第一章

程序

黑匣子

1.硬件：
输入值存放到在硬件，整个运算过程由硬件控制。

*输出存储也是在硬件、

包含多种硬件：**输入设备，存储设备，运算控制设备（如：中央处理器cpu），输出设备。**

输入数据逻辑流动过程：

*数据传输——总线（**什么是总线）**

输入实数C（键盘）>>传送到内存（操作系统）>>运算数据（CPU）>>输出结果并显示（显示器）

功能：将用户需求转换为硬 件能够看懂的语言，同时也控制着硬件的操作步骤和顺序

2.软件

计算机语言控制CPU执行任务，指示其操作的语言：**程序语言**

计算机根据人们预定的安排自动地进行数据的快速计算和加工处理。人
们预定的安排是通过一个指令序列来表达的，这个指令序列称为程序
（或软件），一个程序规定计算机完成的一个完整任务。

软件描述需求，硬件实现需求。

3.操作系统
衔接软件与硬件、控制硬件工 作、为软件提供服务。

本质上是**一组程序。**

主要功能在于管理计算机的各种资源及周边设备的硬件。
1.管理文件系统，管理各种硬件资源，例如U盘、网络、键盘等 2. 管理程序共享的资源，例如CPU、主存等（一个计算系统会有多个 程序同时在执行或等待执行） 3. 管理和调度多个程序的执行 4. 提供程序和硬件的衔接，提供各种系统的服务和接口 5. 设法维护系统的安全，尽量防止病毒（恶意软件）有意或无意的侵 入

计算机系统可以被分为三个层次：硬件层，操作系统层，软件层。

冯诺依曼结构（存储程序计算机架构）

特点：1.采用二进制 2.计算机指令和数据存储在同一记忆装置中，即存储器3.计算机按照预先编制的程序顺序执行，即程序控制。

程序设计语言

按照程序设计语言规则组织起来的一组计算机指令称为计算机程序。

三个大类：**机器语言、汇编语言和高级语言**

机器语言：二进制语言

汇编语言：使用助记符与机器语言中的指令进行一一对应

高级语言：接近自然语言（第一个广泛应用的高级语言是C语言/1972年/

计算机执行源程序的两种方式：**编译和解释**

编译：将源代码一次性转换成目标代码

解释：将源代码逐条转换成目标代码同时逐条运行的过程

关于计算机

四个发展阶段：电子管、晶体管、集成电路（IC） 和超大规模集成电路（VLSI）

通用型计算机：

专用型计算机：/嵌入式系统/

第一台电子计算机：/电子数字积分计算机/ 1946年

第二代计算机：1958-1965 全部采用进踢馆作为电子器件

第三代计算机：1965-1970 以中小规模集成电路为电子器件

第四代计算机：1970以后 采用大规模集成电路和超大规模集成电路为主要电子器件

数据和大数据

数据：指所有能输入到计算机并被计算机程序处 理的，具有一定意义的数字、字母、符号和模拟量等的通称。

计算机利用模数转换器将模拟信号转换为数字信号

第二章

进位制

二进制数的1位：1bit

1字节=8bit

基数与位权

- **如果某一个进制采用R个基本符号，我们就称它为基R进制，R称为进制的“基数（base）”。例如二进制的基数是2，十进制的基数是10。**
- **进制中每一位的单位值称为“位权（weight）”。在整数部分，最低位的位权为R^0，第i位的位权为R^i；对于小数部分，小数点向右第j位的位权R^{-j}。**
  - **在十进制中，个位的位权是10^0，百位的位权是10^2，所以数7在个位时，它的值是7，在百位时它的值就是$700 = 7\times10^2$。在二进制中，最低位的位权是$1=2^0$，所以数1在最低位的值是$1 = 1\times2^0$。**
  - **小数是同样的道理。**

进制转换

计算过程略

有小数点的实数进行进制转换，对小数点先后进行分离分开运算
python实现：可以使用自带的partition（）函数进行分离
>>>(x,t,y) =bin.partition('.')
结果：x='1101', t='.', y='01'
//对于小数点后的位数，2^-1

计算机中二进制运算

1.中央处理器CPU通过引脚与外部连接并交换数据

a.一次只能处理有限数位的二进制数

b.一般能处理32/64比特的数据

2.计算机表示整数

分为：**带符号整数和无符号整数**

无符号整数：表示的是非负整数

带符号整数：可以表示正负整数和0

3.二进制加减法和乘除

n个比特所能表示的最大数是2n-1//注意运算溢出

原码反码补码

正数的这三个是相同的，负数的反为原码按位取反，补码为反码+1

作含有负数的加法则可以用补码相加。

逻辑运算

逻辑运算是对逻辑变量和逻辑运算符号的组合序列所作的逻辑推理

基本运算：与或非

计算机网络中用晶体管实现逻辑。//晶体管是以半导体材料为基础的元件。
每个晶体管可以在不改变自身内部结构的情况下，根据外部电源的变化而 展现不同的状态。 我们可以通过控制晶体管的电源来控制它们开或关的状态

运算方法

半加器和全加器的运算
//写表达式可以用真值表

数据的存储形式

计算机用二进制数的组合来表达所有需要保持的信息，这些二进制数的组 合按照一定的规则存放就形成了计算机里的数据。
 二进制数的数值0或1的存储方式跟随着物理介质的特性不同而不同，基 本是利用物理材料的电信号、磁信号之类的状态来存储0或1。这些载体 就称为存储介质。
 存储介质和辅助数据存储和数据读写的电路、设备等组合在一起，构成了 存储设备，例如我们常用的内存、磁盘、U盘等

在计算机内部，各种信息都是以二进制编码的形式存储的。
 在二进制编码中，指定不同数量的0或1形成的不同的组合表示不同的意 义。编码往往用到一大串0或1，必须要按照一定规则对这些信息进行分 割和识别才能获得有用的信息

计算机中的单位信息
位：bit
字节：=8bit //信息存储常用的基本单位
字：字节的组合，CPU以字为单位读写数据，大部分CPU的字长是32位/64位

存储设备：
　存放0和1组成的二进制信息的物理载体称为存储介质，存储 介质加上配套电路等组件就组成了存储设备

常用的存储设备：寄存器，高速缓存，内存，外存
速度：快>>>慢 容量：小<<<大

***对于每个设备的概念（最后五张ppt）**

第三章

引例

计算机中的两个核心部件：**CPU和主存**
CPU：做运算 主存：存储程序和相关的变量

每一条**程序语句**和每一个**变量**在内存中都有相应的内存地址
//执行a=a+1 这句指令本身也有一个内存地址

*不能直接对内存做运算//寄存器
　a=a+1 需要先读取a到寄存器R，对R+1，再存回内存

CPU中的核心部件

程序计数器 PC
特殊的寄存器，用于语句地址的储存，PC+1则自动指向下一条将要执行的程序语句//考虑while类语句则不只是顺序+1来指向

指令寄存器 IR
特殊的寄存器，用于存放读取的程序指令

*主存中是程序指令吗

算数逻辑单元 ALU
执行运算

汇编指令

读取a到R——load
load R1，（address） //address为数值a的地址，（address）为这个地址内存储的值

R赋值——mov//给寄存器R赋值
mov R1,constant //constant为数值时是十六进制常数，但也可以是寄存器，被赋值给R1寄存器

R+1——add
add R2, R1, constant //R2是储存运算结果的寄存器，后两个是做加法，constant依旧是十六进制常数
eg：add R2,R1,01h
R-1——sub　和add的使用方法一样

左移位指令——shiftl
shiftl　R3, R1, R2//意义：R1的二进制数左移R2位，移出的那几位填0。将最终值存入R3
右移位指令——shiftr　和上述类似

将R存回a——store
store (address), R

***控制语句**//分支跳转指令
使用分支跳转指令是为了实现循环（while/for）

比较x是否小于y——slt
slt R4, R1, constant //比较R1和constant，如果R1小于constant，则R4置1，否则置0
比较小于或等于——sle

选择跳转语句块——beqz指令
beqz R4, label　//如果R4中的数值为零，则跳转到标签 label2标记的指令块处

直接跳转语句块——goto指令
goto label

python函数调用

1.局部变量/全局变量
　全局变量使用global语句来声明

2.栈
　先进后出的特性

3.关于函数的调用
　*用栈存放返回地址//越早被调用的函数越晚返回

4.当函数执行时，这个函数的每一个局部变量就会在栈里有一个空间。 在栈中存放此函数的局部变量和返回地址的这一块区域叫做此函数的 栈帧(frame)

第五章

递归算法

加法问题：F(n) = F(n-1) +an
平面划分问题：F(n)=F(n-1)+n
汉诺塔问题：f(n)=2f(n-1)+1

分治法
求最小值：M(n)=min(an,M(n-1))
贪心算法//最优化问题

***关于如何计算时间复杂度和算法复杂度！！**

第六章

简介：

计算机系统的三个层次：**硬件，软件，操作系统**
操作系统是硬件和软件的中间桥梁，在计算机系统中对下层的硬件进行管理并对上层应用软件提供接口支持
便利，兼容，公平，安全

计算机启动

所有设备在开机启动过程中都会包括得共同阶段：启动自检阶段、初始化启动阶段、启动加载阶段
主要由 **BIOS**来完成

启动自检//加电自检：
接通电源，读取BIOS程序并对硬件进行检测。
功能：检查电脑整体状态是否良好

初始化启动阶段：
根据BIOS设定的启动顺序找到优先启动的设备。

启动加载阶段：
启动管理程序来处理核心文件的下载，这个启动管理程序被称为Boot Loader

内核装载阶段：
在操作系统开始使用内核程序测试和驱动外围设备时，操作系统的核 心才接管了BIOS的工作

登录阶段

操作系统

操作系统是管理计算机硬件与软件资源的计算机程序，是软件与硬件的中间接口

唤醒操作系统的方式叫做 **中断。**
硬件中断，软件中断，异常中断//异常中断不全是错误

硬件中断：
操作系统对硬件资源的管理，包括，I/O设备（键盘显示器等），计算机资源（CPU）和存储资源（包括内存和外存）

IO设备：可以与计算机进行数据传输的硬件
CPU通常使用**轮询**和**硬件中断**检测设备的工作状态
　不断查询设备的状态寄存器——轮询　//弊端：检测速度慢，可能存在饥饿状态即请求无法得到响应，处理中断事物不够灵活
　使用硬件中断码分辨是哪个硬件发起中断——硬件中断　//当 某个设备状态发生变化时可以主动通知CPU反应当前状态。每一个中断都有一个中断类型码。
中断向量表——从中断类型码链接到操作系统要执行的服务程序//中断服务程序
中断向量表和相关的中断服务程序是极其重要的， 需要特别保护起来，一般用户是不可以改变它们 的。
这些都是**放在操作系统的内核（kernel）保护起来**

对CPU的管理
操作系统需要合理的安排和调度任务，所以通过一种机制唤醒操作系统使其在不同任务间进行切换。
操作系统给每个任务一个时间片，用完之后Timer（硬件）发送中断给CPU，之后跳到Timer的中断服务程序执行（调动核心程序 **调度器**）
需要cpu执行的任务：**就绪任务**　维护了一个**就绪任务队列**进行存放。

进程
操作系统为每一个执行中的任务创建了一个 **进程**来保存每个任务执行时的所有环境信息。
进程的**一出一进**叫做 **交换**

对内存的管理
程序执行出现错误会发生 **异常中断**，但是更常发生异常中断的是**有情况需要操作系统管理内存**
管理内存使得 **多个任务可以共享内存资源**
需要对于内存中的数据进行换进换出的管理来应对内存不足的情况。如果读取时发现内存中无这个变量，就会发生**异常。**//会把这一变量存在的一块数据（页）从硬盘载入到内存里
执行语句时，变量未加载进内存，会造成访问变量的异常，出现了**异常中断（页错误）**然后跳跃到**页错**

误处理程序。0
如果没有空位存储该页，则采用不用的页替换算法。（最常用：LRU

操作系统提供服务

内核态和用户态
将指令集分为需要特权的和一般的//所有I/O指令都是需要特权的指令
CPU通过状态寄存器是那种状态来判断是否可以执行指令，用户态不可执行特权指令
执行每一个指令时自动检测

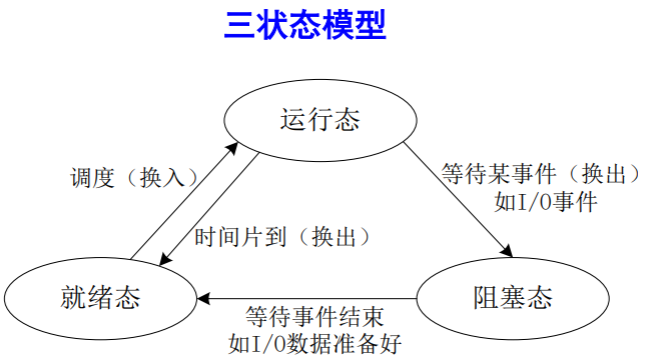
两种状态的转换
1.使用中断方式才能进入内核态。2.执行int指令状态自动置为内核态，它通过执行软件中断得到操作系统的服务。3.操作系统有一个中断向量表

除了操作系统以外的所有软件都运行于用户态

系统调用——软件中断
系统调用函数//内核态
int指令是唤醒操作系统提供服务而不是转换成内核态的，是执行的时候就会自动转换成内核态

****进程调度**

三状态模型



运行态：进程在处理机上运行时则为运行态//进程数目不能超过系统中处理器的数目
就绪态：获得了所需一切资源//除了处理机//一旦得到处理机就可以运行的状态
阻塞态：正在等待某一事件发生而暂时停止运行的状态

进程调度
进程执行所需要的时间：周转时间
平均周转时间：所有进程周转时间的平均数
吞吐量：单位时间内完成任务的数量

先来先到服务FCFS：不利于短作业，平均周转时间变长
短作业优先SJF：对长作业不利

文件系统

进程
包含了代码段、数据段、栈、堆、BSS段（未初始化全局变量）以及进程控制块（PCB）

第七章

并行计算

并行计算的基本思想是将被求解的问题分解成若干部分，每部分由一个 独立的计算资源处理

在硬件层面需要多个计算资源：一台计算机内拥有多个CPU核
核与核之间的通信（数据交换），被称为进程间的通信

两种并行计算的基本架构：**共享内存方式和消息传递方式**
共享内存中的变量S：临界变量//共享内存方式实现简单且速度快，系统中一般最多有64个核
消息传递的优点在于扩展性很好

如果程序的执行时间为T，现有P个核，运行时间一般不能达到T/P
1.程序不能刚好平均分成P段
2.消息传递也需要时间

Python实现

引入多进程包：import multiprocessing
Process 类用来创建子进程； ② Value 与 Array 类，用于创建共享内存变量与数组； ③ Event 与 Semaphore 类，用于维护进程间的执行顺序

进程通信

进程通信是指在运行的进程间传输数据，以达到多个进程能 够协同完成同一个任务的目的
最高效也最常用的通信方法：**共享内存**
共享内存是指在主存中开辟一个共享的存储区域，需要通信的进程将自己的一段地址空间映射到所开辟的共享存储区域

多个进程同时读写共享内存区域时，共享内存方式没有提供一个机制保证读写的顺序。
保证读写顺序的机制：Event类

第八章

计算机网络和物联网

计算机网络

基本功能：信息传送（连通性），资源共享（共享）
五层：**物理层，数据链路层，网络层，传输层，应用层**
应用层：应用进程间的交互完成特定网络应用，应用层协议定义应用进程之间饿通信和交互规则
传输层：复用和分用功能，负责提供通信之间的数据传输服务
网络层：选择合适路由，为不同主机提供通信服务
数据链路：将网络层交下来的IP数据组装成帧
物理层：包括考虑用电压代表1和0

物理层

实现数据在传输介质（有线，无线）上的传输。
要尽可能屏蔽掉不同传输媒体和通信手段的差异

通过编码，用电气方程表示01信号
不归零：正电平1负电平0
归零：正脉冲1负脉冲0
曼彻斯特编码：位周期中心的向上跳变 0，向下跳变 1。但也可反过来定义。

差分曼彻斯特编码：在每一位的中心处始终都有跳变。位开始边界有 跳变代表 0，而位开始边界没有跳变代表 1

信道复用技术
一条传输通道被多个信号共享的方式//降低成本提高利用率
主要包含：**时分复用，频分复用，码分复用**
频分复用：按照频率分配资源，同一个信道的不同频率范围来传输不同的信号
时分复用：按照时间周期性传输信号，一个周期就是一个TDM（时分复用帧）
码分复用：选择不同码型

数据链路层

数据链路层在发送端收到的是来自网络层的数据包，在接收端收到的是来自物理层的比特流
工作内容：给数据包包装//加上头部和尾部，其最终包含控制信息
控制信息包括：封装信息，差错信息，流量控制，链路管理

差错控制
最重要的功能之一
//在传输过程中的模拟信号在外部干扰中发生失真
检错技术：循环冗余检验技术，在传输信号后加入冗余码
//该技术只能检测到数据包是否出现比特差错，无法得知哪里出现也无法纠错。
只能保证收到正确的数据但不负责纠错

网络层

网络之间的互连是通过路由器实现的，路由器决定数据流向哪里
电路交换和包交换（现在计算机网络使用的都是包交换

路由器有一张路由表，记录了数据从哪里来到哪里去//其中指的是IP地址
IP地址是一个表示计算机在网络中位置的32为标识符
前一个字段是网络号/网络地址//标志网络
后一个字段主机号/主机地址//标志主机
主机号中的前半是子网号，这种做法叫做划分子网

子网掩码
左边部分一串1来对应网络号和子网号，右边对应主机
子网网络地址的计算
IP地址与子网掩码逐位相与得到网络地址

公网和内网

私网地址的三个IP段
10.0.0.0— 10.255.255.255（24位，约700万个地址）、
172.16.0.0— 172.31.255.255（20位，约100万个地址）、
192.168.0.0— 192.168.255.255（16位，约6.5万多个地址）

通过NAT协议将内网中主机和外界建立联系

传输层

保证应用程序之间的通信可靠性
常用的两个协议**无连接的用户数据报协议（UDP）和面向连接的传输控制协议（TCP）**

采用面向连接的 TCP 协议时，尽管下面的网络是不可靠的（只提供 尽最大努力服务），但逻辑通信信道就相当于一条全双工的可靠信道。 采用无连接的 UDP 协议时，这种逻辑通信信道是一条不可靠信道

UDP是无连接的，是不需要确认对方收到的，**TCP**是面向连接的协议，在进行通信之前需要建立连接
TCP的传输包括：建立连接，传输数据，释放连接//三次握手协议
TCP 报文段首部的前 20 个字节是固定的
报头包括几个字段：源端口、目的端口、序号、确认号、TCP头长 度、8个1比特的标志位、窗口大小和校验和。报头的这些字段可以确保TCP报文的可靠传输

三次握手

应用层

实现多个系统应用进程相互通信 的同时，完成一系列业务处理所需的服务

DNS（域名系统）
域名（Domain Name）是由一串用点分隔的名字组成的Internet上某 一台计算机或计算机组的名称，用于在数据传输时标识计算机的电子方 位。 ... . 三级域名 . 二级域名 . 顶级域名 mail.cctv.com
不区分大小写，英文和数字组成，每一个标号不超过63个字符
各级域名由上一级域名管理机构管理，最高级的由ICANN管理
顶级域名的三类：国家顶级，通用顶级，基础结构

- **国家顶级域名 nTLD，例如：**
  .cn 表示中国；.us 表示美国；.uk 表示英国等。
- **通用顶级域名 gTLD**
  最早的顶级域名是：
  1. **.com（公司和企业）**
  2. **.net（网络服务机构）**
  3. **.org（非赢利性组织）**
  4. **.edu（美国专用的教育机构）**
  5. **.gov（美国专用的政府部门）**
  6. **.mil（美国专用的军事部门）**
  7. **.int（国际组织）**
- **基础结构域名 (infrastructure domain)**
  只有一个，即 arpa，用于反向域名解析，因此又称为反向域名。

万维网

万维网是分布式超媒体 (hypermedia) 系统，它是超文本 (hypertext) 系统的扩充。 超文本是指包含指向其他文档的链接的文本(text)，一个超文本由多 个信息源链接成。利用一个链接可使用户找到另一个文档。这些文档 可以位于世界上任何一个接在互联网上的超文本系统中。超文本是万 维网的基础。 超媒体与超文本的区别是文档内容不同。超文本文档仅包含文本信 息，而超媒体文档还包含其他表示方式的信息，如图形、图像、声 音、动画，甚至活动视频图像

万维网以客户 - 服务器方式工作。 浏览器就是在用户计算机上的万维网客户程序。万维网文档所驻留的 计算机则运行服务器程序，因此这个计算机也称为万维网服务器。 客户程序向服务器程序发出请求，服务器程序向客户程序送回客户所 要的万维网文档。 在一个客户程序主窗口上显示出的万维网文档称为页面 (page)

网页访问流程
1.在浏览器输入域名
2.DNS转化成IP地址
3.获得IP地址，向该服务器发起访问请求。服务器收到访问请求之后查看自己域名下的网页
4.返回信息（包括代码文件，.html文件和图片等
5.将这些信息组成可以查看的网页//服务器发送的不是整个网页

物联网相关技术
感知互动层、网络传输层、应用服务层

第九章

常见网络威胁

网络钓鱼/钓鱼法，钓鱼攻击
目的：引诱受害人给出敏感信息
手段：电子邮件（虚假信息引诱），假冒网上银行盗取账号密码，虚假电子商务，木马，用户口令漏洞

恶意软件：
不只是弹出广告，可能盗走信息。
依赖主机程序：后门，**木马**，逻辑炸弹，**病毒**
独立于主机程序：**蠕虫**，细菌，拒绝服务程序
病毒：
计算机病毒所寄生的文件或者程序叫做**宿主**
① 计计算机不能正常启动或者可以启动但所需的时间较长； ② 经常出现黑屏甚至死机的现象，无法正常工作； ③ 运行速度变慢，常用的应用程序运行时间明显增加； ④ 不停弹出大量的恶意窗口或者广告；
⑤ 文件长度、类型或者内容异常，文件内出现乱码、文件无法显 示或者直接消失
感染和寄生是指**将其自身嵌入宿主指令序列中，病 毒成为程序的一部分，随宿主程序的执行而执行**
造成>>必须将其寄生的宿主一起消灭才能清除病毒
文件型病毒：感染文件，后缀exe/com
引导性病毒//开机型病毒：感染内存（磁盘引导区），只有在系统启动时才会发作和传播。

蠕虫：
与病毒相似，一种能够**自我复制**的计算机程序//莫里斯蠕虫（互联网蠕虫）通过互联网感染电脑
分为：**主程序和引导程序**
入侵计算机后运行引导程序（下载和安装主程序）

木马：
在远程计算机之间建立联系、使远程计算机能够通过网络控制本地计算机的非法程序
分为：**客户端程序和服务端程序。**
客户端程序：让攻击者远程控制已植入木马的计算机
服务端程序：用户计算机中的木马程序
常见：盗号木马，键盘记录

其他威胁
路由器DNS劫持：对请求返回假的IP地址或者什么都不做使其失去响应
拒绝服务：对某个**服务器**IP发送大量数据请求，消耗服务器网络资源，导致无法处理正常的用户请求
三次握手中找机会下手
① SYN洪泛攻击：攻击者向服务器发送大量的半连接请求
② LAND攻击：攻击者将源IP和目的IP均改成目标服务器的IP
③ Smurf攻击：攻击者伪造源IP为目标IP向局域网中的所有 主机发送ICMP应答请求服务

措施和技术

密码学

古典密码学：每个字母和每个字母的对应//可以通过统计密文频率进行破解
*除了单个字母，字母对和字母组的出现频率也有一定规律
安全性不高

现代密码学：即使完全准确收到信号也无法回复原始信息
香农：**扩散，混淆**
扩散：一位影响很多位
混淆：统计关系变得复杂
对称加密：加密和解密使用同一个密钥 //AES无明显缺点
速度快但是存在管理问题
非对称加密：存在公钥和私钥（只有拥有私钥的人可以解密信息 //RSA 最有影响力
加密解密过程慢

防火墙

一台或多台设备及其结合的软件程序组成，用于加强对计算机的访问控制，所用于内部网和外网之间。只用经过授权的信息才可以通过防火墙
防火墙三大功能：**过滤和管理，保护和隔离，日志和警告**
常见种类：**包过滤防火墙，状态包检查防火墙 ，应用代理防火墙**
包过滤防火墙：最简单的防火墙，通常只包括对源和目的 IP地址及端口进行的检查
工作是规则 表的设置。规则表确定了过 滤规则，过滤系统根据过滤 规则决定是否让数据包通过。 只有满足过滤条件的数据包 才被转发到相应的目的地， 其余数据包则被丢弃。

入侵检测

入侵检测分为3个步骤：信息收集、数据分析和响应
信息收集：包括系统和网络日志文件、目录和文件异常改变、 程序执行异常行为和物理形式入侵信息
数据分析：包括模式匹配、统计分析和完整性分析。前两种 方法都是实时的入侵检测方式，而完整性分析属于事后的分析
响应：即发现有入侵行为之后做出相对应的应对策略