Development on Linux

目录

第一章　Linux系统

操作系统（Operating System, OS）是管理计算机硬件与软件资源的计算机程序，也是计算机系统的内核与基石。操作系统可用于管理与配置内存、决定系统资源供需的优先次序、控制输入/输出设备、操作网络与管理文件系统等基本事务，还能够提供一个让用户与系统交互的操作界面。

20世纪70年代，AT&T公司贝尔实验室开发出Unix操作系统。后来，Berkeley加州分校计算机科学系通过修改扩充Unix，发布BSD系统（Berkeley Software Distribution, BSD）。BSD开创了现代计算机的潮流，率先包含库，以支持互联网协议栈（Stack）、伯克利套接字（Sockets），通过整合套接字与Unix 操作系统文件描述符，用户可以通过网络方便地读写数据，如同直接操作磁盘。20世纪80年代，Unix V与BSD成为两大主流操作系统。1991年，芬兰学生Linus Torvalds开发出Linux操作系统内核，并以GNU通用公共许可证发布，成为自由软件Unix系统，改变了Unix系统的世界格局。

Linux系统的设计理念来自于Unix操作系统，并完全遵从POSIX（Portable Operating System Interface）标准，能够在普通计算机上实现全部的Unix特性，具有多任务、多用户的处理能力。目前，已经发布的Linux版本主要有：RedHat Linux（世界上使用最多），CentOS（Community Enterprises Operating System），Gentoo Linux，Ubuntu Linux（Canonical公司），SuSE（Novell公司）。

1.1　Linux系统框架

Linux系统采用分层结构设计，由硬件交互层、内核层、操作系统接口层和应用层组成。硬件交互层处于Linux结构的底层，由管理外围设备的软件构成，如终端控制器、磁盘控制器、存储设备控制器等，给内核层提供基础。内核层是Linux系统的核心，主要包括四个部分：

- 进程管理：负责进程控制、进程通信、进程调度
- 文件管理：管理文件和目录，包括创建、删除、维护文件
- 内存管理：负责内存储器管理和虚拟存储器管理
- 磁盘管理：负责分配和回收磁盘空间以及磁盘调度

操作系统接口层包括三部分：

- Shell：Linux系统的命令解释器，通过它终端用户可以使用内核提供的系统环境，同时还可以使用它实现程序开发。它是Linux系统一个最突出的优势。
- 系统调用：Linux提供给应用程序的**使用接口**，在应用程序中通过函数调用进入核心，直接使用系统资源。
- 窗口系统：Linux提供给应用程序的**图形接口**，用户可借助于图形接口应用操作系统。

应用层包括终端用户应用和应用程序应用。终端用户通过命令方式或以Shell脚本方式使用系统资源，也可以通过Linux的图形终端方式使用操作系统。应用程序可以通过系统调用方式使用系统资源。

1.2 文件系统

计算机文件系统（File System）是一套实现了数据**存储、分级组织、访问和获取等**操作的抽象数据类型。为方便数据查找及访问，文件系统使用**文件和树形目录的抽象逻辑**概念代替硬盘/光盘等物理设备使用的**物理块（Block）**概念，通过管理（分配与释放）物理设备的存储空间，以一定的组织逻辑利用存储空间保存计算机数据。用户使用文件系统存取数据，只要记住数据文件所在目录及名称，完全不必关心数据在物理设备中存储的具体地址（Block）。

文件系统将存储设备（如硬盘，光盘等）中的存储空间划分成特定大小的物理块，数据则存储在物理块，由文件系统负责将物理块组织成文件和目录，并记录物理块与文件的映射关系及未使用的物理块。文件存储在硬盘中，最小存储单位称作**扇区（Sector）**，每个扇区大小为512字节（约0.5KB），文件系统每次读取多个扇区，常见的有8个扇区，形成一个物理块（约4KB）。文件数据储存在块中，文件系统自然需要储存文件元信息，如创建者、创建日期、文件大小等，这种存储文件元信息的区域称作**索引节点（inode）**。

目前文件系统有很多中，主要可以分成磁盘文件系统，光盘文件系统，闪存文件系统，网络文件系统等。

- 磁盘文件系统：FAT、exFAT、NTFS、HFS、HFS+、ext2、ext3、ext4、ODS-5、btrfs
- 光盘文件系统：ISO9660、UDF
- 闪存文件系统：JFFS2、YAFFS
- 网络文件系统（Network File System, NFS）将远程主机上的分区（目录）经网络挂载到本地系统

目前，每个操作系统（Linux，Windows等）都支持多种文件系统。Linux文件系统使得每个系统用户有独立的文件目录环境和文件访问控制机制，保证了用户文件的安全。它以字符流作为文件的基本结构，实现对多种文件类型的支持，并将对设备的管理以文件管理的方式实现，简化了设备的应用和维护。

1.2.1 用户管理

Linux系统是多用户操作系统，用户分为**系统管理员**与**普通用户**。每个用户在系统中都有唯一的账号（用户名），为用户使用系统的凭证。系统管理员（超级用户）帐号是root，在系统中具有最高权限，主要负责系统管理工作。Linux系统中多个用户可以组成一个用户组，同一用户组的用户享有用户组的权限。用户组根据成员构成分为系统管理组和普通用户组，在系统管理组中，每个成员都是系统管理员。无论是用户还是用户组，系统都会分配一个唯一的识别码。用户识别码为UID（超极用户UID约定为0），用户组识别码为GID。

所有用户均通过用户帐号和密码进入操作系统，而进入系统的方式可以是本地直接进入，也可以通过远程登录进入。在用户成功进入系统后，系统终端提示符为$或%，如果用户是超级用户，则提示符为#。Linux系统支持一个用户远程登录会占用大约1MB的内存，用户退出系统不但可以回收占用的内存，还可以避免系统记帐日志继续记录，或者用户帐号为他人利用，发生用户文件破坏等现象。

用户管理是Linux系统管理的一个重要部分，系统管理员通过管理用户或用户组，实现对系统的访问控制。Linux系统中常用的用户管理文件有：

- /etc/passwd：认证系统用户访问权限的第一个文件，记录用户的基本信息
- /etc/shadow：管理用户密码，主要用于系统管理员修改、取消用户密码
- /etc/group：管理用户组

1.2.2 文件类型

文件和目录是文件系统的基本元素，文件系统最基本功能是让用户可以访问文件和目录，并执行操作。在Linux文件系统中，文件主要分成如下几类：

- 标准文件：用户使用最普遍的文件类型，可以存储任何类型的数据。文件内容可按文本、应用程序指定格式、系统可执行的二进制方式存储。Linux系统中使用**-**表示此文件类型。

- 目录文件：将多个相关文件放入同一个目录中以方便文件管理。Linux系统中使用**d**表示此文件类型。
- 软链接文件：使用文件中的链接指针指向其他文件或目录。软链接是一种文件共享机制，实现多个文件指向同一个物理文件。它可以实现本地文件共享，也可以实现网络多主机文件共享。它的缺点是每次访问磁盘时要读盘多次，增加访问磁盘的频率与访问文件的时间消耗。Linux系统用**l**表示该文件类型。
- 管道文件：使用一个文件作为管道，实现进程间的通信。发送进程将发送的数据写入管道文件，接收进程从管道文件读取接收的数据。写入内容并读出内容时总是按照先写入先读出的顺序。管道文件存储时使用直接块而非间接块的形式。Linux系统用**p**表示该文件类型。
- 特殊文件：外围设备管理文件。Linux系统中字符设备文件用**c**表示文件类型，块设备文件用**b**表示文件类型。
- Socket文件：用于网络TCP接口通信的系统程序。Linux系统用**s**表示该文件类型。

Linux文件结构采用字符流的形式，将所有文件中的内容都看作由字符构成，把文件看作是流式文件。Linux文件系统由此能兼容多种文件系统。

硬链接与软链接

Linux系统中的文件不仅包括实际内容外，还包括各种属性，如访问权限、拥有者、群组、时间参数等。文件系统将文件数据划分成三个区块：实际数据存储到数据块（Data Block），索引节点（Inode）存储文件属性与Block号，文件系统的整体信息（inode与数据块的空间使用信息）存储到超极区块（Superblock）。索引节点号（Inode Index）是文件的唯一标识，系统可以通过它定位文件数据块。

为解决文件共享问题，Linux系统引入链接：**硬链接（Hard Link）**与**软链接（又称符号链接，Soft Link/Symbolic Link）**。链接不仅解决了文件共享问题，还带来了隐藏文件路径、增加权限安全、节省存储空间等好处。如果多个文件名指向一个索引节点号，则称这些文件为硬链接。硬连接的作用是允许一个文件拥有多个有效路径名，那么用户可以建立硬连接到重要文件，防止误删发生。由于硬链接是文件名不同但inode相同的文件，存在如下几点特性：

- 硬链接拥有相同的inode与文件数据块
- 只能对文件，不能对目录创建硬链接
- 只能对已经存在的文件创建硬链接
- 不能跨文件系统创建硬链接
- 删除一个硬链接不影响其他具有相同inode的文件，如果删除原文件则不会影响硬链接下的文件，但链接不复存在

如果一个文件数据块存放的是另一文件的引用，那么该文件就是软链接。软链接与硬链接不同，它具有如下特征：

- 软链接有自己的文件属性及权限等
- 可对不存在的文件或目录创建软链接
- 可以跨文件系统创建软链接
- 删除软链接不影响其指向的文件（原文件），如果原文件被删除，则软链接称为死链接（Dangling Link），如果原文件重建，则死链接恢复为正常的软链接

无论是硬链接还是软链接，只要链接关系存在，则在一个文件修改内容，其他所有相链接的文件都会发生改变。

1.2.3 文件所有者与权限

Linux系统是安全的操作系统，用户不仅需要认证进入系统，在进入操作系统后还需要拥有相应的权限才能访问、修改与执行文件和目录的相应操作。在Linux系统中，创建文件、目录的用户是文件、目录的所有者。系统为文件、目录设定三组访问权限：文件所有者（u）权限、文件所有者用户组（g）权限、其他用户（o）权限。访问权限分四种：可读（r）、可写（w）、可执行（x）和无权限（-）。

使用ls -l可以看到文件、目录的权限信息。文件、目录的权限字段由10个字符组成，比如

$$-rwxr-xr-x$$

第1列为文件类型，2 – 4列红色区域是文件所有者权限（u），5 – 7列蓝色区域表示文件所有者所在用户组权限（g），8 – 10列青色区域代表其他用户权限（o），2 – 10 列权限总称a权限。根据权限信息可知，此文件的所有者具有可读、可写和可执行权限（rwx），对于文件所有者所在组成员和其他组外成员均具有读、可执行的权限（r – xr – x）。如果将文件、目录三组访问权限用二进制字串表示，则rwx可以表示成111，十进制数值等于7，r-x（101）则等于5。

Linux系统允许超极用户或者已经获取超极权限的普通用户，使用chmod命令改变文件、目录的访问权限，使用chown命令改变文件、目录的所有者，使用chgrp命令改变文件、目录所有者所在的用户组。

1.3 磁盘与文件管理

一个磁盘可以划分成多个分区，每个分区必须格式化（Format）为特定文件系统以后才能正常使用。格式化会在磁盘上写入一些管理存储布局的信息。Linux文件系统是标准的树型目录结构，最上层是根目录，其他所有目录都是从根目录出发生成。微软的DOS和Windows系统也是采用树型结构，但是它们的树型结构的根目录是磁盘分区的盘符，有几个分区就有几个树型结构，多个树型结构并列存在；对于Linux，无论操作系统管理几个磁盘分区，只有一个树型结构的目录。在Linux系统安装后，自动为用户创建文件系统，创建功能明确、完整固定的目录结构，方便系统文件、用户文件的统一管理。

1.3.1 目录结构

- **/** 文件系统根目录，包含系统中其他所有文件目录，它也是系统挂载的第一个文件目录
- **/boot** 存放启动Linux时需要的核心文件，如操作系统内核、系统引导加载程序等
- **/home** 系统默认的所有用户的主目录，比如用户john的主目录就分配到/home/john
- **/etc** 系统配置文件目录，包含存储系统用户账号信息的/etc/passwd文件，系统启动时的配置文件/etc/rc.d
- **/bin** 存放系统用户使用的可执行二进制程序与基本命令，不含子目录
- **/sbin** 存放系统管理员使用的可执行二进制程序，用于分区、文件系统的创建、守护进程的安装等
- **/lib，/lib64** 系统程序标准库，也称动态链接共享库，保证/bin与/sbin的正常运行，部分子目录安放系统内核使用的模块
- **/usr** Unix软件资源**Unix Software Resources**目录，其结构与根目录相似，不同之处在于根目录中文件多是系统级文件，而/usr 存放的都是用户级文件，一般与具体系统无关。除安装、卸载软件之外，一般无需修改/usr的内容。在系统正常运行时，甚至/usr可以只读挂载，由于这一特性，/usr常被划分到单独的分区，甚至多台计算机共享一个/usr
- **/var** 用于存放变量或临时数据：临时文件、日志文件、缓存文件，安装包管理数据库等。
- **/proc** 存在于内存中的虚拟文件系统，存放内核、进程、周围设备和网络状态信息，由于数据都在内存中，它本身不占用硬盘空间
- **/sys** 存放硬件设备的驱动程序信息

- **/mnt** 临时设备挂载点，也称挂载目录，用于挂载其他文件系统
- **/media** 可移动设备挂载点，通常将U盘等设备自动挂载到此目录
- **cdrom** CD-ROM的挂载点，它不符合Linux文件系统层级标准（Filesystem Hierarchy Standard, FHS），根据标准CD-ROM应该被挂载到/media 目录下
- **/dev** 系统设备文件目录，目录包含各种系统设备，如CD-ROM、磁盘驱动器、调制解调器等。大致可以分成两类：用于保存数据的块设备（Block Device）和用户传输数据的字符设备（Character Device）
- **/run** 存放应用程序运行时需要的数据，如进程ID、Socket信息、锁文件等
- **/tmp** 存放公用临时文件，重启则清空
- **/opt** 第三方软件默认安装目录，如Adobe Reader等
- **/srv** 服务目录，存放特定站点数据，如FTP、RSYNC、WWW、CVS等协议
- **/lost+found** 它并非Linux目录结构的组成部分，而是ext3文件系统用于存放

1.3.2 配置文件

Linux系统包含各种类型的配置文件，用于系统初始化、文件系统管理、用户管理、Shell配置、系统环境设置与网络配置，大多数配置文件都存放在/etc/目录下。

系统初始化

- /etc/init：运行级别、控制台数量
- /etc/timezone：时区

文件系统管理

- /etc/fstab：开机时挂载的文件系统
- /etc/mtab：当前挂载的文件系统

用户管理

- /etc/passwd：用户信息
- /etc/shadow：用户密码
- /etc/group：群组信息
- /etc/gshadow：群组密码
- /etc/sudoers：Sudoer列表

Shell配置

- /etc/shell：可用于Shell列表
- /etc/inputrc：ReadLine控件设定
- /etc/profile：用户首选项
- /etc/bash.bashrc：bash配置文件

系统环境设置

- /etc/environment：环境变量
- /etc/updatedb.conf：文件检索数据库配置信息
- /etc/issue：发行信息
- /etc/screenrc：屏幕设定

网络配置

- /etc/iftab：网卡MAC地址绑定
- /etc/hosts：主机列表
- /etc/hostname：主机名
- /etc/resolv.conf：域名解析服务器地址
- /etc/network/interfaces：网卡配置文件

1.4 进程管理

1.4.1 系统引导

在计算机电源打开后，Linux系统进入引导过程（Boot），需要遍历**基本输入输出系统（Basic Input/Output System, BIOS）**，**引导装载程序（Boot Loader）**和**系统内核（Kernel）**。在计算机CPU运行BIOS自检（Self Test）程序以后，随即从磁盘中读取系统引导装载程序Linux Loader（LILO）或Grand Unified Boot Loader（GRUB）并执行系统引导装载程序开启引导过程，将Linux内核可执行代码写入内存，并初始化硬件设备，创建存储器空间的映射图。在核心程序装载完毕以后，系统开始执行系统核心代码，获取CPU的控制权。内核启动后即运行**调度程序（Scheduler）**实现多任务（multi-tasking）处理机制，并执行第一个用户空间（User Space）上的程序/sbin/init，产生一个编号PID等于1的进程（init进程）。系统运行初始化程序生成一系列初始进程，并读取配置文件/etc/inittab 设置的系统运行级（Run Level），设置系统环境，启动各种后台服务进程（守护进程），等待用户登录。

1.4.2 进程

进程（Process）是处于运行中**计算机程序（Computer Program）**的实例。所谓计算机程序，就是一组为完成特定任务的指令序列，它是静态的，本身没有任何运行的含义，进程则是真正运行这些指令的对象。在用户下达运行程序的指令后，进程随即诞生。多个进程可以与同一个程序相关联，并以同步（Synchronous，循环）或异步（Asynchronous，平行）的方式独立运行。进程是系统进行资源分配的基本单位，一个进程包括虚拟地址空间、可执行代码、关联数据（如变量，内存，缓冲区等）和其他操作系统资源（如进程创建的文件、管道、同步对象等）。现代计算机系统可以在一段时间内将多个程序加载到存储器中，通过分时复用（Time Sharing）机制，在不同程序间进行切换，在单个CPU上营造出同时（Simultaneous，平行性）运行的假象。

在Linux系统中，大多数进程（除init进程）都是其他进程通过调用系统函数fork创建的，并称调用fork函数的进程为**父进程**，新创建的进程为**子进程**。实际上，init进程是Linux系统启动后创建的第一个进程，其他所有进程均是它通过fork创建的子进程。有赖于fork函数，每个进程都可以关联多个子进程。

操作系统维护一个进程表，维护系统中所有进程的基本数据，如进程标识符（PID）、父进程标志符（PPID）、分配的内存、环境变量、资源使用情况等。当进程结束运行（调用exit函数、运行时发生致命错误或收到终止信号所致）时，子进程的退出状态（返回值）会反馈给操作系统，系统维持子进程的进程控制块（Process Controlling Block, PCB）并将子进程结束运行的SIGCHLD信号发送给其父进程。默认地，父进程在接收到SIGCHLD信号以后，会立即调用wait函数获取子进程的退出状态，继而内核可以从内存中释放已结束运行的子进程PCB。但是，如果父进程未及

时处理，迟迟没有调用wait函数获取子进程退出的状态，则已结束运行的子进程PCB将会在内存常驻，成为所谓的**僵尸进程（Zombie Process）**。通俗地讲，僵尸进程俨然是“白发人送黑发人”的结局，父进程得知噩耗（SIGCHLD）后悲痛至极，以致子进程的后事（PCB）无法即时料理。如果父进程先于子进程结束运行，则子进程就成为所谓的**孤儿进程（Orphan Process）**。在类Unix系统中，孤儿进程一般会由init进程所收养，成为init的子进程。

系统与用户进行交流的界面称为**终端（Terminal）**，从终端启动的每个进程都依附于它，则称终端是依附进程的**控制终端（Controlling Terminal）**，当控制终端关闭时依附进程都会自动关闭。为了能够突破这种“一荣俱荣一损俱损”的限制，在终端关闭以后还能够在系统中持续存活，系统生成了一类特殊的进程，称为**后台服务进程或守护进程（Daemon）**。守护进程的生命周期较长，在开机内核引导装入时启动，在系统关闭时终止。它独立于控制终端，周期性地执行某种任务或等待处理某些发生的事件（如作业规划进程crond、打印进程lqd等，结尾字母d是Daemon的首字），几乎所有的服务器程序都以守护进程的形式出现。

Linux允许在不同的场合，分配不同的开机启动程序，称作“运行级别”（Run Level）。Linux预置10 种运行级别（0 – 9），运行级别0 表示关机（System Halt），级别1为单用户模式（也称维护模式），级别3是多用户模式，级别6为重启（System Reboot）。

1.4.3 线程

线程（Thread）存在于线程之中，它是操作系统能够调度的基本单位，也是进程中的实际运作单位。通常一个进程可以包含多个线程，不同线程并发执行不同的任务。线程只拥有在运行中必不可少的资源，如程序计数器、寄存器和栈，但是可以与同属一个进程的其他线程共享进程所拥有的全部系统资源，如内存、虚拟地址空间、公共变量等。

1.5 Linux系统命令

Linux系统命令分**内部命令**和**外部命令**两种，内部命令由Shell程序实现，如cd、echo等，数量有限。每个Linux外部命令都是一个应用程序，如ls、cp等绝大多数命令都属于外部命令，它们以可执行文件的形式存在，绝大部分放在目录/bin和/sbin中。

1.5.1 帮助命令

在Linux终端连续键入**两个TAB键**，系统指令超过2000多，我们可以利用帮助文档获取大量的指令及选项（OPTIONs）使用信息。查询终端命令的帮助文档有三种用法：（1）cmd - -help （2）man cmd（3）info cmd。相对于第一种用法，后两种提供更加详细的信息帮助信息。

在帮助文档指令语法介绍中，中括号表示参数可选，省略号表示可同时使用多个选项，分割号则表示“或”从多个选项中选择一个使用。在许多Linux指令中，存在短选项（单个字母）与长选项（单词）两种，如ls -a 与ls - -all，短选项单个横杠，长选项双横杠。在帮助文档的首行，查询指令随后跟着一个代码（1-8）表示指令的类型，见图1.1。对于篇幅较长的帮助文档，上下翻页可以使用上下键，也可以键入长空格键向下翻页，键入 q 返回终端，键入 g 跳转到文档首行，键入 G 跳转到文档尾行。执行/keywords则在帮助文档向下搜索关键词，执行?keywords则是向上搜索关键词。帮助文档文本放在/usr/share/doc。

1.5.2 系统管理指令

- su,sudo
  切换系统用户（switch user），无参情况下直接切换至超级用户，也可以切换至其他普通用户：su user。从系统安全性角度，使用su可能会给系统带来风险，替换方案是使用sudo。sudo无需超级用户账号口令就能拥有超级用户权限。正常使用sudo指令，需要root用户预先修改/etc/sudoers 文件。由于配置文件语法特殊，需要使用visudo执行编辑。普通用户使用sudo指令获取超极权限有个时间上限，前后两次执行时间间隔超过上限则需要用户重新输入密码。此外，我们还可以找到Defaults env_reset，添加timestamp_timeout=x，若x=-1则限制此项功能。用户使用exit，或logout 可以终止顶级权限的使用。

代号	代表内容
1	使用者在shell环境中可以操作的命令或可运行文件
2	系统核心可呼叫的函数与工具等
3	一些常用的函数(function)与函式库(library)，大部分为C的函式库(libc)
4	装置文件的说明，通常在/dev下的文件
5	配置文件或者是某些文件的格式
6	游戏(games)
7	惯例与协议等，例如Linux文件系统、网络协议、ASCII code等等的说明
8	系统管理员可用的管理命令
9	跟kernel有关的文件

图 1.1: 文档代码与类型映射表

- chown,chgrp
  变更文件或目录的所有者，群组

- chmod
  修改文件权限的两种表示法：**数字表示法**、**文本表示法**。数字表示法，比如chmod 764 file，对文件file用户开放所有权限（rwx），对于文件所有者所属用户组只能读（rw-），而对于其他用户则只能执行（r–）。文本表示法用4个字母表示不同的用户，用户（u）、用户组（g）、其他成员（o），所有人（a），在原始权限的基础上增加（+）、减少（-）或修改权限（=），如chmod u-x,g-w,o=x file，则文件file的减少用户的执行权限，减少用户组的写权限，设置其他用户的权限为执行权限。修改目录权限与之类似，但相应的权限表示不同的意思，如r-可列出目录中的文件，w-可在目录中创建、删除和修改文件，x-可以使用cd命令切换到此目录，如chmod 666 /dir/*放开目录dir下的所有文件的rwx权限

- adduser,addgroup
  向系统添加新用户或用户组。

- passwd
  修改特定用户口令，普通用户可借之修改账号口令，超极用户可以用它修改系统中所有用户口令，如passwd username修改用户username的口令。

- shutdown,halt,reboot,poweroff
  关机、重启命令，比如立即关机shutdown now，定时20:00关机shutdown 20:00，10分钟后关机并提示shutdown +10 “shutdown”，立即关机并重启shutdown -r now，立即关机并断电shutdown -P now。安全关机模式halt，并在关机前将信息写入/var/log/wtmp，系统断电poweroff，重启系统reboot

- login,logout
  用户登陆,退出终端

1.5.3 进程管理指令

- ps
  查看当前系统中运行的进程信息

- top
  监视系统当前运行的各进程CPU内存等利用情况等

- kill,killall
  终止进程，终止进程之前可以通过top指令查看当前所有进程及其PID，利用PID终止进程

1.5.4 网络通讯指令

- rlogin,rsh
  远程登录，远程运行程序

- telnet,ssh
  远程登录，用法：ssh username@hostname，telnet hostname，hostname是要登陆系统的域名或IP地址。

- ftp, lftp
  用于客户端和服务器之间上传下载数据，后者更为强大，还可以处理http协议

- wget, curl
  支持网络上传下载，可以批量下载及镜像网站，并可以模拟网页点击，进行网络登录后的操作，比如设置网络通等。wget是一个命令行下载工具，支持通过HTTP、HTTPS、FTP三个最常见的TCP/IP协议下载，并可以使用HTTP代理。如wget -P /dir url将目标文件下载并存储到/dir。

- axel
  支持并发多线程同时从多个或单个服务器下载，如两个线程下载:axel n 2 url

- nslookup
  查询DNS记录，检查域名解析是否正常，在网络出现故障时用于诊断问题

- ping
  通常用来测试与目标主机的连通性，在发送ICMP ECHO_REQUEST数据包到网络主机后，根据响应信息，判断目标主机是否可访问（并非绝对）。通过防火墙禁止ping或者在内核参数中禁止ping，可以防止通过ping探测服务器的开启状态。Linux下的ping和Windows下的ping稍有区别，前者不会自动终止，需要按CTRL+c终止或者用参数-c指定完成的回应次数。用法：ping [options] destination

- traceroute
  追踪网络路由，判断网络出现问题的地方

- finger
  获得网络中其他用户的信息（如最后登陆的时间、使用的Shell类型、主目录路径等）

1.5.5 系统配置指令

- export
  将自定义变量转成环境变量，修改后的环境变量仅仅对本次登陆有效。使用echo $ var可以读取变量var

- unset
  取消变量

1.5.6 磁盘与文件管理指令

- cd
  切换目录，让用户在不同目录之间进行切换，前提是用户拥有进入目录的权限。用法：cd directory，如cd / 切换到根目录，cd回到用户的home目录

- cp
  复制文件和目录
- ls
  显示目录中的文件，ls -l显示文件详细格式列表，ls -a显示所有文件或目录，ls /dir/v*列出子目录中以字母v打头的全部非隐藏文件。参数-R递归处理，-t按照时间排序，-S按照大小排序，-r逆向排序
- mkdir,rmdir
  创建用户工作目录，删除**空**目录
- mv
  移动文件或者文件重命名
- rm
  删除文件或者目录，如rm -rf file强制删除非空目录-r向下递归，子目录一并删除；-f强制删除
- pwd
  显示当前工作目录
- ln
  建立硬连接和符号连接：**硬连接**是一个文件的额外名字，相当于一个同步更新的副本，删除源文件，硬连接的内容还存在；**符号连接**相当于快捷方式，当源文件被删除后，符号连接仍然存在，但链接的内容已经不存在
- find,locate,whereis,which
  搜索指定目录下具有某种特征的文件。一般来说，find命令功能最强大，直接读硬盘对硬件的损耗也最大。locate与whereis是通过系统数据库查找文件。
- df
  报告挂载的文件系统名称、硬盘空间使用统计、挂载点
- du
  报告文件、目录硬盘空间使用状况
- uname
  显示当前操作系统关键信息
- who
  查看当前登录到系统的用户信息
- date
  显示或设置此时系统的时间
- clear
  清空终端屏幕
- mount,unmount
  在Linux中，如果需要使用一个存储设备（如硬盘、光驱等device），第一步就得将其挂载（mount）到文件树上，以作为一个文件目录直接访问。使用mount指令挂载设备时，需要至少三个参数：挂载对象的文件系统类型、挂载对象的设备名称、挂载点。mount指令的使用方法：mount [option(s)] [<device>] mountpoint。执行cat /proc/filesystems可以查询系统支持的文件系统，/dev目录保存系统中所有的存储设备名称，/mnt是专门用作挂载点的目录。unmount则是将存储设备从文件数上取下。每次开机时，Linux会自动将需要挂载的Linux分区挂载上，/etc/fstab文件就列出了开机时自动挂载的文件系统列表

1.5.7 访问文件内容

- echo
  echo “xxx”>>file将文本内容追加到指定文件，echo “xxx”>file将文本内容写入指定文件（覆盖）
- cat
  显示文件内容、创建新文件或合并文件内容成一个文件
- tac
  由最后一行开始反向显示文件内容
- head, tail
  显示文件前几行与后几行
- od
  读取非文本格式文件的内容
- grep,fgrep
  grep利用正则表达式执行搜索，fgrep搜索固定字符串
- diff
  比较文件或目录
- wc
  统计文件字数行数等数据

1.5.8 文本编辑工具

vi（visual interface）是一种全屏幕编辑器，最初由加州大学Berkeley分校为BSD系统开发，后作为标准包含在所有Linux版本中。受到图形显示及键盘功能（当时键盘无功能键）的限制，最初vi没有提供鼠标的使用，只能使用键盘上的字母、数字、标点符号和ESC键完成文件编辑。时至今日，vi仍然是Linux系统程序员及管理员最喜欢的编辑器之一。vi窗口全屏只能显示20行内容，可上下移动窗口，浏览文件全部内容。vi编辑的文件大小也有限制，最大行数不超过25,000，每行最多1,024个字符。目前，Linux系统常用的一个文本编辑工具是从vi发展出来的vim（vi iMproved），与Emacs并列成为类Unix系统用户最喜欢的编辑器。为帮助初学者学习vim，Linux系统通过命令vimtutor可以访问详细的帮助文档。

vim包含三种模式：命令行模式（Command Mode）、插入模式（Insert Mode）与尾行模式（Last-line Mode）。用户可以通过执行vim file，进入命令行模式的全屏编辑界面。使用按键**i**、**a**、**o**都可以切换到插入模式，并在终端左下角标示**INSERT**表明可以执行编辑。在插入模式使用按键**ESC** 可以切换回命令行模式。在命令行模式使用冒号命令**:**可以切换到尾行模式。

剪切并粘贴：移动光标到文本块的起始部分，输入指令md，移动光标到文本块的结束位置，输入指令d’d，移动光标到目标粘贴位置，输入指令P或p。

复制并粘贴：移动光标到文本块的起始部分，输入指令my，移动光标到文本块的结束位置，输入命令y’y，移动光标到目标粘贴位置，输入指令P或p。

1.5.9 打包与解包，压缩与解压缩

TAR

Linux的应用程序tar最初是为了制作磁带存档而设计的：把文件和目录复制到磁带中，然后从存档中提取或恢复文件。现在已经可用于任何设备，也是数据备份中最常用的命令之一。打包结果（.tar）称作TARBALL。生成TARBALL后，就可以用其它程序（zip/gzip/bzip2）进行压缩。比如，tar -cf dest.tar *.txt是将目录中所有的文本文件打包，tar -xf dest.tar对文件dest.tar解包。选项f几乎所有的指令都有，c表示创建（create）函数，x表示抽取（eXtract）

命令	说明
gg	移动到文件的第一行
G	移动到文件的最后一行
yy	复制当前行的内容至粘贴板
P	将粘贴板上的内容粘贴至光标所在行的行首
p	将粘贴板上的内容粘贴至光标所在行的行尾
dd	删除整行
J	删除当前行的换行符
u	撤销最近一个编辑操作
U	撤销在一行上的编辑操作
:s/a/b/	替换**本行**第一个a为b
%s/a/b/	替换**每行**第一个a为b
:s/a/b/g	替换**本行**所有的a为b
%s/a/b/g	替换**每行**的a为b
:q	未修改文件即退出
:q!	强制退出不保存
ZZ	退出并保存，也可以使用冒号命令:wq

表 1.1: vim命令

函数。tar还提供了一种特殊的功能：打包/解包的同时压缩/解压，tar -czf dest.tar.gz *.txt或者tar -cjf dest.tar.bz2 *.txt打包的同时进行压缩（使用**gzip**或**bzip2**），tar -xzf source.tar.gz或者tar -xjf source.tar.bz2解包的同时进行解压缩（使用**gunzip** 或**bunzip2**）。**tar.gz文件与tgz文件一致**。

GZIP，GUNZIP

gzip是压缩程序，gunzip是解压程序

BZIP2，BUNZIP2

bzip2是压缩程序，bunzip2是解压程序

ZIP，UNZIP

zip是压缩程序，unzip是解压程序

RAR

Linux下处理.rar文件，需要安装RAR for Linux

1.5.10 软件包下载与安装

通常Linux应用软件安装包有：（1）**TAR包**，如software-1.2.3-1.**tar.gz**，使用UNIX下的打包工具TAR封装。（2）**RPM包**，如software-1.2.3-1.i386.**rpm**，使用Redhat Linux 提供的一种包封装格式。（3）**DEB包**，如software-1.2.3-1.**deb**，使用Debain Linux提供的一种包封装格式。TAR（Tape ARchive）出现在还没有软盘驱动器、硬盘和光盘驱动器的计算机早期阶段。大多数Linux应用软件包的命名遵循一定的规律：名称-版本-修正版-类型。

在GNU/Linux操作系统中，软件包管理工具提供软件安装，升级，卸载，以及软件状态信息查询功能，是一种十分重要的工具。流行的软件包管理工具RPM（Redhat Package Manager）是由RedHat公司推出；DPKG（Debian PacKaGe）是Debian操作系统（如Ubuntu）的软件包管理工具；APT（Advanced Package Tool）属于Debian及其衍生版

本的软件包管理工具，可自动下载、配置和安装二进制或源代码格式的软件包；apt-get是一个简单的下载安装软件包命令行接口，最常用的功能是软件更新与安装。一般地，软件包管理工具都需要root权限执行命令。

软件包下载

软件包管理命令行程序**apt-get**是建立在APT (Advanced Packaging Tool)库上的，用于安装新软件包、移除与更新已经安装的软件包，甚至可用于更新整个操作系统。**apt-cache**命令行工具则用于搜索APT软件包缓存，简单而言用于搜集软件包信息。apt-cache pkgnames列出所有的软件包资源，apt-cache search pkg搜索指定软件包pkg，apt-cache showpkg pkg显示指定软件包pkg的依赖包，apt-cache stats关于软件包列表的统计数据。sudo apt-get update根据/etc/apt/sources.list更新系统中的软件包，sudo apt-get install pkgname1 pkgname2更新并安装包pkgname1与pkgname2，sudo apt-get remove pkgname移除安装包pkgname，sudo apt-get purge pkgname彻底清除pkgname（包括其配置文件）。

我们还可以利用add-apt-repository（python脚本）向/etc/apt/sources.list添加PPA（Personal Package Archive）源，允许用户建立自己的软件仓库，自由的上传软件。用法：sudo add-apt-repository ppa:user/ppa-name

软件包安装

每个应用程序软件包中通常包含两种常见的文件：（1）**可执行文件(.bin)**，解开包就可以直接运行。（2）**源程序(.src)**，解开包后还需要将其编译成可执行文件。通常用TAR打包的，都含源程序，用RPM、DPKG打包的只有可执行程序。软件包安装之前首先要进行解压缩，根据INSTALL/README文件进行相应的设置或直接进行安装，如果软件包只有源程序，可以根据下面几个步骤进行安装：

- 执行./configure进行编译配置
- 执行make进行编译
- 执行make install完成安装
- 执行make clean删除安装产生的临时文件

1.5.11 编译指令

- cc,gcc
- javac,java

1.5.12 常用快捷键

- CTRL+c 停止执行命令
- CTRL+d 结束传输或屏幕输入
- CTRL+s 临时停止输出
- CTRL+q 恢复输出
- CTRL+u 擦除光标以前的内容
- CTRL+k 擦除光标以后的内容
- CTRL+r 在以前的命令中搜索
- Alt+F2 在GNOME中搜索终端（TERMINAL）或SHELL
- BACKSPACE 纠正错误
- TAB 补充文件名或命令

1.6 Linux应用程序安装与配置

1.6.1 创建启动器（Launcher）

创建桌面快捷方式/启动器（Launcher）：启动器实际上是桌面配置文件，desktop后缀，多放在/usr/share/applications/，可以从Terminal添加。本节介绍如何将Eclipse 设置为启动器。

（1） 进入桌面：cd Desktop

（2） 模仿example.desktop，创建桌面配置文件eclipse.desktop：sudo vim eclipse.desktop

[Desktop Entry]

Name=eclipse

Comment=eclipse

Exec=/home/chunheng/Develop/Java/eclipse/eclipse

Icon=/home/chunheng/Develop/Java/eclipse/icon.xpm

Terminal=false

Type=Application

Categories=Application;Development;

Encoding=UTF-8

StartupNotify=true

（3） 为新添加的桌面文件赋予一定权限：sudo chmod 777 eclipse.desktop。如果Eclipse是解压安装，则执行sudo chmod u+x eclipse.desktop

1.6.2 中文输入法

Ubuntu自动安装的输入法是IBUS输入法，本节介绍搜狗输入法的安装（默认使用CTRL+长空格键进行中英文切换，也可以重新设置组合键）：

apt-get remove fcitx*	1、如果以前安装过fcitx，将其删除
apt-add-repository ppa:fcitx-team/nightly	2、添加源
apt-get update	3、更新源
apt-get install fcitx-sogoupinyin	4、安装搜狗拼音输入法
reboot	5、重启系统

表 1.2: 安装搜索输入法基本步骤

1.7 Linux下的Java开发

1.7.1 Java开发环境

- 从Oracle官网下载SE JDK（Java Development Kit）二进制安装包，如jdk-6u22-linux-i586.bin
- 将安装包拷贝到用户主目录/home/chunheng/并执行安装sudo sh jdk-6u22-linux-i586.bin
- 打开隐藏文件.bashrc，添加如下脚本设置环境变量JAVA_HOME：

  export JAVA_HOME=/home/username/java/jdk1.6.0_22

  export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH

  export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

- 测试安装是否成功，重启系统并运行java -version显示安装的JDK版本

测试Java程序的编译及运行：

- 创建源代码：在用户主目录下创建新目录并编写Java源代码，如Test.java
- 编译源代码：在源代码所在目录下执行编译javac Test.java
- 运行程序：java Test

1.7.2 线程

Java程序中线程都是通过Thread类创建和控制的，具体创建方式有两种，一种是继承Thread类，另外一种是实现接口Runnable。在继承Thread类或实现接口Runnable时，都需要覆写run方法。

1.7.3 关键字

static

native

关键字native主要目的在于使用JNI（Java Native Interface）编程框架扩展Java代码的功能，实现Java代码同本地应用程序、其他语言编写的类库之间的相互调用，从而能够直接同操作系统底层（如系统硬件）交互，有利于程序性能的提升。

sychronized

关键字sychronized表示**同步**，一旦某个线程进入synchronized修饰的方法，直到此线程从方法体返回，其他调用类中任何synchronized方法的线程尽皆被阻塞。关键字sychronized修饰代码块，又称sychronized block，在进入此代码块之后，其它线程只能等待进入代码块的线程离开，获得锁以后才能访问。

由关键字static sychronized修饰的方法是类方法，对所有该类的对象有效。基类的sychronized特性不会自动为子类所继承，子类必须显式表达。

1.8 Linux下的C/C++开发

1.8.1 C/C++开发环境配置

- 安装C/C++编译器：g++、gcc或make，由于安装包build essential包含g++，gcc和make，只要安装它即可。验证安装是否成功：gcc -v
- 安装自动生成makefile的工具automake1.9
- 安装源代码版本控制工具git

测试C程序的编译及运行：

- 编译源代码：gcc hello.c -o hello.out
- 运行输出文件hello.out

1.8.2 重要概念

预处理

在C/C++程序中，以# 开头的代码，通常放在源文件的最前面，称为预处理指令（Preprocessor Directive）。预处理是C语言的一个重要功能，由预处理器负责完成，它是在真正编译之前所做的工作。#的作用域是以行为单位的，指令无需分号（;）换行即结束，如果单行内容过多，可以使用\续行。

C/C++提供了多种预处理功能：宏定义、文件包含、条件编译等。

宏定义

在C语言源程序中允许使用一个标识符来表示一个字符串，称为“宏”。被定义为“宏”的标识符称为“宏名”。在编译预处理时，对程序中所有出现的“宏名”，都用宏定义中的字符串去代换，这称为“宏代换”或“宏展开”。

宏的定义语法：#define identifier replacement

宏存在两种形式：有参、无参。无参的比较常见，而有参的比如：#define getmax(a,b) a>b?a:b

宏的生存周期是预处理指令#define与#undef二者的间隔，#undef取消宏的定义。

使用预处理指令可以定义常量，比如#define PI 3.14159

条件包含

指令#ifdef，#ifndef，#if，#else，#endif 和#elif可以灵活地选择是否包含某些程序，其中#elif表示else if。

源文件包含

当预处理器发现了指令#include，则会将其替换成指定的文件，存在两种用法：（1）如果是标准库提供的头文件，则使用#include <xx>（2）否则，则使用#include “xx.h”。

Pragma指令

#pragma设定编译器的状态或者是指示编译器完成一些特定的动作，因此它是依赖于编译器的。比较常用的是#pragma once，在头文件的第一行加入这条指令能够保证头文件仅仅被编译一次。

1.9 集成开发环境

Eclipse平台是用于开发工具的一个框架，它不直接支持C/C++，但可以使用外部插件来提供支持。CDT 是完全用Java 实现的开放源码项目（Common Public License 特许），是Eclipse SDK平台的一组插件。CDT将C/C++透视图添加到Eclipse 工作台（Workbench），从而使得Eclipse能够支持多种视图和向导以及高级编辑和调试。

在安装完Eclipse for Java（Juno）以后，选择Help-Install New Software，在地址栏里输入http://download.eclipse.org/tools/cdt/releases/juno/，然后勾选Name列表中的CDT Main Features，连续点击Next[1]。

此外，也可以从Oracle网站分别下载eclipse-java-juno-SR1-linux-gtk-x86_64.tar.gz和eclipse-cpp-juno-SR1-linux-gtk-x86_64.tar.gz，安装两个集成环境：Eclipse for Java和Eclipse for C++。

1.9.1 Eclipse批量重命名

选中类中的一个方法，按组合键Ctrl+Alt+H，显示出调用它的层次结构图，再按组合键Shift+Alt+R就可以统一修改名称。此外，也可以直接使用Refactor+Rename功能重命名。

[1] http://www.360doc.com/content/10/0329/00/829197_20672348.shtml

1.9.2 MakeFile

如果编写的工程比较大，设计到多个类，逐个编译文件就比较麻烦。makefile可以更高效的编译源文件，特别是当修改部分源代码时，按照makefile中制定的规则，只编译修改的源文件，从而节约大量的时间。makefile文件主要在Linux系统上编程使用[2]。

1.9.3 Ant

Like many open source java projects, Lucene uses Apache Ant for build control. Ant is "kind of like make without make's wrinkles", Ant is implemented in Java and uses xml-based (build.xml) configuration files.

Ant是一个基于JAVA的自动化脚本引擎，脚本格式为XML，除了做JAVA编译相关任务外，Ant还可以通过插件实现很多应用的调用，比make脚本还要好维护。下边介绍如何手动安装Ant：

（1） 到Apache官网下载最新版本的ant：http://ant.apache.org/bindownload.cgi

（2） 解压下载下来的.tar.gz文件：tar -xf apache-ant-1.8.2-bin.tar.gz (可能会要求输入密码)

（3） 将解压出来的文件移动到/opt/下：sudo mv apache-ant-1.8.2 /opt/ （sudo 不能省，否则没有权限）

（4） 配置环境变量：sudo vim /etc/profile，在原来基础上添加以下蓝体字

export ANT_HOME=/opt/apache-ant-1.8.2

export JAVA_HOME=/usr/lib/jvm/java-6-openjdk

export PATH=$JAVA_HOME/bin:$PATH:$ANT_HOME/bin

export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar

（5） 让配置生效：*source /etc/profile*

1.9.4 版本控制

目前比较流行的几款版本控制系统包括：svn，cvs和git，其中，cvs和git是Eclipse juno默认安装的两个系统。本节主要介绍如何使用git做版本控制（Win7平台）。

（1） 进入eclipse，选择一个project，右键-Team-Share Project-选择repository type(cvs,git)-git-configure git repository-create a new git repository。初始化该项目的一个git版本库，这样就可以在新建目录中发现新建了一个同名项目，还有一个.git文件夹。

（2） 选择eclipse对应项目，可以发现多了一个?号，右键选择Team可以选择commit。如果某个文件有改动，可以commit changes to git repository，添加comment并勾选对应文件。然后对应项目、文件夹就出现一个仓库图标，表明已经有commit。

（3） 如果有多次改变，并进行多次Commit，则Team-Show Annotation可以查看Commit历史，点选Id，右键-Checkout可以恢复到指定时刻的状态。

（4） 可以通过Preference-Git-Configuration修改Commitor的Name和Email。

1.9.5 Lucene

在Ubuntu下安装使用Lucene：

（1） 下载Lucene源文件，如lucene-4.1.0.src.tgz

[2]The CDT can generate makefiles automatically when you create a Managed Make C project or a Managed Make C++ project. You have the option of creating a Standard Make C project or a Standard Make C++ project and providing the makefile yourself

（2） 安装配置（前提是安装了jdk，ant，ivy）进入eclipse安装根目录,直接键入ant执行build任务

（3） 输入命令ant，在build目录下可以找到jar文件，将其添加为CLASSPATH路径变量，就可以全局使用

（4） 导入Eclipse项目: 工程Properties-Java Build Path-Libraries-Add External JARs

1.9.6 Nutch

在Ubuntu上安装配置Nutch:

（1） 下载Nutch源文件apache-nutch-1.6-src.tar.gz

（2） 拷贝到指定目录/usr/local/ *sudo cp Downloads/apache-nutch-1.6-src.tar.gz /usr/local/*

（3） 解压文件*tar zxvf apache-nutch-1.6-src.tar.gz*

（4） 进入到目录/src/bin/，执行nutch或crawler

1.9.7 Hadoop

Hadoop itself is written in Java; it thus accepts Java code natively for Mappers and Reducers. Hadoop also comes with two adapter layers which allow code written in other languages to be used in MapReduce programs.Pipes is a library which allows C++ source code to be used for Mapper and Reducer code. Applications which require high numerical performance may see better throughput if written in C++ and used through Pipes. This library is supported on **32-bit** Linux installations.

1.9.8 Apache Tika

Apache Tika是一个用Java实现的开源库，支持从多种格式中（如HTML、PDF和Word等）抽取文本和元数据，也能用于语言和MIME类型识别。实际上它就是现有的第三方解析器（如PDFBox）的包装器，只是提供了一个统一的API来使用这些解析器。

Tika的API十分便捷，核心是Parser interface，其中定义了一个parse方法：public void parse(InputStream stream, ContentHandler handler, Metadata metadata) 用stream参数传递需要解析的文件流，文本内容会被传入handler，而元数据会更新至metadata。

可以使用Tika的ParserUtils工具来根据文件的mime-type来得到一个适当的Parser来进行解析工作。或者Tika还提供了一个AutoDetectParser根据不同的二进制文件的特殊格式（比如说Magic Code），来寻找适合的Parser。

1.9.9 环境变量

在Ubuntu中有如下几个文件可以设置环境变量：

（1） /etc/profile:登录时，操作系统定制用户环境时使用的第一个文件，此文件为系统的每个用户设置环境信息，当用户第一次登录时，该文件被执行。

（2） /etc/environment:在登录时操作系统使用的第二个文件，系统在读取你自己的profile前,设置环境文件的环境变量。

（3） /.bash_profile:在登录时用到的第三个文件是.profile文件，每个用户都可使用该文件输入专用于自己使用的shell信息，当用户登录时,该文件仅仅执行一次!默认情况下，他设置一些环境变量，执行用户的.bashrc 文件。

（4） /.bashrc:为每一个运行bash shell的用户执行此文件。当bash shell被打开时，该文件被读取。该文件包含专用于你的bash shell的bash信息，当登录时以及每次打开新的shell时，该文件被读取。

命令行：使用env命令显示所有的环境变量，使用unset命令来清除环境变量，使用echo显示常见环境变量，如*echo $HOME*即可找到根目录，使用export可以设置一个环境变量，如*export HELLO="CH"*，则使用*echo $HELLO*即显示*CH*，属于临时环境变量，关闭shell则失效。

通过修改环境变量定义文件来修改环境变量:

（1） cd 到用户根目录下

（2） ls -a 查看所有文件，包含隐藏的文件

（3） vim .bash_profile 修改环境变量定义文件

（4） 编辑PATH声明，其格式为：PATH=$PATH:PATH 1:PATH 2:PATH 3。各个PATH之间用冒号隔开。环境变量更改后，在用户下次登陆时生效，如果想立刻生效，则可执行语句：*source .bash_profile*

1.9.10 Java环境变量

（1） /etc/profile文件添加

- JAVA_HOME=/usr/local/develop/jdk1.6.0.22
- PATH=$JAVA_HOME/bin:$PATH
- CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
- export JAVA_HOME PATH CLASSPATH

（2） .bashrc文件添加

- set JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.22
- export JAVA_HOME
- set PATH=$JAVA_HOME/bin:$PATH
- export PATH
- set CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
- export CLASSPATH

1.10 虚拟机的安装

在Win7系统上安装虚拟机VirtualBox，并在虚拟机上安装Ubuntu系统的基本步骤：

- 下载虚拟机安装文件：VirtualBox-4.2.6-82870-Win.exe双击执行安装
- 新建–虚拟电脑（电脑名称：ubuntu64bit，系统类型：Linux-Ubuntu64bit）

  设置内存大小512m–1024m

  创建新的虚拟硬盘（动态扩展或固定大小）

  指定虚拟硬盘所在位置和大小（20G）
- 安装Linux系统

  下载Ubuntu镜像文件：ubuntu-12.04-desktop-amd64.iso

  基本设置：设置–存储–控制器–分配光驱：选择Ubuntu镜像文件

第二章　Latex

2.1　Latex绘制图片

绘制图片的途径有多个，可以使用matlab，excel，也可以使用gnuplot。比如使用bar([d1;d2;d3]')在matlab上绘制柱形图。matlab绘图比较方便，但缺点是输出的图片不能直接插入Latex，需要经过各种转换。本节主要介绍使用gnuplot绘制柱形图，导出eps图片，然后插入到Latex。

2.1.1　Excel

LaTeX中绘制表格是比较麻烦的, excel宏excel2latex可以直接将excel中的表格转化为latex源代码的格式，用excel打开excel2latex.xla，然后你就会在工具菜单上看到一个新的按钮. 选定要转换的表格部分, 然后点击按钮，就可以得到表格的LaTeX源代码。详情点击。

2.1.2　Gnuplot

gnuplot小巧实用，主要用来绘制2D/3D的数据或者函数图像。主流Linux发行版都包含gnuplot，因此在Linux上安装很简单，只要用各相应发行版的软件安装工具直接安装就可以了。在Windows下，可以直接到gnuplot在sourceforge的下载页面下载最新版本，解压后到binary目录里找到gnuplot.exe[1]。

gnuplot基本用法，可以参考系列文章：http://blog.sciencenet.cn/blog-373392-527507.html

- **表示幂数，比如2**3表示2^3
- 2D作图命令：plot，比如plot sin(5*x)，默认函数样本点取100个，可以通过samples参数控制;重新绘制图片调用：replot
- 在gnuplot里面，所有参数赋值都由set 命令完成，比如修改样本点数目set samples 5000。相反地，使用unset可以取消一个参数设置
- gnuplot右上角图例(图示)称作key，标题（title）、坐标轴标签（xlabel,ylabel）。论文插图要求的文字说明（Caption）。在gnuplot里，很多跟坐标有关的参数，都有成对出现的，以x和y打头
- 横坐标取值范围由xrange 参数控制,比如set xrange [-2*pi:2*pi]
- 横轴主刻度和分刻度(主刻度之间)，分别用xtics 和mxtics 表示（m 表示minor）。比如set xtics pi 表示主刻度之间距离pi，set mxtics 2表示相邻主刻度被分成两份。也可以通过字符控制，set xtics (“-2π” -2*pi, “” -1.5*pi 1, “-π” -pi, “” -0.5*pi 1, “0” 0, “” 0.5*pi 1, “π” pi, “” 1.5*pi 1, “2π” 2*pi)直接规定了每个刻度的位置和显示的字符。每一个刻度对应三个参数：显示字符、刻度位置、刻度等级。刻度等级为0 时表示主刻度，等级为1 时表示分刻度。对于主刻度（等级为0 时），表示等级的参数也可以省略不写。各个刻度的参数之间用逗号隔开。还可以通过set ytics -1,0.5,1表示最小主刻度、主刻度步长、最大主刻度
- 数据绘图需要的数据文件是纯文本：filename.dat，文本中#表示注释行。根据数据文件绘图：plot “filename.dat”默认描点，也可以是plot “filename.dat” with lines连线

[1] Gnuplot in Action

- with命令紧跟绘图方式，默认为points，gnuplot支持了大概30种方式，可以调用help plot查看帮助文档。gnuplot里面有几个控制点和线画法风格的参数：linestyle，linetype，linewidth，linecolor，pointtype，pointsize，比如plot “datafile.dat” with linespoints linecolor 3 linewidth 2 pointtype 7 pointsize 2，若要详细了解各个数字的含义，可以输入命令test显示示例
- 把多组数据绘制到同一个图上（两列以上），可以用using 指定使用哪列数据。例如using 1:3 表示使用第一列和第三列数据，第一列为横坐标，第二列为纵坐标。如果想把多组数据绘制到一个图上，只要使用一个plot 命令，后面跟多组数据，每组数据之间用逗号隔开就可以了。
- 缩写plot “precipitation.dat” u 1:2 w lp pt 5 title “北京”, “precipitation.dat” u 1:3 w lp pt 7 title “上海”，其中的缩写u-using,w-with,lp-linespoints,pt-pointtype。title表示图例的label
- 引号内的内容为字符串，大多数情况下双引号和单引号没有区别，除非遇到特殊字符（例如换行符\n），这时候单引号会把特殊字符当成一般字符处理，而双引号会按照特殊字符的意义将其展开
- gnuplot里面控制图像输出方式的命令是terminal，比如set terminal postscript eps讲terminal设置为postscript，而eps属于postscript的一个参数。还可以修改图片参数set terminal postscript eps color solid linewidth 2 “Helvetica” 20，color-彩色，solid-实线，helvetica字体，字号20，了解更多字体，可以参考http://xfig.org/userman/attributes.html#font-panel
- set output “precipitation.eps”设置输出文件的文件名,unset output关闭输出，切换回屏幕显示set term wxt
- unset border 命令把图像边框去掉，border后面的数字是一个12个bit 的整数，每一位bit 表示一个边框。于是1 ,2, 4, 8 分别对应下，左，上，右边框，所以15（1+2+4+8）绘制出全部的边框，若7（15-8）则不显示右边框。

使用gnuplot直接逐行输入命令比较费时，可以编辑plt后缀的脚本文件，使用gnuplot打开即执行，类似于批处理文件。注意保存的编码方式可能影响识别（UTF-8 无BOM）。也可以是通过命令行load ’.plt’ 调整图形可以使用gnuplot的load命令，即时运行，并使用GSview查看输出效果，然后微调。

2.2 插入图片

了解如何在LaTex插入图片，可以参考“插图指南”。

任何LaTex对象（字符，图形等）都把盒子作为单位，每个盒子在它的左侧均有一参考点（Reference point）。盒子的基线（baseline）是通过参考点的一条水平线。当LATEX 排列文本时，这些字符的参考点被从左到右的排成一条直线，称为当前基线（current baseline），并使它与字符的基线对齐。LaTex也用同样的方法来处理图形和其它对象，每个对象的参考点都被放置于当前基线上。每个LaTex盒子的大小由高度、深度、宽度（height,depth,width）来决定。高度是参考点到盒子顶部的距离，深度是参考点到盒子底部的距离，宽度则是盒子的宽度。全部高度（totalheight）被定义为从盒子底部到顶部的距离，即：全部高度=高度+深度

所有未曾旋转的EPS 图形的参考点都是它的左下角，深度为零，高度就等于全部高度。

LaTeX中一般只直接支持插入eps(Encapsulated PostScript)格式的图形文件, 因此在图片插入LaTex文档之前应先设法得到图片的eps格式的文件，比如可以访问:http://www.tlhiv.org/rast2vec/，对图片做在线转换。EPS文件必须含有一个BoundingBox 行来确定EPS 图形的大小。BoundingBox 行包含两对整数值：左下角的坐标，右上角的坐标。

使用Excel、在线转换组合十分好。首先在Excel使用实验数据生成图表，然后复制到粘贴板，直接保存为png或者jpeg，然后通过在线转换成eps文件即可。在绘图时，单击右键，选择数据可以编辑坐标轴和图例标签文本。如果还要创建与刚创建的图表相似的图表，则可以将该图表保存为模板，以用作其他类似图表的基础。单击要另存为模板的图表。在“设计”选项卡上的“类型”组中，单击“另存为模板”。

2.2.1 includegraphics命令

\includegraphics[option]{file}: option

- width=0.80\textwidth 将所插入图形缩放为文本行宽的80%
- \ scalebox{水平缩放因子}[垂直缩放因子]{对象}
- \ resizebox{宽度}{高度}{对象}
- \ resizebox*{宽度}{全部高度}{对象}

把\includegraphics 命令放到\fbox 中会使所插入的图形置于一个带框盒子中，还可以通过使用\setlength 命令设置LaTex 的长度变量\fboxrule 和\fboxsep 来修改盒子边框宽度，图形与边框的距离，如:

1. \begin{figure}
2. \centering
3. \setlength{fboxrule}{3pt}
4. \setlength{fboxsep}{1cm}
5. \fbox{\includegraphics[totalheight=2in]{pend.eps}}
6. \end{figure}

若希望在两栏的文档中插入跨栏图表，则改变到环境figure*，图表改变到环境table*。图形的浮动参数：**H**ere，**T**op，**B**ottom，**P**age。

2.3 Trick

2.3.1 查看Latex系统自带的包文档

在命令行终端执行texdoc ulem，系统自动弹出ulem的文档页面

2.3.2 编译错误处理

编译报错时，在Console文本输入e按回车会自动定位错误在源文档中的位置。造成编译报错的原因很多，简单的处理方式就是直接删除输出文件（工具条中的“垃圾回收站”标志），重新编译。

2.3.3 文本上标

如果不希望文本中出现数学环境下的上标标记，则可以使用\textsuperscript命令。

2.3.4 下标换行：substack

下标换行实例：

$$\min_{\substack{\theta>0\\ \beta<-1}} \sum_{i=1}^{n} x_i\theta/\beta^2$$

表格

插图