**一、计算机浅谈：**

电子计算机（英语：computer），亦称电脑，是一种利用电子学原理，根据一系列指令对数据进行处理的工具。

在现代，机械计算机的应用已经完全被电子计算机所替换，其所相关的技术研究叫计算机科学，而“计算机技术”指的是将计算机科学的成果应用于工程实践所派生的诸多技术性和经验性成果的总合。“计算机技术”与“计算机科学”是两个相关而又不同的概念，它们的不同在于前者偏重于实践而后者偏重于理论。此外，电子计算机亦被形象地称作电脑。至于由数据为核心的研究则称为信息技术，通常人们接触最多的是个人计算机。

计算机种类繁多，但实际来看，计算机总体上是处理信息的工具。根据图灵机理论，一部具有最基本功能的计算机，应当能够完成任何其它计算机能做的事情。因此，只要不考虑时间和存储因素，从个人数码助理到超级计算机都应该可以完成同样的作业。就是说，即使是设计完全相同的计算机，只要经过相应改装，就应该可以被用于从公司薪金管理到无人驾驶飞船操控在内的各种任务。由于科技的飞速进步，下一代计算机总是在性能上能够显著地超过其前一代，这一现象有时被称作“摩尔定律”。发明于20世纪的电脑以飞跃般的姿态发展，在21世纪的现在，已被应用于手机、电视、数位相机、游戏机等生活用品之中。甚至自动验票机、ATM终端机的控制、顾客管理、财务管理等，各种企业的业务也都仰赖电脑辅助。这些电脑都是由程序进行运作。

计算机在组成上形式不一，早期计算机的体积足有一间房屋的大小，而今天某些嵌入式计算机可能比一副扑克牌还小。当然，即使在今天依然有大量体积庞大的巨型计算机为特别的科学计算或面向大型组织的事务处理需求服务。比较小的，为个人应用而设计的称为微型计算机（Personal Computer，PC），在中国地区简称为“微机”。我们今天在日常使用“计算机”一词时通常也是指此，不过现在计算机最为普遍的应用形式却是嵌入式，嵌入式计算机通常相对简单、体积小，并被用来控制其它设备—无论是飞机、工业机器人还是数码相机[1]。

上述对于电子计算机的定义包括了许多能计算或是只有有限功能的特定用途的设备，然而当说到现代的电子计算机，其最重要的特征是：只要给予正确的指示，任何一部电子计算机都可以模拟其他任何计算机的行为（只受限于其本身的存储容量和执行速度）。据此，现代电子计算机相对于早期的电子计算机也被称为通用型电子计算机。

发展历史

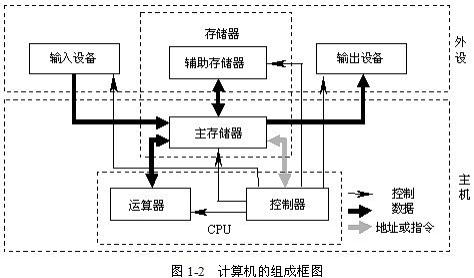
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阶段 | 时期（年） | 主要器件 | 特征 | 应用领域发展 |
| 第一代 | 1946—1958 | 电子管数字机 | 电子管，机器语言，汇编语言，体积大，价格贵 | 军事和科学计算 |
| 第二代 | 1958-1964 | 晶体管数字机 | 操作系统，算法语言，体积缩小 | 工业控制 |
| 第三代 | 1964-1970 | 中小规模集成电路数字机 | 集成电路，可靠性，性能提高 | 文字处理和图形图像处理领域 |
| 第四代 | 1970-至今 | 大规模集成电路机 | 大规模集成电路，并行处理 | 家庭，企业以及各方面应用 |

硬件组成部分

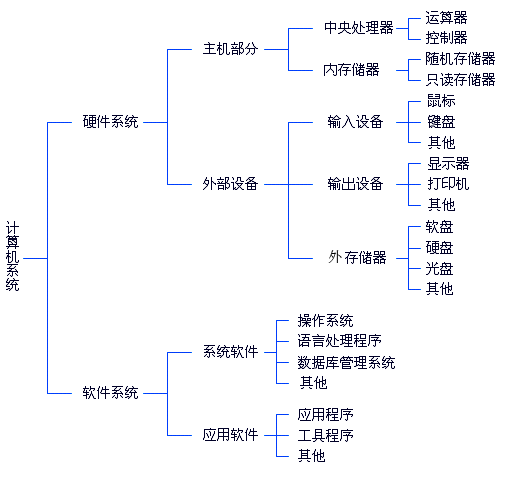
计算机的基本组成:

* **存储器**:　　　　　实现记忆功能的部件用来存放计算程序及参与运算的各种数据
* **运算器**:　　　　　负责数据的算术运算和逻辑运算即数据的加工处理
* **控制器**:　　　　　负责对程序规定的控制信息进行分析,控制并协调输入,输出操作或内存访问
* **输入设备**:　　　　实现计算程序和原始数据的输入
* **输出设备**:　　　　实现计算结果输出

组成的联系:

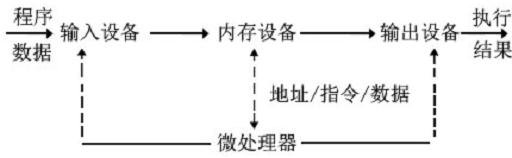


计算机系统的体系结构:



计算机的工作过程:

* 用户打开程序
* 系统把程序代码段和数据段送入计算机的内存
* 控制器从存储器中取指令
* 控制器分析,执行指令,为取下一条指令做准备
* 取下一条指令,分析执行,如此重复操作,直至执行完程序中全部指令,便可获得全部指令



冯·诺依曼机制:

* 程序存储
* 采用2进制

具体描述：

1、运算器（Arithmetical and Logical Unit）：计算机中执行各种算术和逻辑运算操作的部件。运算器的基本操作包括加、减、乘、除四则运算，与、或、非、异或等逻辑操作，以及移位、比较和传送等操作，亦称算术逻辑部件。

2、控制器（Control Unit）：控制器是整个计算机系统的控制中心，它指挥计算机各部分协调地工作，保证计算机按照预先规定的目标和步骤有条不絮地进行操作及处理。

3、存储器（Memory unit）：存储器的主要功能是存储程序和各种数据信息，并能在计算机运行过程中高速、自动地完成程序或数据的存取。

4、输入设备（Input device）：向计算机输入数据和信息的设备，是计算机与用户或其他设备通信的桥梁，是用户和计算机系统之间进行信息交换的主要装置之一。常见的输入设备有，键盘、鼠标、图形输入板、视频摄像机等。

5、输出设备（Output Device）：是计算机硬件系统的终端设备，用于接收计算机数据的输出显示、打印、声音、控制外围设备操作等。也是把各种计算结果数据或信息以数字、字符、图像、声音等形式表现出来。常见的输出设备有显示器、打印机、绘图仪、影像输出系统、语音输出系统、磁记录设备等。

6、操作系统（Operatinig System）：是管理和控制计算机硬件与软件资源的计算机程序，是直接运行在“裸机”伤的最基本的系统软件，任何其他软件都必须在操作系统的支持下才能运行。

7、应用软件（application software）：是和系统软件相对应的，是用户可以使用的各种程序设计语言，以及用各种程序设计语言编制的应用程序的集合，分为应用软件包和用户程序。为满足用户不同领域、不同问题的应用需求而提供的那部分软件。 它可以拓宽计算机系统的应用领域，放大硬件的功能。

**二、linux浅谈：**

Linux的发行版

* Linux 发行版可以大体分为两类： 一类是商业公司维护的发行版；另一类是社区组织维护的发行版；Redhat 是商业公司维护的代表，Debian是社区组织维护的代表；
* Redhat系列目前有三个重要分支：RHEL、 Fedora以及CentOS。RHEL的全称是Redhat Enterprise Linux，中文名为红帽企业版，是一种服务器发行版，它的稳定性非常好，但需要收费；Fedora主要面向桌面发行版，它的内核与应用软件更新非常快， 适合于Linux发烧友使用；CentOS是RHEL的社区克隆版，它通过将RHEL源代码全部重新编译而成，通常在RHEL发布之后的1-3个月发布，其重要优势是完全开源免费。在稳定性方面，RHEL和CentOS的稳定性非常好，适合于服务器使用；Fedora由于软件更新 快，其稳定性较差，最好只用于桌面应用。
* D e b i a n 系 列 分 为 三 个 版 本 分 支 ： S t a b l e 、 Testing和Unstable。其中，Stable一般只用于服务器， 稳定和安全性都非常的高；Testing的版本都经过较 严格的测试，相对稳定；Unstable为最新的测试版 本，其中包括最新的软件包，适合桌面用户。著名的Ubuntu发行版是基于Debian的Unstable版本加强而来，它的界面非常友好，可以说是近乎完美的桌面系统

Linux的哲学思想

* 一切接文件，把几乎所有资源统统抽象成为文件形式，包含硬件设备，甚至是通信接口等。（linux中一切都由根开始，没有windows的盘符概率，只有文件或文件夹之说）
* 尽量避免跟用户交互，其目的在于，易于以编程的方式实现自动化任务。（能实现脚本编程，实现功能自动化运行）
* 使用纯文本文件保存配置信息。（文本编辑器就能完成系统配置工作）
* 由众多单一目的小程序组成：一个程序只实现一个功能，而且要做好，组合小程序完成复杂任务。

Linux系统上命令（几个小命令：

ifconfig、echo、tty、startx、export、pwd、history、shutdown、poweroff、reboot、hwclock、date

）

inux系统上命令的使用格式：

COMMAND [OPTIONS] ARGUMENTS COMMAND:

命令本身是一个可执行的程序文件: 二进制格式的文件，有可能会调用共享库文件；  
  多数程序文件都存放在： /bin, /sbin, /usr/bin; /usr/sbin. /usr/local/bin; /usr/local/sbin  
             普通用户命令： /bin, /usr/bin, /usr/local/bin;  
             管理员命令： /sbin, /usr/sbin, /usr/local/sbin;  
     共享库： /lib, /lib64, /usr/lib, /usr/lib64, /usr/local/lib, /usr/local/lib64.   
              32bits 的库 ： /lib, /usr/lib, /usr/local/lib  
              64bits的库： /lib64, /usr/lib64, /usr/local/lib64.

OPTIONS: (选项）

  指定的命令的运行特性。          
  选项有两种表现形式：  
      短选项：-C, 例如： -l, -d  
          如果同一命令同时使用多个选项， 多数可以合并：-l -d = -ld  
          注意： 有些命令的选项没有"-" ;  
      长选项： --word, 例如：--help, --human-readable.  
      注意：不同的命令的参数： 有些命令可同时带多个参数，多个之间以空白字符分隔。

ifconfig：查看和配置网络接口IP，掩码，网关等信息；

* 1．命令格式： ifconfig [网络设备] [参数]
* 2．命令功能： ifconfig 命令用来查看和配置网络设备。当网络环境发生改变时可通过此命令对网络进行相应的配置。
* 3．命令参数：
  + up 启动指定网络设备/网卡。
  + down 关闭指定网络设备/网卡。该参数可以有效地阻止通过指定接口的IP信息流，如果想永久地关闭一个接口，我们还需要从核心路由表中将该接口的路由信息全部删除。
  + -arp 设置指定网卡是否支持ARP协议。
  + -promisc 设置是否支持网卡的promiscuous模式，如果选择此参数，网卡将接收网络中发给它所有的数据包
  + -allmulti 设置是否支持多播模式，如果选择此参数，网卡将接收网络中所有的多播数据包
  + -a 显示全部接口信息
  + -s 显示摘要信息（类似于 netstat -i）
  + add 给指定网卡配置IPv6地址
  + del 删除指定网卡的IPv6地址
  + <硬件地址> 配置网卡最大的传输单元
  + mtu<字节数> 设置网卡的最大传输单元 (bytes)
  + netmask<子网掩码> 设置网卡的子网掩码
  + tunel 建立隧道
  + dstaddr 设定一个远端地址，建立点对点通信
  + -broadcast<地址> 为指定网卡设置广播协议
  + -pointtopoint<地址> 为网卡设置点对点通讯协议
  + multicast 为网卡设置组播标志
  + address 为网卡设置IPv4地址
  + txqueuelen<长度> 为网卡设置传输列队的长度
* 使用实例：
* 1， 显示活跃的网络设备信息：
* [root@nfs\_server ~]# ifconfig
  + eth0 Link encap:Ethernet HWaddr 00:0C:29:BE:03:E1

  inet addr:192.168.106.128  Bcast:192.168.106.255  Mask:255.255.255.0  
  inet6 addr: fe80::20c:29ff:febe:3e1/64 Scope:Link  
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
  RX packets:502 errors:0 dropped:0 overruns:0 frame:0  
  TX packets:210 errors:0 dropped:0 overruns:0 carrier:0  
  collisions:0 txqueuelen:1000   
  RX bytes:53835 (52.5 KiB)  TX bytes:49147 (47.9 KiB)

* + lo Link encap:Local Loopback

  inet addr:127.0.0.1  Mask:255.0.0.0  
  inet6 addr: ::1/128 Scope:Host  
  UP LOOPBACK RUNNING  MTU:65536  Metric:1  
  RX packets:12 errors:0 dropped:0 overruns:0 frame:0  
  TX packets:12 errors:0 dropped:0 overruns:0 carrier:0  
  collisions:0 txqueuelen:0   
  RX bytes:840 (840.0 b)  TX bytes:840 (840.0 b)

说明：

* + eth0 表示第一块网卡， 其中 HWaddr 表示网卡的物理地址，可以看到目前这个网卡的物理地址(MAC地址）是 00:0C:29:BE:03:E1
  + inet addr 用来表示网卡的IP地址，此网卡的 IP地址是 192.168.106.128，广播地址， Bcast: 192.168.106.255，掩码地址Mask:255.255.255.0
  + lo 是表示主机的回坏地址，这个一般是用来测试一个网络程序，但又不想让局域网或外网的用户能够查看，只能在此台主机上运行和查看所用的网络接口。比如把 HTTPD服务器的指定到回坏地址，在浏览器输入 127.0.0.1 就能看到你所架WEB网站了。但只是您能看得到，局域网的其它主机或用户无从知道。
  + 第一行：连接类型：Ethernet（以太网）HWaddr（硬件mac地址）
  + 第二行：网卡的IP地址、子网、掩码
  + 第三行：网卡的IPV6地址
  + 第四行：UP（代表网卡开启状态）RUNNING（代表网卡的网线被接上）MULTICAST（支持组播）MTU:1500（最大传输单元）：1500字节
  + 第四、五行：接收、发送数据包情况统计
  + 第七行：接收、发送数据字节数统计信息。
* 2、启动和关闭网卡：
  + ifconfig eth0 add 33ffe:3240:800:1005::2/64
  + ifconfig eth0 del 33ffe:3240:800:1005::2/64
  + 输出说明：
  + [root@nfs\_server ~]# ifconfig eth0 192.168.106.128 netmask 255.255.255.0 broadcast 192.168.106.255
  + 说明：ifconfig 配置的ip地址在系统重启后会失效，建议用system-config-network-tui 配置或修改配置文件。
  + [root@nfs\_server ~]# ifconfig eth0 hw ether 00:0C:29:BE:00:BB
  + ifconfig eth0 add 33ffe:3240:800:1005::2/64 为网卡eth0配置IPv6地址；
  + ifconfig eth0 add 33ffe:3240:800:1005::2/64 为网卡eth0删除IPv6地址；
  + 用ifconfig 修改MAC地址：
  + 用ifconfig 配置IP地址
  + [root@nfs\_server ~]# ifconfig eth0 up
  + [root@nfs\_server ~]# ifconfig eth0 down
  + 说明：
  + ifconfig eth0 up 为启动网卡eth0 ；
  + ifconfig eth0 down 为关闭网卡eth0。
  + 为网卡配置和删除IPv6地址
  + 命令：

echo: 回显，命令的功能是在显示器上显示一段文字，一般起到一个提示的作用。

* echo [SHORT-OPTION] ……….[STRING]……………
* -n : 不进行换行。
* -e : 让转义字符生效

**例如：**

* + /bin/bash
  + 如用-e 参数支持的转义字符有如下：
  + If -e is in effect, the following sequences are recognized:
  + \ backslash
  + \a alert (BEL) 发出警告声
  + \b backspace 退格键，表示删除前面一个字符；
  + \c produce no further output 最后不加上换行符号；
  + \e escape
  + \f form feed 换行但光标仍旧停留在原来的位置
  + \n new line 换行
  + \r carriage return
  + \t horizontal tab 水平或横向tab键
  + \v vertical tab 纵向tab键
  + STRING 可以使用引号， 单引号和双引号均可用：
  + 单引号： 强引用， 变量引用不执行替换；
  + 双引号： 弱引用，变量引用会被替换；
  + [root@nfs\_server ~]# echo "$BASH" 双引号例子
  + [root@nfs\_server ~]# echo '$BASH' 单引号例子
  + $BASH
  + 注意： 变量引用的正规符号。
  + ${name}

tty命令用于显示终端机连接标准输入设备的文件名称

* 语法:
  + tty [-s][–help][–version]
  + 参数说明：
  + -s或–silent或–quiet 不显示任何信息，只回传状态代码。
  + –help 在线帮助。
  + –version 显示版本信息。

**实例：**

[root@nfs\_server ~]# tty /dev/pts/0

Startx命令：

* startx：打开一个X（图形）会话
  + ~]#starx &

export命令用于设置或显示环境变量。

* 在shell中执行程序时，shell会提供一组环境变量。export可新增，修改或删除环境变量，供后续执行的程序使用。export的效力仅及于该次登陆操作。
* 命令格式：export [-fnp][变量名称]=[变量设置值] 参数：

-f 　代表[变量名称]中为函数名称。   
-n 　删除指定的变量。变量实际上并未删除，只是不会输出到后续指令的执行环境中。   
-p 　列出所有的shell赋予程序的环境变量。

使用实例： 1、查看环境变量

  [root@nfs\_server ~]# export $PATH

* -bash: export: `/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin'

Pwd 命令：

* pwd 命令来查看”当前工作目录“的完整路径。 简单得说，每当你在终端进行操作时，你都会有一个当前工作目录。 在不太确定当前位置时，就会使用pwd来判定当前目录在文件系统内的确切位置。
* 1、命令格式： Pwd [OPTION]…
* 2、常用参数： 一般情况下不带任何参数, 参数：

-L：目录连接链接时，输出连接路径  
-P：输出物理路径

* 实例：如果目录是链接时：
* 格式：pwd -P 显示出实际路径，而非使用连接（link）路径。
* 目录连接链接时，pwd -P 显示出实际路径，而非使用连接（link）路径；pwd显示的是连接路径
* 命令：
  + pwd -P
  + 输出：
  + [root@nfs\_server ~]# cd /etc/init.d/
  + [root@nfs\_server init.d]# pwd
  + /etc/init.d
  + [root@nfs\_server init.d]# pwd -P
  + /etc/rc.d/init.d

history：主要用于显示历史指令记录内容, 下达历史纪录中的指令。

    命令格式：history [n]  
    参数：  
    n：数字,要列出最近的 n 笔命令列表  
    -c ：将目前的shell中的所有 history 内容全部消除  
    -a ：将目前新增的history 指令新增入 histfiles 中，若没有加 histfiles,则预设写入~/.bash\_history  
    -r ：将 histfiles 的内容读到目前这个 shell 的 history 记忆中  
    -d offest ; 删除指定命令历史；  
    -w ：将目前的 history 记忆内容写入 histfiles。Linux系统当你在shell(控制台)中输入并执行命令时，shell会自动把你的命令记录到历史列表中

* 使用实例：
  + 显示历史中最近的#条命令
  + [root@nfs\_server init.d]# history 3
  + 1015 pwd
  + 1016 pwd -P
  + 1017 history 3
* 调用命令历史列表中的命令：
  + ！# ： 再一次执行历史列表中的第#条命令：
  + ！！ ： 再一次执行上一条命令；
  + ！STRING : 再一次执行命令历史列表中最近一个以STRING 开头的命令。
  + 注意：命令的重复执行有时候需要依赖于幂等性.

Shutdown 命令：

* Shutdown 关机或重启命令
* shutdown [OPTION]… [TIME] [WALL…..]
* 参数:
  + -h : halt将系统的服务停掉后，立即关机
  + -r : reboot在将系统的服务停掉之后就重新启动
  + -c : cancel取消已经在进行的shutdown进程
  + -n：不采用正常程序来关机，用强迫的方式杀掉所有执行中的程序后自行关机
  + -k：不要真的关机，只是发送警告信息出去！
  + TIME:
  + now
  + hh:mm
  + +m 多数分钟
  + WALL 广播通告消息。
* 实例：
  + [root@nfs\_server init.d]# shutdown -h now
  + poweroff 命令：
  + poweroff : 关闭计算机操作系统并且切断系统电源。

reboot 命令：重启

reboot [选项]  
reboot指令用于重启该计算机。它的作用和shutdown -r now是一样的。  
选项：  
-n  重启前保留当前数据  
-f  强制重启计算机  
-w 模拟重启，并将数据写入到”/var/log/wtmp” 日志文件中  
-d  重启，但不写入日志  
-i  重启计算机前关闭网络。

Hwclock 命令： 显示与设定硬件时钟

hwclock [functions] [options]  
在Linux中有硬件时钟与系统时钟等两种时钟。硬件时钟是指主机板上的时钟设备，也就是通常可在BIOS画面设定的时钟。系统时钟则是指kernel中的时钟。当Linux启动时，系统时钟会去读取硬件时钟的设定，之后系统时钟即独立运作。所有Linux相关指令与函数都是读取系统时钟的设定。  
参数：  
-s, --hctosys：以硬件为准，把系统调整为与硬件时间相同；  
-w, --systohc：以系统为准，把硬件时间调整为与系统时钟相同；  
[root@nfs\_server ~]# hwclock –w  
[root@nfs\_server ~]# hwclock   
Wed 21 Sep 2016 12:00:43 PM PDT  -0.204118 seconds

Date 命令 ：

* 1．命令格式：
* date [参数]… [+格式]
* 2．命令功能：
* date 可以用来显示或设定系统的日期与时间。
* 3．命令参数：
* 必要参数:
  + %F 完整日期格式，等价于 %Y-%m-%d
  + %H 小时(以00-23来表示)。
  + %M 分钟(以00-59来表示)。
  + %r 时间(含时分秒，小时以12小时AM/PM来表示)。
  + %s 总秒数。起算时间为1970-01-01 00:00:00 UTC。
  + %S 秒(以本地的惯用法来表示)。
  + %T 时间(含时分秒，小时以24小时制来表示)。
* 2.在设定时间方面：
* date -s //设置当前时间，只有root权限才能设置，其他只能查看。
* date -s 20160921 //设置成20160921，这样会把具体时间设置成空00:00:00
* date -s “20160921 01:01:01″ //这样可以设置全部时间
* 3.加减：
* date +%Y%m%d //显示前天年月日
* date +%Y%m%d –date="+1 day" //显示前一天的日期
* date +%Y%m%d –date="-1 day" //显示后一天的日期
* date +%Y%m%d –date="-1 month" //显示上一月的日期
* date +%Y%m%d –date="+1 month" //显示下一月的日期
* date +%Y%m%d –date="-1 year" //显示前一年的日期
* date +%Y%m%d –date="+1 year" //显示下一年的日期

**实例：**

* [root@nfs\_server ~]# date +%F
  + 2016-09-21
* [root@nfs\_server ~]# date +%Y-%m-%d
  + 2016-09-21
* [root@nfs\_server ~]# date "+%H:%M:%S"
  + 12:52:59
* [root@nfs\_server ~]# date "+%F %H:%M:%S"
  + 2016-09-21 12:53:42

如何在Linux系统上获取命令的帮助信息，请详细列出，并描述man文档的章节是如何划分的。

查询步骤：

1. 命令 –help
2. man 要查询的命令 可以在：后通过“/+关键字” 方式进行搜索; man -k 关键字 搜索包含关键字的帮助文档;
3. info 要查询的命令 可以在：后通过“/+关键字” 方式进行搜索;
4. 绝大多数程序都有相应的帮助文档，并保存在/usr/share/doc文件夹中 Man 文档的章节划分： 手册分为man1-man8。

 man1:用户命令；man2:系统调用；  
 man3:C库调用；man4:设备文件及特殊文件；  
 man5:文件格式：(配置文件格式)； man6:游戏使用帮助；  
 man7:杂项；man8:管理工具及守护进行。  
    帮助手册路径：/usr/share/man    
    语法格式：#man COMMAND  
注：1、并非每个COMMAND在所有章节下都有手册。  
    2、可以通过whatis查看命令存在哪些章节。  
    3、whatis执行过程是查询数据进行的，  
     但是如果有新命令需要手动更新数据库：#makewhatis

Linux发行版的基础目录名称的命名法则及功用规定

目录描述

* /主层次 的根，也是整个文件系统层次结构的根目录
* /bin存放在单用户模式可用的必要命令二进制文件，所有用户都可用，如 cat、ls、cp等等
* /boot存放引导加载程序文件，例如kernels、initrd等
* /dev存放必要的设备文件，例如/dev/null
* /etc存放主机特定的系统级配置文件。其实这里有个关于它名字本身意义上的的争议。在贝尔实验室的UNIX实施文档的早期版本中，/etc表示是“其他（etcetera）目录”，因为从历史上看，这个目录是存放各种不属于其他目录的文件（然而，文件系统目录标准 FSH 限定 /etc 用于存放静态配置文件，这里不该存有二进制文件）。早期文档出版后，这个目录名又重新定义成不同的形式。近期的解释中包含着诸如“可编辑文本配置”或者“额外的工具箱”这样的重定义
* /etc/opt存储着新增包的配置文件 /opt/.
* /etc/sgml存放配置文件，比如 catalogs，用于那些处理SGML(译者注：标准通用标记语言)的软件的配置文件
* /etc/X11X Window 系统11版本的的配置文件
* /etc/xml配置文件，比如catalogs，用于那些处理XML(译者注：可扩展标记语言)的软件的配置文件
* /home用户的主目录，包括保存的文件，个人配置，等等
* /lib/bin/ 和 /sbin/中的二进制文件的必需的库文件
* /lib<架构位数>备用格式的必要的库文件。 这样的目录是可选的，但如果他们存在的话肯定是有需要用到它们的程序
* /media可移动的多媒体(如CD-ROMs)的挂载点。(出现于 FHS-2.3)
* /mnt临时挂载的文件系统
* /opt可选的应用程序软件包
* /proc以文件形式提供进程以及内核信息的虚拟文件系统，在Linux中，对应进程文件系统（procfs ）的挂载点
* /root根用户的主目录
* /sbin必要的系统级二进制文件，比如， init, ip, mount
* /srv系统提供的站点特定数据
* /tmp临时文件 (另见 /var/tmp). 通常在系统重启后删除
* /usr二级层级存储用户的只读数据； 包含(多)用户主要的公共文件以及应用程序
* /usr/bin非必要的命令二进制文件 (在单用户模式中不需要用到的)；用于所有用户
* /usr/include标准的包含文件
* /usr/lib库文件，用于/usr/bin/ 和 /usr/sbin/中的二进制文件
* /usr/lib<架构位数>备用格式库(可选的)
* /usr/local三级层次 用于本地数据，具体到该主机上的。通常会有下一个子目录, 比如, bin/, lib/,share/.
* /usr/local/sbin非必要系统的二进制文件，比如用于不同网络服务的守护进程

http://www.178linux.com/wp-content/plugins/ueditor/ueditor/themes/default/images/spacer.gif