图灵先后提出了图灵机和图灵测试，我们这里只关注图灵机，看看它究竟  
有什么神奇之处，又是如何与我们现代的计算机关联起来的。  
       图灵机是图灵提出的一种思想模型，是抽象的，是存在于大脑之中、存在  
于想象之中的。也就是说图灵并没有做出他所描述中的这种物理机器。那么这  
种机器是什么样子呢？它到底能做些什么呢？简单的说它是这样的，它有一条  
无限长的纸带，纸带分成了一个一个的小方格。有一个机器头在纸带上移来移  
去。机器头能根据读取小方格信息作出一些动作。在每个时刻，机器头都要从  
当前纸带上读入一个方格信息，然后自己的作出相应的动作，比如向小方格内  
写入信息或者移动自己到下一个小方格。如此反复直到遇到一个小方格内放的  
是停止标志。如图（P-2.1.1.1）所示。  
   
  
                                               图（P-2.1.1.1）  
       上面概述了什么是图灵机，下面我们就来仔细的看看这图灵机。图灵想出  
这种机器，是想用这种机器来模拟人们用纸笔进行数学运算的过程，他把这样  
的过程看作下列两种简单的动作：  
　　 1.在纸上写上或擦除某个符号；  
　　 2.把注意力从纸的一个位置移动到另一个位置；  
　　 而在每个阶段，人要决定下一步的动作，依赖于此人当前所关注的纸上某  
个位置的符号和此人当前思维的状态。  
　　 为了模拟人的这种运算过程，图灵构造出一台假想的机器，该机器由以下  
几个部分组成：  
　　 1.一条无限长的纸带。纸带被划分为一个接一个的小格子，每个格子上包  
          含一个来自有限字母表的符号，字母表中有一个特殊的符号表示空白。  
          纸带上的格子从左到右依此被编号为 0,1,2,...，纸带的右端可以无限伸  
          展。  
　　 2.一个读写头。该读写头可以在纸带上左右移动，它能读出当前所指的格  
          子上的符号，并能改变当前格子上的符号。  
　　 3.一套控制规则。它根据当前机器所处的状态以及当前读写头所指的格子  
          上的符号来确定读写头下一步的动作，并改变状态寄存器的值，令机器  
          进入一个新的状态。  
　　 4.一个状态寄存器。它用来保存图灵机当前所处的状态。图灵机的所有可  
          能状态的数目是有限的，并且有一个特殊的状态，称为停机状态。  
　 　读者或许觉得这太简单了，就这么简单的思想。是啊，在人类的世界里构  
建出的任何复杂的东西，都是从最简单的思想开始的。就像瓦特因为壶盖会动  
制造出蒸气机一样。只是想出这些简单思想的人不是我们。但我们不必灰心，  
也许有天我们能比他们做出更伟大的产品。过去属于他们，但未来是我们的。  
　　 好了关于图灵机的介绍，笔者就不在啰嗦了，总体感觉是这个假想的机器  
很简单。和我们现代用到的计算机好像差的很远，甚至没有什么联系。下面我  
们就来用个实例来看看图灵是如何运转并最终完成计算任务的。为了正确并顺  
利的说明这个实例在图灵机上的运转，我们得把图灵机小小的扩展一下，增加  
几条规则。好，我们继续往下看。  
　　 如果让读者用C语言写出一个1加到100的程序，笔者可能立马就能听到一  
阵不屑一顾的声音，然后是读者以迅雷不及掩耳之势“盲码”出和下面差不多的代  
码。

代码：  
……  
int sum=1;  
for(int i=0;i<100;i++)  
{  
 sum++;  
}  
……  
       写这几行代码读者可能不需要20秒钟就能完成，不需要5秒钟就能看懂它，  
甚至不需要2秒钟就能知道它的结果。但是我们要想让我们的图灵机去执行这计  
算过程，就不是那么容易了。可能我们的图灵机和图灵描述的机器有点差异，  
但我们保留了图灵的核心思想，所以我们的机器仍然可以称为图灵机。  
　　首先我们假定我们的图灵机能根据纸带上小方格中的信息执行如下规则：  
      1.加法规则；当图灵机在小方格内读到的符号是“+”时，我们的图灵机就  
         执行加法规则，并且规定紧接着下面两个方格里放着两个加数。执行  
         的动作是：将两个数相加结果放在内部寄存器中。  
      2.存放规则；当图灵机在小方格内读到的符号是“S”时，我们的图灵机就  
         执行存放规则，并且规定紧接着下面1个方格里放着存放的地址，也就  
         是哪个小方格。执行的动作是：将内部寄存器中的数存放在指定的小  
         方格中。  
      3.装载规则；当图灵机在小方格内读到的符号是“L”时，我们的图灵机就  
         执行装载规则，并且规定紧接着下面1个方格里放着装载的地址，也就  
         是要装载哪个小方格。执行的动作是：装载指定的小方格中的数到内  
         部寄存器中。  
      4.跳转规则；当图灵机在小方格内读到的符号是“J”时，我们的图灵机就  
        执行跳转规则，并且规定紧接着下面1个方格里放着跳转的地址，也就  
        是跳到哪个小方格。执行的动作是：将读头移动到指定的小方格上读取  
        这个小方格中的内容。  
      5.小于则跳转规则；当图灵机在小方格内读到的符号是“JL”时，我们的图  
        灵机就执行小于则跳转规则，并且规定紧接着下面1个方格里放着跳转  
        的地址，也就是跳到哪个小方格。执行的动作是：根据内部寄存器中的  
        内容，决定是否将读头移动到指定的小方格上，否则不执行任何动作，  
        读头直接顺序移动一个小方格并读取其内容 。  
       6.比较规则；当图灵机在小方格内读到的符号是“<”时，我们的图灵机就  
          比较规则，并且规定紧接着下面两个方格里放着两个比较的数。执行  
          动作是：将两个数相减，看结果是负数、还是正数、或者是零，把这  
          状态放在内部寄存器中。读头直接顺序移动一个小方格并读取其内容。  
       7.累加规则，当图灵机在小方格内读到的符号是“A”时，我们的图灵机就  
          行累加规则。执行的动作是：将内部寄存器中的值自动加一，读头顺  
          移动一个小方格并读取其内容。  
       现在我们的图灵机有了7条执行规则，我马上就用我们的这些规则来描述  
上面C代码，把我们的C代码转换成我们图灵机能识别的规则。转换的结果如下：  
1：  <  
2：  0    //对应于上面C代码中i变量。   
3：  100  
4：  JL  
5：  8  
6：  J  
7：  20  
8：  +  
9：  1    //对应于上面C代码中sum变量。  
10： 1  
11:  S  
12： 9  
13： L  
14： 2  
15： A  
16： S  
17： 2  
18： J  
19： 1  
20： STOP  
下面我们就把上面这些规则和数据依次放进我们图灵机的纸带的小方格中。最左  
边的数字代表纸带上小方格的编号，STOP表示图灵机的停机标志，当读头读到这  
个标志，我们的图灵机就停止运行。如图（P-2.1.1.2）所示:  


                                                   图（P-2.1.1.2）  
       我们来开始运行这个图灵机，看看它是如何完成上面C程序的计算任务的。我  
们假定图灵机开始运行时的读头R指向的1号方格。  
       1. 读头R指向1号方格:读出“<”号，根据上述规则，读头R继续下移分别读取2、  
           3号两个方格的数据并对其执行比较动作。读头R继续下移。  
       2. 读头R指向4号方格:读出“JL”符号，根据上述规则，读头R继续下移数据读  
           取5号方格并根据上一步比较的状态，对其执行跳转动作。因为0<100，  
           所以读头R跳转到8号方格。  
       3. 读头R指向8号方格:读出“+”号，根据上述规则，读头R继续下移分别读取9、  
          10号两个方格的数据并对其执行加法动作，结果放在寄存器中。读头R继续  
          下移。  
       4. 读头R指向11号方格:读出“S”符号，根据上述规则，读头R继续下移读取12  
           号方格的数据并对其执行存放动作。将寄存器的结果存放到9号方格中，因  
           此9号方格中变成了2。读头R继续下移。  
       5. 读头R指向13号方格:读出“L”符号，根据上述规则，读头R继续下移读取14  
           号方格的数据，并装载1号方格的数据到寄存器中，因此寄存器中是0。读  
           头R继续下移。  
       6. 读头R指向15号方格:读出“A”符号，根据上述规则，对寄存器执行累加动作。  
           由于第6步中寄存器中保存的数据是0，累加后变成了1。读头R继续下移。  
       7. 读头R指向16号方格:读出“S”符号，根据上述规则，读头R继续下移读取17号  
           方格的数据并对其执行存放动作。将寄存器的结果存放到2号方格中，因此  
           2号方格中变成了1。读头R继续下移。  
       8. 读头R指向18号方格:读出“J”符号，根据上述规则，读头R继续下移数据读取  
           19号方格的数据并执行跳转动作。所以读头R跳转到1号方格。又从第1步开  
           始重复执行到第8步。  
       9. 反复执行第1步到第8步100次后，会发现2号方格中的数据不在小于100，因  
           此JL规则不会发生跳转。读头R继续下移指向6号方格。  
       10. 读头R指向6号方格:读出“J”符号，根据上述规则，读头R继续下移数据读取  
            7号方格的数据并执行跳转动作。所以读头R跳转到20号方格。  
       11. 读头R指向20号方格:读出“STOP”符号，由于这个符号表示停机，如果到这  
            里我们图灵就停止了。  
       啊哈！终于停机了。我们也可以松一口气了，奇怪吧，如此简单的程序让图灵  
机去运行却变得如此复杂，因为它是机器不是人，没有人如此高等的智慧。只懂几  
个有限的规则和动作，傻傻在纸带上蹦来跳去的。  
       如果你能一字不漏的看到这里，说明你比笔者的耐性好很多。我们可以暂时歇  
息一下出去仰望一下天空，或者喝杯茶……  
       你或许发现我们给这个图灵机赋以的规则越多，这个机器的功能也就越强，我  
还可以赋以它减法规则、乘法规则、除法规则、移位规则、数据交换规则、各种跳  
转规则等。还可以在读头里加入更多的寄存器。  
       时间在向前走，人类从未停止对这种机器的神往。人们不断的尝试着各种电子  
元件，梦想着有天能做出完成上面这些计算规则的电路和实现那条纸带的电路。终  
于人们用晶体管配合其它电子元件，做出了加法器、乘法器、除法器、移位器、译  
码器……最重要的还有存储器。人们又做出了用于控制这些部件的逻辑，比如：什  
么时间开始访问存储器，又在什么时刻进行控制这些部件完成各自的计算任务。最  
后人们把这些部件封装在一起，并起了个响亮的名字——CPU。现代CPU无论是通用  
计算机的CPU还是嵌入式系统的CPU。逻辑上都和下图差不多，如图（P-2.1.1.3）  
所示。  


                                     图（P-2.1.1.3）  
　　CPU就好像是我们图灵机的那个读头，只不过它比我们的那个读头所具有规则  
功能要多很多，执行规则的速快也快很多。做CPU的工程师还对这个CPU能完成的  
所有规则进行统一的编码，并对每个编码写明它是干什么的，比如：完成加法运算、  
数据传送、数据移位……给这个编码集起了个名字——指令集，最后把这一套指令集  
交给程序员。程序员用这些指令不同的组合，构建出了各种复杂的程序……  
　　但是我们用一条条指令写出的程序还没有地方放呢，更不谈让CPU去执行了。  
所以还要有像图灵机那样的纸带。于是人们又做出了存储器，叫内存。也许今天  
的内存所用的技术和电子元件比以前所用的可能高级好多倍。但是我们从逻辑上看  
仍然和下图差不多。如图（P-2.1.1.4）所示:  


                                   图（P-2.1.1.4）  
       内存中的每个位都能表示两种状态，如果你还能想到这两种状态就是我们逻辑  
上常常说的0和1，那就太好了。内存中8个状态位，组成一个字节，每个字节分配  
一个编号，叫内存地址，每个以字节为单位的内存单元中保存的电子状态，就是我  
们逻辑上说的数据。内存中放的数据可以是常规数据，也可以是CPU定义的指令。  
CPU主要的任务就是根据指令“改变”这些电子状态，即我们说的数据。“改变”的手段  
有多种：比如对这些电子状态做加法运算、做移位计算等……  
       图灵机为什么要让读头能够在纸带上来回移动，原因很简单，当然是为了读取  
或者改写纸带任意位置上的小方格中的数据。我们要如何才能让CPU读取和写入内  
存单元呢，总不能让CPU在内存条上移来移去吧，即使这样可以，但是速度和稳定  
性实在是太差了。既然只有两个动作——读内存单元和写内存单元。那么我们期望  
是下面的情况：  
      1. 读内存单元  
          1.CPU对内存发出读内存的控制信号，并等待内存的确认信号。  
          2.CPU发现内存已确认，就发出地址数据信号。  
          3.内存收到CPU的地址，读出这个地址内存单元上的数据并通知CPU数据已  
             经准备好。  
          4.CPU通过数据信号线，把数据传送到CPU中。  
       2. 写内存单元  
　　    1.CPU对内存发出写内存的控制信号，并等待内存的确认信号。  
          2.CPU发现内存已确认，就发出地址数据信号。   
          3.内存收到CPU的地址，并等待数据信号线上的数据。  
          4.CPU通过数据信号线，把数据传送到内存单元中  
也许实际上比笔者描述的更加复杂，但是比图灵机那种纯粹的物理机械运动要好很多，  
让地址信号的变化，代替机械运动。于是，一个逻辑上最简单的计算系统就算做好  
了，如图（P-2.1.1.5）所示：  


                                               如图（P-2.1.1.5）  
       现代的CPU都有专门的指令地址寄存器（IP），这个寄存器里放着下次即将执行  
的指令的地址，即该指令放在这个地址的内存单元中。假如我们规定一条指令的长度  
为4个字节，每个执行完一条指令，IP就自动加4。如果是跳转指令就将跳转地址直接  
放进IP寄存器中，接着读取指令，然后执行指令，如此反复，直到切掉电源。下面我  
们用一副图来表示这种运行机制。如图（P-2.1.1.6）所示：  


                                               图（P-2.1.1.6）  
就这样在IP寄存器的引导下，CPU一条一条的、高速的执行着内存中的指令。时而顺序  
执行，时而跳转执行，它从来不问，也不知道执行那么条指令后会有什么结果，只是  
任劳任怨的完成每条指令规定的功能和动作。  
       不管现代的CPU功能多么强，速度多么快，体积多么小，功耗多么低，但它最核  
心的机理的实现从来没有跳出图灵的思想。或许你正用计算机听着杜比级别的音乐、  
看着1080P的电影、玩着如梦如幻的3D游戏、正在用3G移动电话和远方的恋人视频通  
话，但是别忘了，这些玩意儿都是从图灵的思想中诞生的……  
       到这里，你的大脑中应该能呈现出一副计算机如何执行程序的图像，如果能这样，  
自然最好，如果不能这样，也别怕，适当的囫囵吞枣，也不是坏事儿。休息一下我  
们继续赶路。