

第三章 嵌入式Linux操作系统

华东理工大学计算机系
罗飞

Content

1 嵌入式Linux简介

2 内存管理

3 进程管理

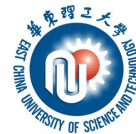
4 文件系统

5 OpenWrt

6 Linux编程基础



Linux的诞生与发展



◆Linux的诞生与版本历史

◆Linux的诞生

◆Linux的版本发展历史

◆Linux的应用领域

◆Linux在嵌入式领域的延伸

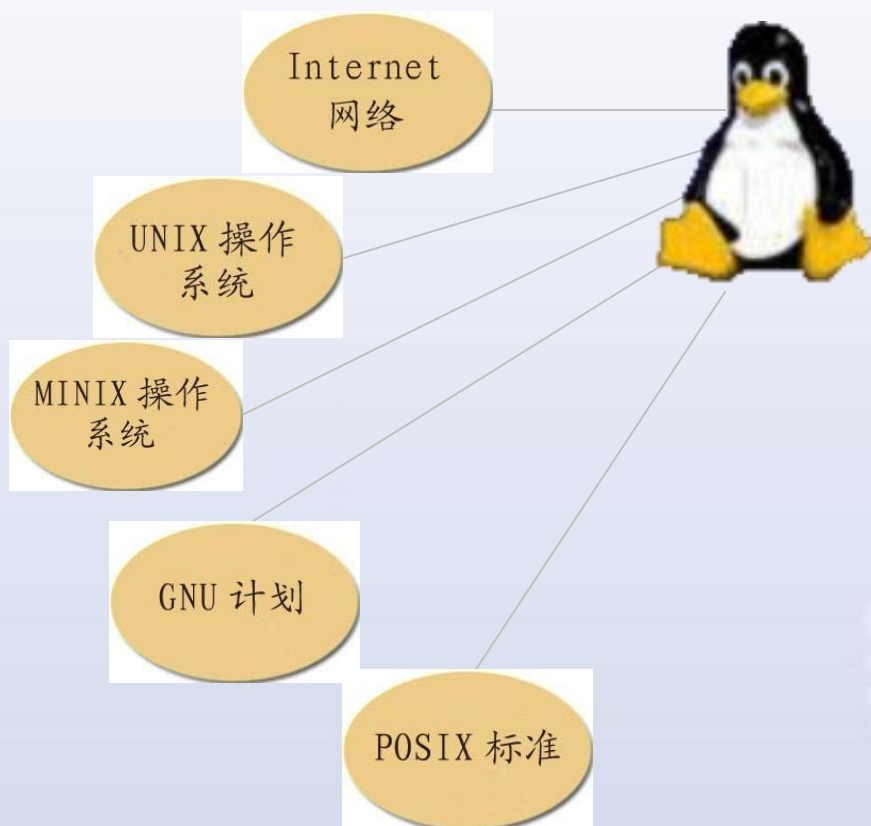
◆嵌入式Linux的发展及应用前景



- 从1991年的4月份开始，Linus尝试将GNU软件移植到Minix系统上。
- 1991年的10月5日，Linus在comp. os. minix新闻组上正式向外宣布Linux内核系统的诞生。



Linux 操作系统的诞生、发展和成长过程跟以下五个因素有关：



★ **UNIX 操作系统**是美国贝尔实验室于1969年夏在 DEC PDP-7 小型计算机上开发的一个分时操作系统

➤ **MINIX 系统**是由 Andrew S. Tanenbaum (AST) 1987 年开发的，主要用于学生学习操作系统原理

➤ **GNU 计划**旨在开发一个类似 Unix、并且是自由软件的完整操作系统：GNU 系统。

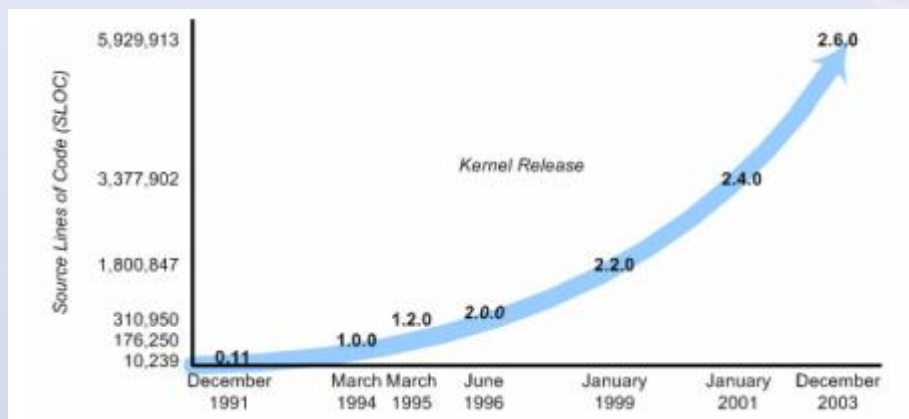
➤ **POSIX可移植操作系统接口标准**是由 IEEE 开发的，由 ISO/IEC 标准化的一簇标准。该标准是基于现有的 UNIX 实践和经验，描述了操作系统的调用服务接口，用于保证编制的应用程序可以在源代码一级上在多种操作系统上移植运行。

➤ 如果没有 **Internet 网**，没有遍布全世界的无数计算机黑客通过网络的无私奉献，那么 Linux 绝对不可能发展到现在的水平。



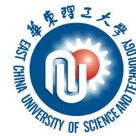
Linux的版本发展历史

- 从 Linux 诞生开始，Linux 内核就从来没有停止过升级，从 0.02 版本到 1999年具有里程碑意义的2.2 版本，一直到我们现在看到的 x.x.xx版本。

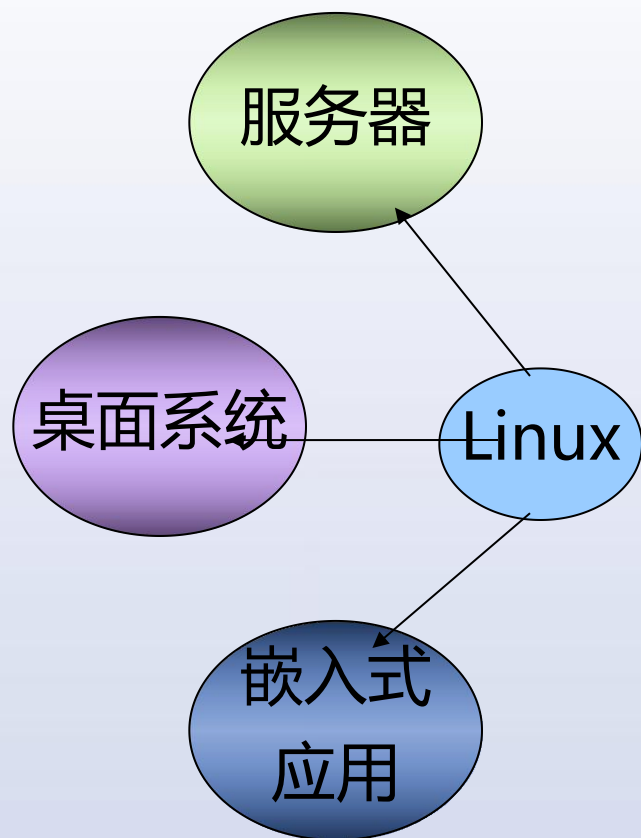


从未停止过升级

- Linux 内核版本有两种：
 - - 稳定版和开发版
- Linux内核的命名机制：num.num.num。
 - - 第一个数字是主版本号
 - - 第二个数字是次版本号
 - - 第三个数字是修订版本号



Linux的应用领域





- ◆ 由于 Linux 具有对各种设备的广泛支持性，因此，能方便地应用于机顶盒、IA 设备、PDA、掌上电脑、WAP 手机、寻呼机、车载盒以及工业控制等智能信息产品中。
- ◆ 与 PC 相比，手持设备、IA 设备以及信息家电的市场容量要高得多，而 Linux 嵌入式系统的强大的生命力和利用价值，使越来越多的企业和高校表现出对它极大的研发热情。
- ◆ Linux 嵌入式操作系统所具有的技术优势和独特的开发模式给业界以新异，有理由相信，它能成为Internet 时代嵌入式操作系统中的最强音。



1、主流嵌入式Linux

➤ **uClinux**

➤ **RT-Linux**

➤ **红旗嵌入式Linux**

- uCLinux, 代表着 “Micro-Control-Linux” , 意为 “针对微控制领域而设计的Linux系统” 。
- uCLinux是Lineo公司的主打产品, 同时也是开放源码的嵌入式Linux的典范之作。
- uCLinux支持2.6内核, 已经被成功地移植到很多平台上。

- ◆ RT-Linux就是Real-time Linux的简写，RT-Linux是是源代码开放的具有硬实时特性的多任务操作系统。
- ◆ RT-Linux是美国新墨西哥州大学计算机科学系VictorYodaiken和Micae Brannanov开发的嵌入式Linux操作系统。
- ◆ RT-Linux已经成功地应用于航天飞机的空间数据采集、科学仪器测控和电影特技图像处理等广泛领域。



- 由北京中科红旗软件技术有限公司推出，是国内做得较好的一款嵌入式Linux操作系统。
 - 精简内核，适用于多种常见的嵌入式CPU；
 - 提供完善的嵌入式GUI和嵌入式X-Windows；
 - 提供嵌入式浏览器、邮件程序和多媒体播放程序；
 - 提供完善的开发工具和平台。

➤ Embedix

- Luneo根据嵌入式应用系统的特点重新设计的Linux发行版本

➤ XLinux

- 美国网虎公司（陈盈豪），号称是世界上最小的嵌入式Linux系统，内核只有143KB

➤ PoketLinux

- Agenda公司，提供跨操作系统构造统一的、标准化的和开放的信息通信基础结构，在此结构上实现端到端方案的完整平台

➤ MidoriLinux

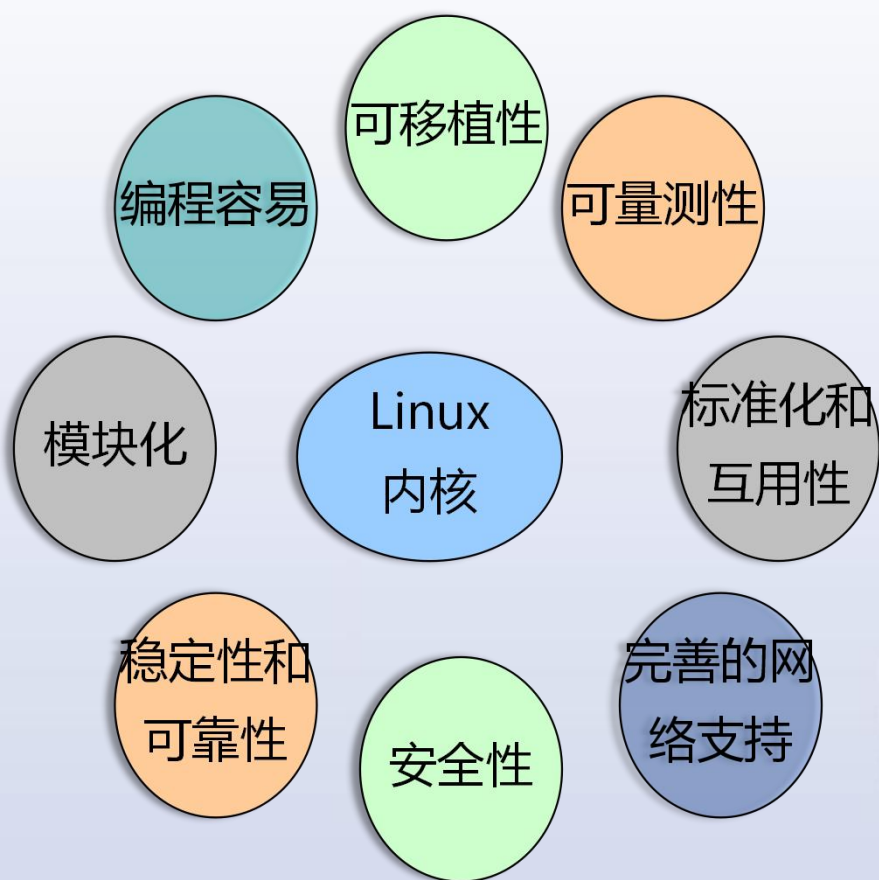
- Transmeta公司



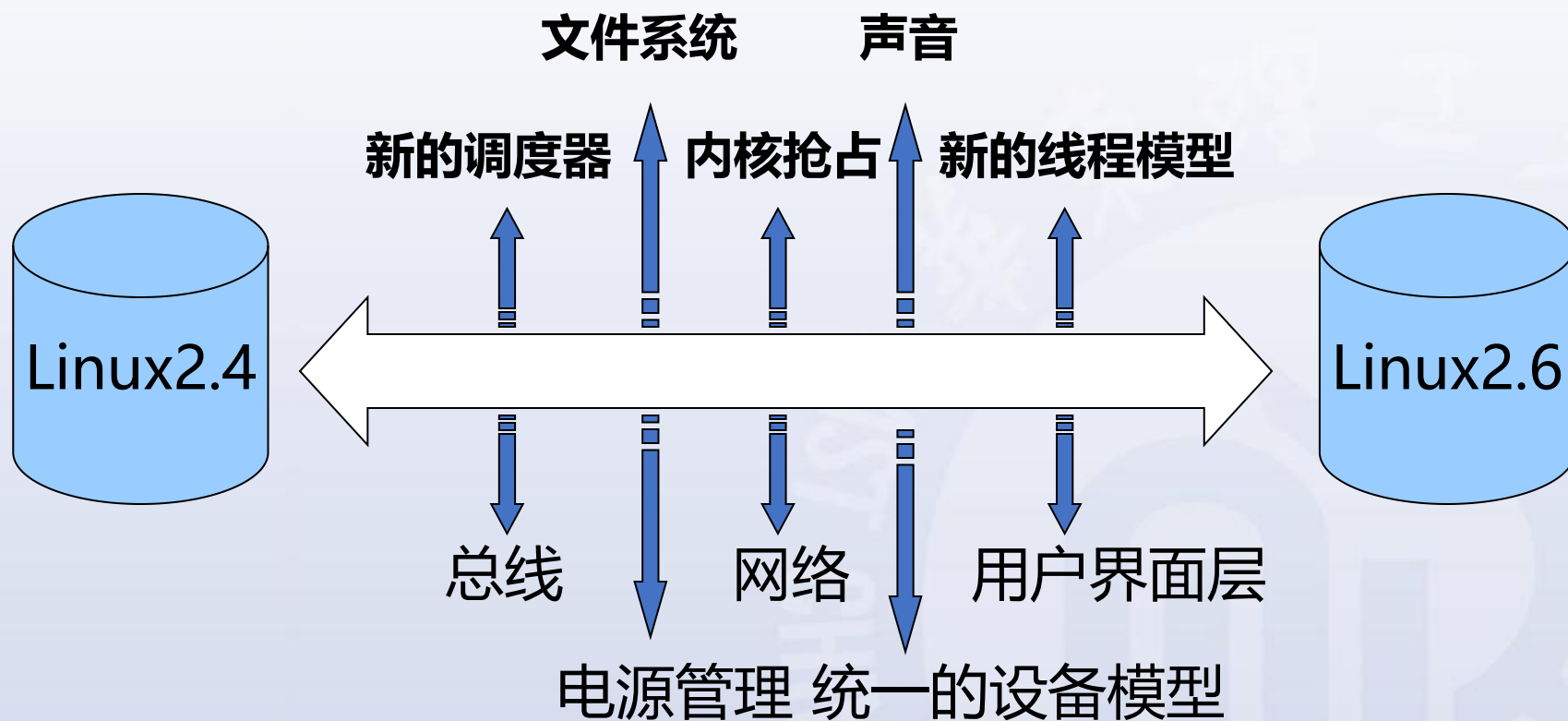
- ◆ Linux内核概述
 - ◆ Linux内核特点
 - ◆ Linux2.6内核新特性
 - ◆ Linux内核的组成
- ◆ 存储与进程管理
 - ◆ 存储管理
 - ◆ 进程调度
- ◆ 内核源代码目录结构



Linux内核特点



- Linux 内核既能支持 32 位体系结构又能支持 64 位体系结构。
- Linux 内核全部源代码是遵守 GPL 软件许可的免费软件
- 开发者可以免费得到社区的贡献、支持、检查代码和测试。驱动程序可以免费发布给其他人，可以静态编译进内核。
- 以源码形式发布驱动程序，可以不必为每一个内核版本和补丁版本都提供二进制的程序

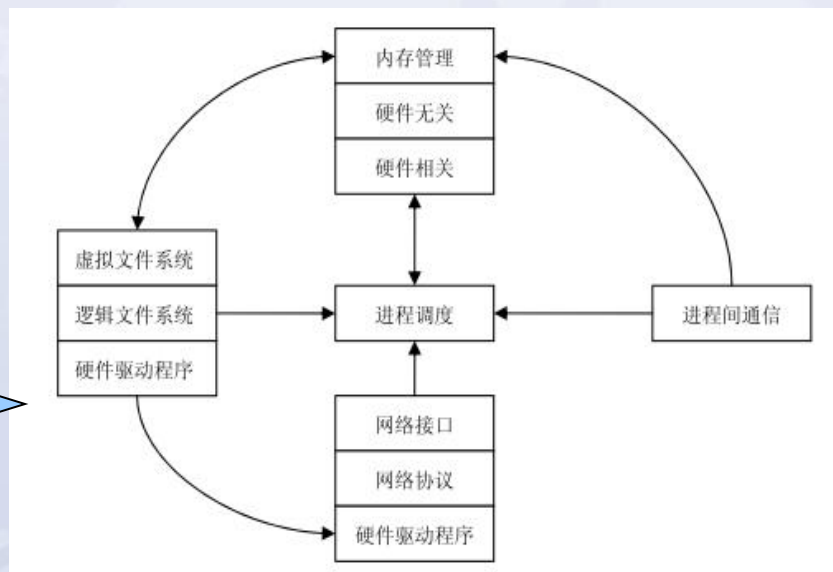
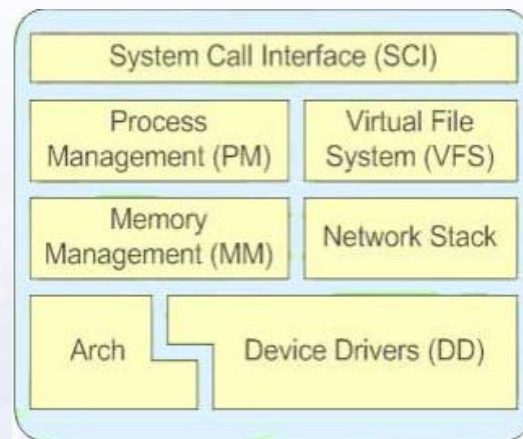




Linux内核的组成

- ◆ 进程调度程序 (SCHED) 负责控制进程访CPU。
- ◆ 保证进程能够公平地访问 CPU, 同时保证内核可以准时执行一些必需的硬件操作;
- ◆ 内存管理程序 (MM) 使多个进程可以安全地共享机器的主存系统, 并支持虚拟内存;
- ◆ 虚拟文件系统 (VFS) 通过提供一个所有设备的公共文件接口, VFS 抽象了不同硬件设备的细节。此外, VFS 支持与其他操作系统兼容的不同的文件系统格式;
- ◆ 网络接口 (NET) 提供对许多建网标准和网络硬件的访问;
- ◆ 进程间通信 (IPC) 子系统为进程与进程之间的通信提供了一些机制。

这 5 部分之间是相互依赖的关系



2、内存管理

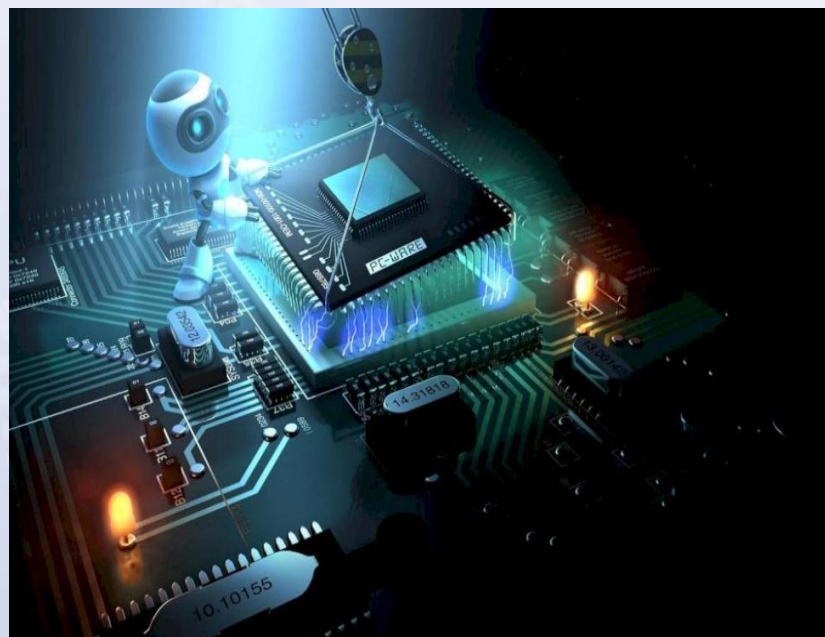
- 内存管理和MMU
- 标准Linux的内存管理
- uCLinux的内存管理



内存管理

- 包含地址映射、内存空间的分配，有时候还包括地址访问的限制（即保护机制）
- 如果将I/O也放在内存地址空间中，则还要包括I/O地址的映射
- 另外，像代码段、数据段、堆栈段空间的分配等等都属于内存管理

- ◆ 内存管理单元
- ◆ 主要作用是两个方面的
 - ◆ 一是地址映射
 - ◆ 二是对地址访问的保护和限制





◆使用虚拟存储器技术

- ◆运行比内存还要大的程序
- ◆先加载部分程序运行，缩短了程序启动的时间
- ◆可以使多个程序同时驻留在内存中提高CPU的利用率
- ◆可以运行重定位程序



标准Linux的内存管理(2)

- 使用虚拟存储器技术
 - 写机器无关的代码
 - 减轻程序员分配和管理内存资源的负担
 - 可以进行内存共享
 - 提供内存保护



uCLinux的内存管理

- ◆ 不能使用处理器的虚拟内存管理技术
- ◆ 采用存储器的分页管理--实存储器管理 (Real Memory Management)
- ◆ 开发人员参与系统的内存管理
 - 编译内核时，开发人员必须告诉系统这块开发板到底拥有多少的内存
 - 开发应用程序时必须考虑内存的分配情况并关注应用程序需要运行空间的大小



内存映射: Linux VS uCLinux





3、进程管理

- ◆进程和进程管理
- ◆RT-Linux的进程管理
- ◆标准Linux的进程管理
- ◆uClinux的进程管理



- 进程是一个运行程序并为其提供执行环境的实体
- 它包括一个地址空间和至少一个控制点
 - 进程在这个地址空间上执行单一指令序列



进程和进程管理

- 任务调度主要是协调进程对计算机系统内资源（如内存、I/O设备、CPU）的争夺使用
 - 任务—资源匹配
- 进程调度
 - CPU调度



➤ 基本概念

➤ 上下文、上下文切换

➤ 抢占、优先抢占

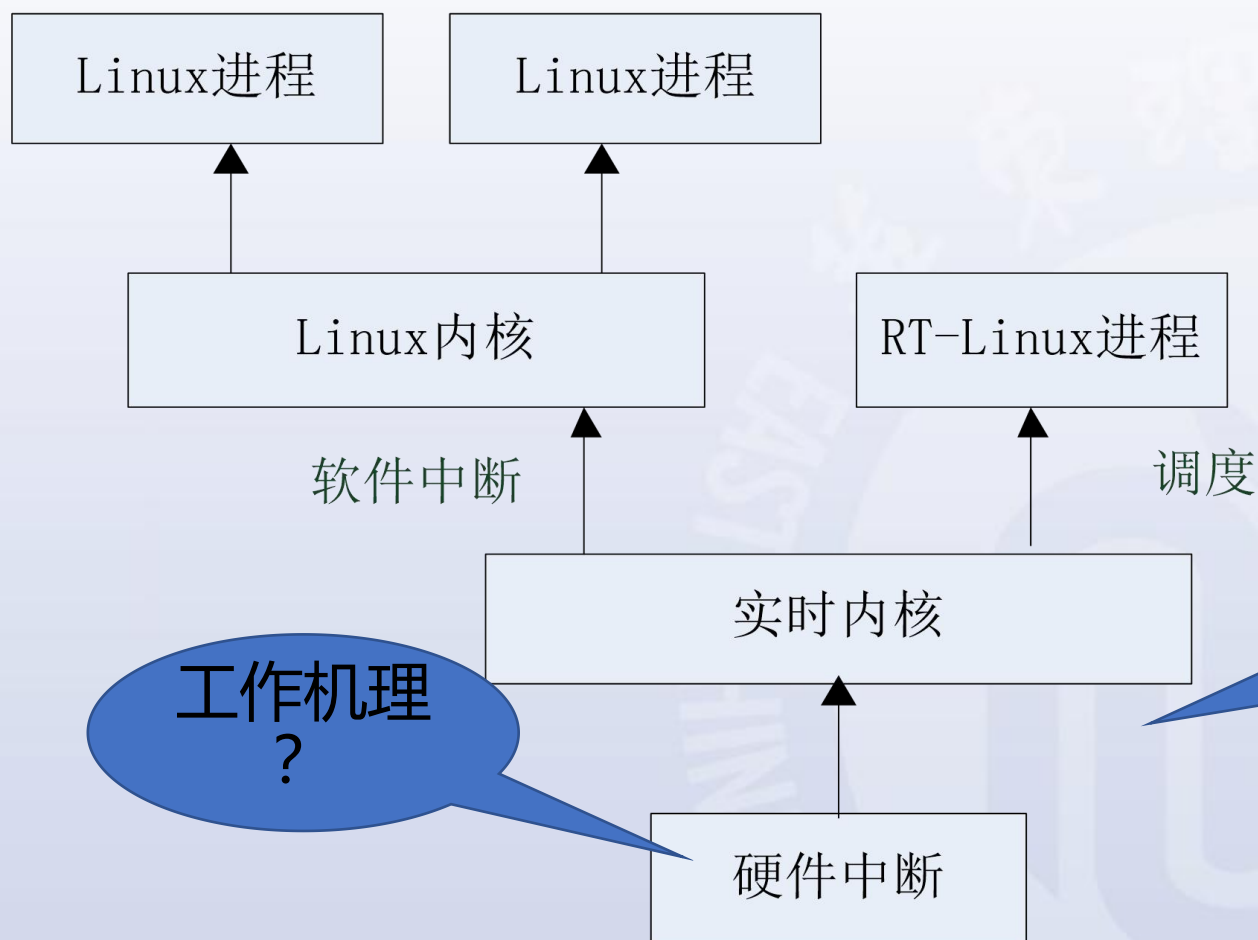
➤ 轮转调度

➤ 进程调度策略

➤ 抢占式调度

➤ 非抢占式调度

- RT-Linux有两种中断：硬中断和软中断





RT-Linux进程调度

- 将标准Linux内核作为实时操作系统里优先权最低的线程来运行
- 默认采用优先级的调度策略
- 也支持其他策略
 - 最短时限最先调度、确定周期调度
 - 基于模块化设计的任务调度器
- RT-FIFO：实时与非实时进程数据交换



标准Linux的进程管理

◆ Linux进程由进程标识符（PID）表示

◆ 其他进程属性

- 父进程和父进程的ID（PPID）
- 启动进程的用户ID（UID）和所归属的组（GID）
- 进程状态：运行R、休眠S、僵尸Z
- 优先级
- 进程所连接的终端名
- 进程资源占用



标准Linux的进程管理

- ◆ Linux系统提供系统调用拷贝现行进程的内容，以产生新的进程
 - ◆ 父、子进程异同点？
- ◆ Linux进程还可以通过exec系统调用产生
 - ◆ 所发生变化？
- ◆ vfork





➤ Linux进程的调度

➤ 分时调度策略 (SCHED_OTHER)

➤ 先到先服务的实时调度策略 (SCHED_FIFO)

➤ 时间片轮转的实时调度策略 (SCHED_RR)

- uCLinux的进程调度沿用了Linux的传统
- 进程切换：由于没有MMU，系统虽然支持fork系统调用，但其实质上是vfork
- 启动新的应用程序时系统必须为应用程序分配存储空间

➤ 文件系统定义

➤ Linux文件系统

➤ 嵌入式Linux文件系统



- 文件系统定义：包含在磁盘驱动器或者磁盘分区的目录结构，整个磁盘空间可以给一个或者多个文件系统使用。
- 在对某个文件系统做在某一个挂载点的挂载（Mount）操作后，就可以使用该文件系统了。

- ◆ Linux支持许多种文件系统。
- ◆ ext2是Linux事实上的标准文件系统
- ◆ ext3是一种日志式文件系统，是对ext2系统的扩展，它兼容ext2

➤ Journal File System

➤ 将整个磁盘的写入动作完整记录

- 以便回溯
- 如突然关机

➤ Ext3多种日志模式

- 对所有文件数据及metadata进行日志
- 只对metadata进行日志



◆存储设备不同于PC系统

◆基于合适存储设备上的文件系统

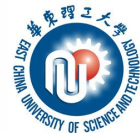
◆根文件系统

◆装载过程?





Flash Memory



- ◆ Non-Volatile内存

- ◆ 主要技术

 - NAND和NOR

- ◆ 比较

 - NAND: 单元排列串行, 以块和页为单位进行读写, 顺序读写, 随机读写慢

 - NOR: 单元排列并行, 按字节进行读写, 随机读写快, 可以片内执行



嵌入式Linux文件系统分类

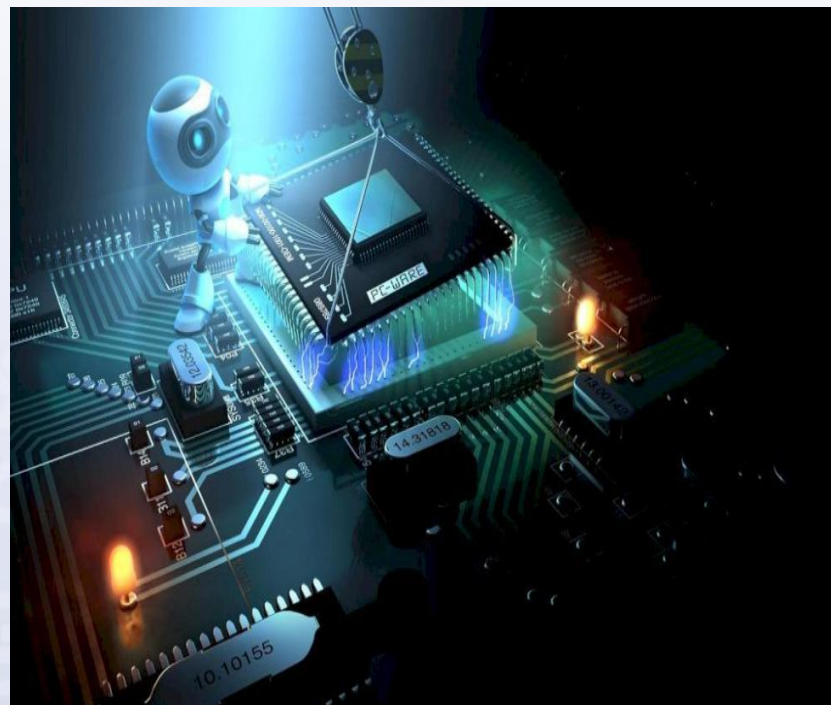
- 基于Flash的文件系统
- 基于RAM的文件系统
- 网络文件系统NFS





基于Flash的文件系统

- ◆ JFFS2文件系统
- ◆ YAFFS2文件系统
- ◆ Cramfs文件系统
- ◆ Romfs
- ◆ 其他文件系统





◆基于RAM的文件系统

- ◆RamDisk

- ◆Ramfs/Tmpfs

◆网络文件系统NFS (Network File System)



- ◆ OpenWrt是一个高度模块化、高度自动化的嵌入式的Linux发行版

强大的网络组件和扩展性

用于工控设备、智能家居、路由器等设备中

提供了已编译好的软件

- ◆ OpenWrt使用框架来构建应用程序
- ◆ 支持各种处理器架构

ARM, X86, PowerPC, MIPS等



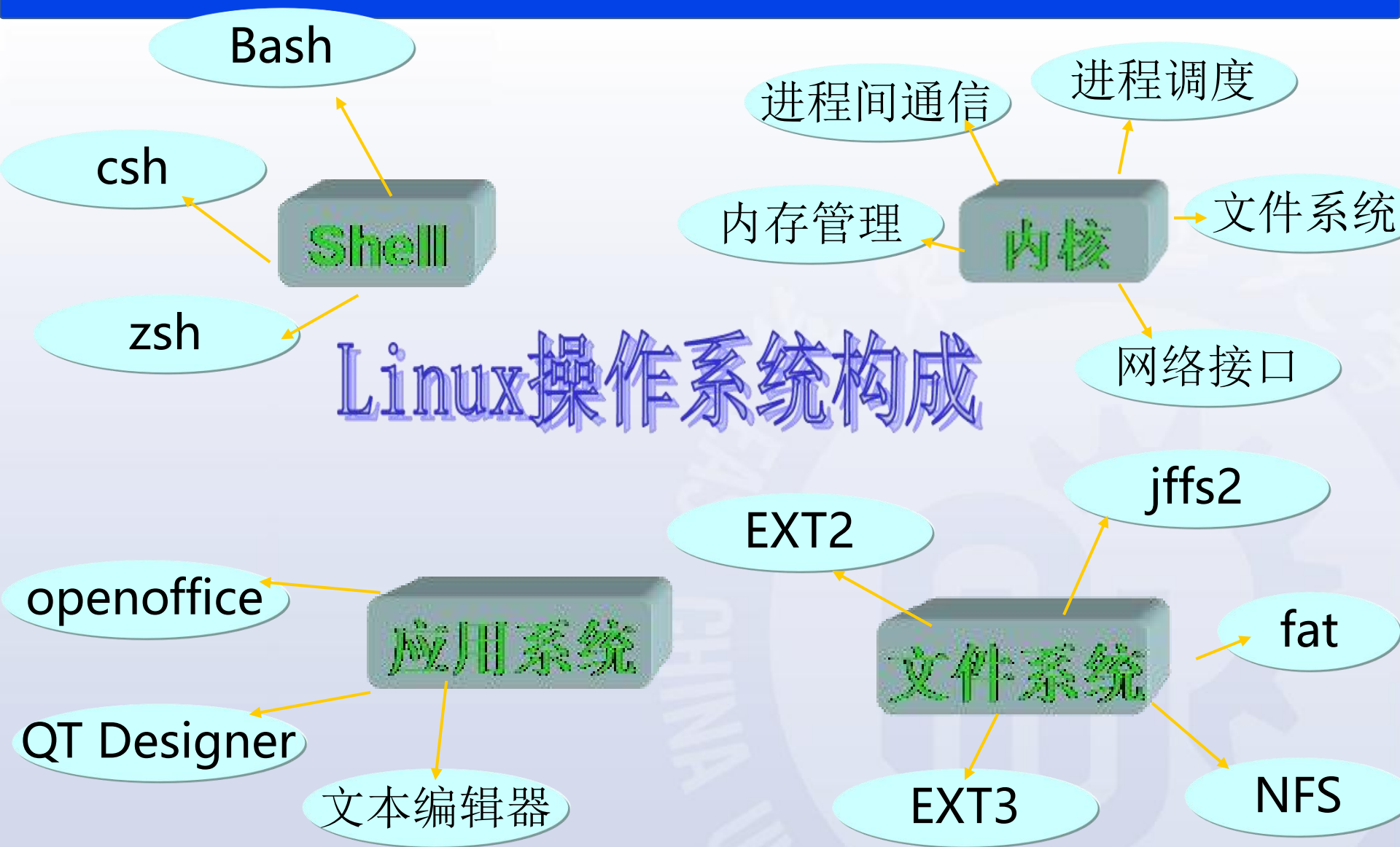


各个发行版

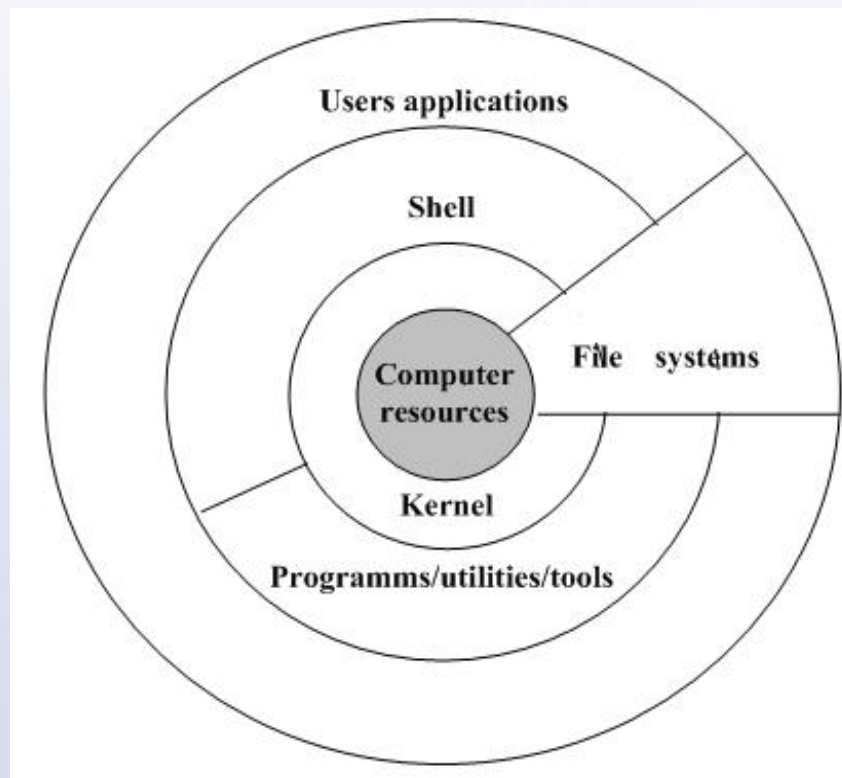
- White Russian : 2007年发布
-
- OpenWrt 19.07
- Bleeding edge: 2020年, 对应OpenWrt版本: OpenWrt 21.02.x 。
- 22.03
- Current stable release - OpenWrt 23.05.2

Status	Component	Version	Description
WIP	GCC	tbd	
WIP	musl	tbd	
WIP	Linux	5.15	
WIP	mac80211	6.1	Currently on 6.1 RC3

5、Linux编程基础



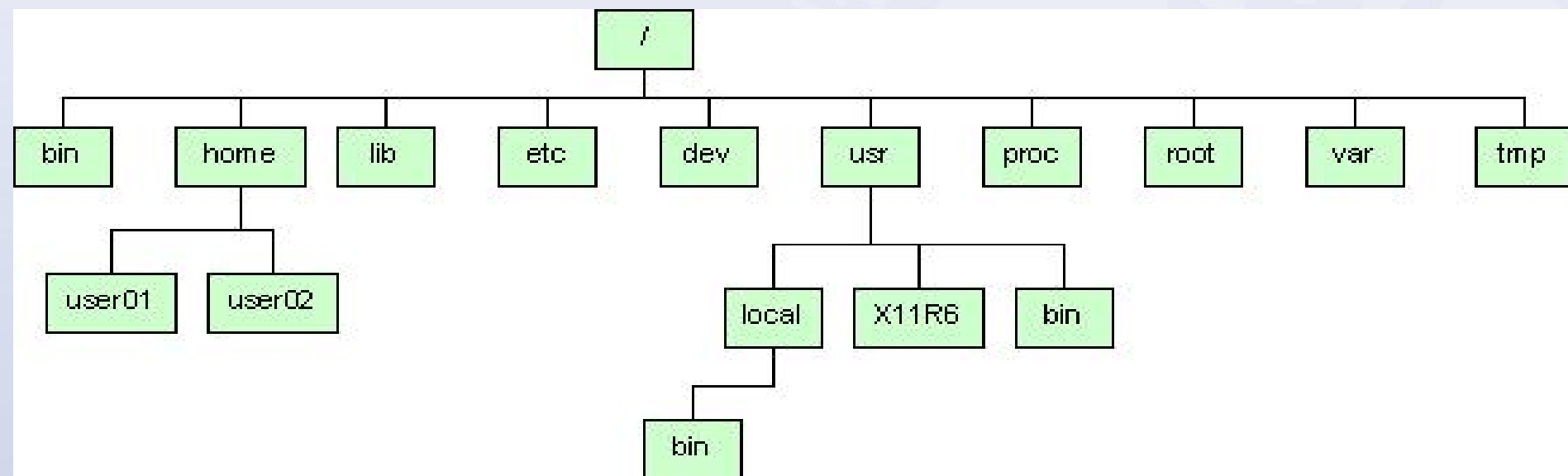
- Linux一般有4个主要部分组成
 - 内核
 - Shell
 - 文件系统
 - 应用程序



- 1. Linux内核 (Kernel)
 - 内核 (Kernel) 是运行程序和管理硬件设备的内核程序，决定着系统的性能和稳定性，实现操作系统的基本功能
 - 在硬件方面：控制硬件设备，管理内存，提供接口，处理基本I/O。
 - 在软件方面：管理文件系统，为程序分配内存和CPU时间等。

2. Linux 文件系统

- 按照目录层次的方式进行组织，可以设置目录和文件的权限、文件的共享程度
- 每个目录可以包括多个子目录以及文件，系统以“/”为根目录，用户可以创建自己的子目录，保存自己的文件
- Linux能支持多种目前流行的文件系统，如ext2、ext3、fat、vfat、ISO9660、nfs等

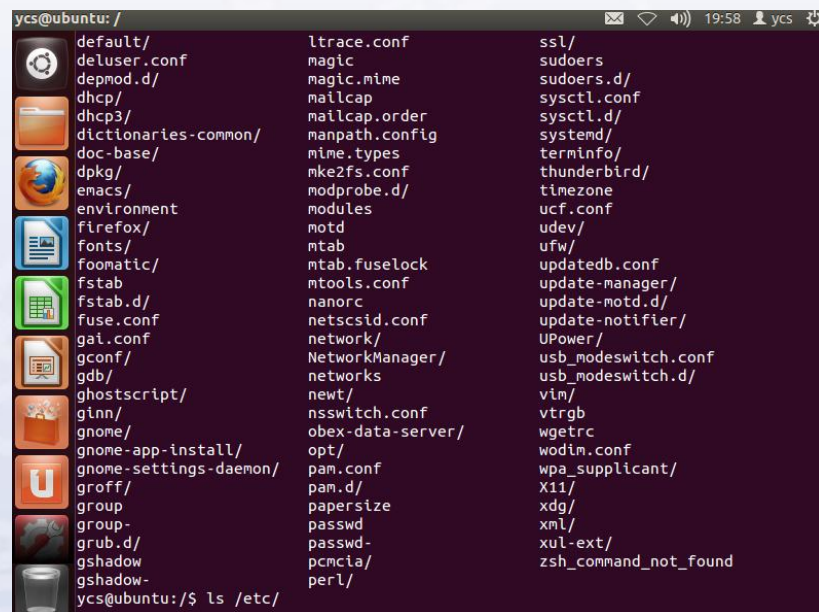
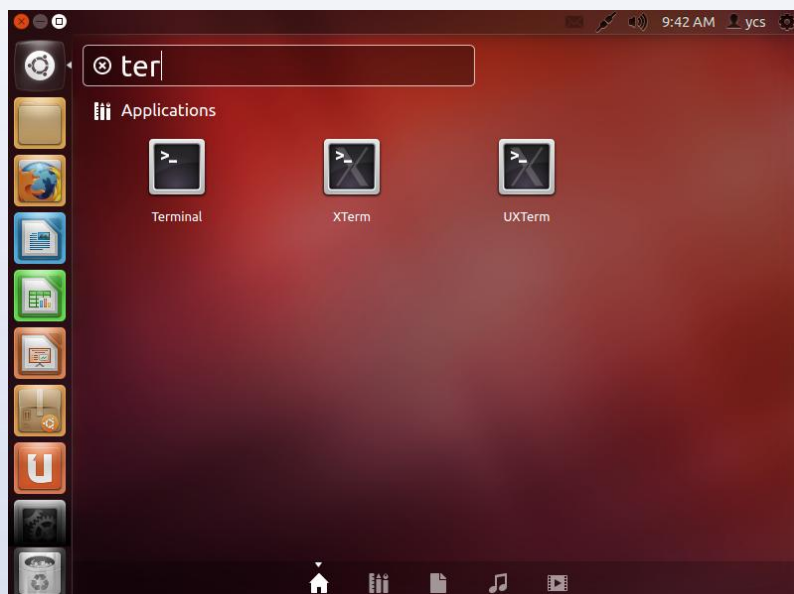


- **/bin:** 存放二进制可执行命令目录;
- **/home:** 用户主目录的基点目录, 默认情况每个用户主目录都设在该目录下, 如默认情况下用户user01的主目录是/home/user01;
- **/lib:** 存放标准程序设计库目录, 又叫动态链接共享库目录, 目录中文件类似windows里的.dll文件;
- **/etc:** 存放系统管理和配置文件目录;
- **/dev:** 存放设备特殊文件目录, 如声卡文件、磁盘文件等;

- **/usr**: 最庞大的目录, 存放应用程序和文件目录;
- **/proc**: 虚拟目录, 是系统内存的映射, 可直接访问这个目录来获取系统信息;
- **/root**: 系统管理员的主目录;
- **/var**: 存放系统产生的经常变化文件目录, 例如打印机、邮件、新闻等假脱机目录、日志文件、格式化后的手册页以及一些应用程序的数据文件等等;
- **/tmp**: 存放公用临时文件目录。

- 3. Linux的Shell
 - Shell是系统的用户界面，提供用户与内核进行交互操作的一种接口。
 - Shell是命令解释器
 - Shell是编程语言

- 在Ubuntu 操作系统中，提供了Terminal、Xterm、UXTerm三种终端

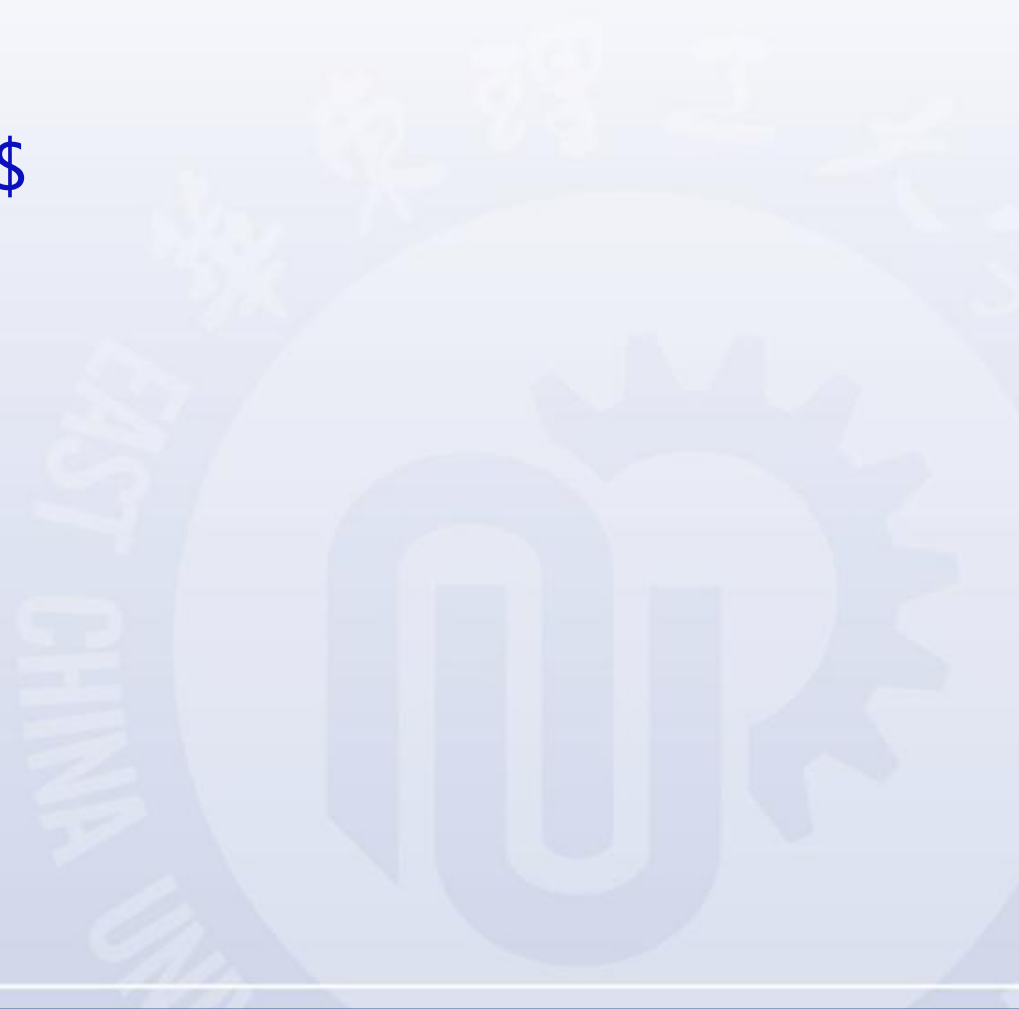


- 打开Terminal，并输入查看/etc目录的命令：

ls /etc/

- Linux是多用户系统，用户分为根用户（系统管理员）和普通用户两大类
 - 根用户（系统管理用）：又称为超级用户，用户账号为root，在系统中拥有最高权限，主要负责系统的管理工作
 - 普通用户：由根用户创建，命名时不能以数字和下划线作为第一个字符。普通用户是系统的使用者，只在自己的目录下工作，没有系统管理权限

- shell : 命令解释程序
- 普通用户
 - sh, **bash**, ksh 提示符 \$
 - csh 提示符 %
- root 用户
 - 提示符 #



命令格式

`[root@localhost root]# command [-option] parameter1 parameter2`

命令

参数选项

参数1

参数2

eg1: `[root@localhost root]#cp -r /mnt/hgfs/share/EELiod /`
eg2: `[root@localhost root]#tar xvfz xxx.tar.gz`

- 1. **ls**命令：列出当前文件夹下的文件

```
guest-3tttzl@ubuntu: ~  
guest-3tttzl@ubuntu:~$ ls  
Desktop      Downloads      Music          Public         Videos  
Documents    examples.desktop Pictures        Templates  
guest-3tttzl@ubuntu:~$
```

- 2. **--help**选项
 - **--help**选项放在命令之后，用来显示命令的信息。

```
guest-3tttzt@ubuntu:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                do not ignore entries starting with .
  -A, --almost-all        do not list implied . and ..
  --author                 with -l, print the author of each file
  -b, --escape             print C-style escapes for nongraphic characters
  --block-size=SIZE        scale sizes by SIZE before printing them; e.g.,
                           '--block-size=M' prints sizes in units of
                           1,048,576 bytes; see SIZE format below
  -B, --ignore-backups     do not list implied entries ending with ~
  -C                       with -lt: sort by, and show, ctime (time of last
                           modification of file status information);
                           with -lt: show ctime and sort by name;
```

- 3. **mkdir**命令:在当前工作区域内创建文件夹

```
guest-3tttzl@ubuntu:~$ mkdir aaa
guest-3tttzl@ubuntu:~$ ls
aaa      Documents  examples.desktop  Pictures  Templates
Desktop  Downloads  Music             Public    Videos
guest-3tttzl@ubuntu:~$
```


- 4. **cd**命令：进入某一个文件夹
 cd ..命令：回到上一级目录

```
guest-3tttzl@ubuntu:~$ mkdir aaa
guest-3tttzl@ubuntu:~$ ls
aaa      Documents  examples.desktop  Pictures  Templates
Desktop  Downloads  Music             Public    Videos
guest-3tttzl@ubuntu:~$ cd aaa
guest-3tttzl@ubuntu:~/aaa$
```


- 5. `pwd`命令：打印当前工作区文件路径

```
guest-3tttzl@ubuntu:~$ mkdir aaa
guest-3tttzl@ubuntu:~$ ls
aaa      Documents  examples.desktop  Pictures  Templates
Desktop  Downloads  Music             Public    Videos
guest-3tttzl@ubuntu:~$ cd aaa
guest-3tttzl@ubuntu:~/aaa$ pwd
/tmp/guest-3tttzl/aaa
guest-3tttzl@ubuntu:~/aaa$
```

- 6. cp file1 file2命令
 - 把file1复制到file2

```
guest-3tttzl@ubuntu: ~/bb
guest-3tttzl@ubuntu:~$ cp aaa/a.txt bb/
guest-3tttzl@ubuntu:~$ cd bb/
guest-3tttzl@ubuntu:~/bb$ ls
a.txt  b.txt
guest-3tttzl@ubuntu:~/bb$
```

- 7. **rm**命令：删除文件

```
guest-3tttzl@ubuntu:~/bb$ rm a.txt
guest-3tttzl@ubuntu:~/bb$ ls
b.txt
guest-3tttzl@ubuntu:~/bb$
```

- 8. **cat**命令：查看文件中的内容

```
guest-3tttzl@ubuntu:~/bb$ cat b.txt  
aaaaa:  
guest-3tttzl@ubuntu:~/bb$
```

- 9. **echo**命令

- 用于回显输入内容
- 格式: **echo strings**

```
guest-3tttzi@ubuntu: ~/bb
guest-3tttzi@ubuntu:~/bb$ echo hello word
hello word
guest-3tttzi@ubuntu:~/bb$
```

- 10. **clear**命令

- clear命令清除shell窗口中的内容
- 格式: **clear**

• 11. su命令

- 用户进入到系统后，切换到其他用户，使用**su**命令。
- 退回到原来用户使用**exit**命令
- 若想成功切换需要切换用户账号的密码。

```
zbb@ubuntu: ~  
zbb@ubuntu:~$ su root  
Password:  
root@ubuntu:/home/zbb# exit  
exit  
zbb@ubuntu:~$
```



文件操作命令-查找文件和目录命令

- **格式:** find [路径名...] [表达式]
- **说明:** find命令用于查找符合条件的文件和目录。**路径名**是用空格隔开的要搜索文件的目录名清单，**表达式**包含要寻找的文件的匹配规范或说明。表达式是从左向右求值的，只要表达式中的测试结果为真，进行下一个测试。



查找文件和目录命令

常用表达式:

表 2.2 find 命令常用表达式

表达式	说明
-amin n	查找系统中最后 n 分钟曾被访问过的文件或目录
-anewer<参考文件或目录>	查找其存取时间较指定文件或目录的存取时间更接近现在的文件或目录
-atime n	查找系统中最后 n*24 小时被访问过的文件或目录
-cmin n	找系统中最后 n 分钟被更改的文件或目录



查找文件和目录命令

-cnewer<参考文件或目录>	查找其更改时间较指定文件或目录的更改时间更接近现在的文件或目录
-ctime n	查找系统中最后 n*24 小时被改变状态的文件
-depth	从指定目录下最深层的子目录开始查找
-daystart	从本日开始计算时间
-exec<执行指令>	假设 find 指令的回传值为 True，就执行该指令
-fstype<文件系统类型>	只寻找该文件系统类型下的文件或目录
-gid<群组识别码>	查找符合指定群组识别码的文件或目录
-group<群组名称>	查找符合指定之群组名称的文件或目录
-inum<inode 编号>	查找符合指定的 inode 编号的文件或目录



查找文件和目录命令

-mmin n	查找在 n 分钟内曾被更改过的文件或目录
-mtime n	查找在 n 天内曾被更改过的文件或目录
-name<范本样式>	指定字符串作为寻找文件或目录的范本样式
-path<范本样式>	指定字符串作为寻找目录的范本样式
-perm<权限数值>	查找符合指定的权限数值的文件或目录
-print	假设 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称之前皆有"/"字符串
-prune	不寻找字符串作为寻找文件或目录的范本样式



查找文件和目录命令

-mmmin n	查找在 n 分钟内曾被更改过的文件或目录
-mtime n	查找在 n 天内曾被更改过的文件或目录
-name<范本样式>	指定字符串作为寻找文件或目录的范本样式
-path<范本样式>	指定字符串作为寻找目录的范本样式
-perm<权限数值>	查找符合指定的权限数值的文件或目录
-print	假设 find 指令的回传值为 True，就将文件或目录名称列出到标准输出。格式为每列一个名称，每个名称之前皆有"/"字符串
-prune	不寻找字符串作为寻找文件或目录的范本样式

-size<文件大小>	查找符合指定的文件大小的文件
-type<文件类型>	只寻找符合指定的文件类型的文件。b - 块设备文件；d - 目录；c - 字符设备文件；p - 管道文件；l - 符号链接文件；f - 普通文件
-uid<用户识别码>	查找符合指定的用户识别码的文件或目录
-user<拥有者名称>	查找符合指定的拥有者名称的文件或目录



查找文件和目录命令

- 示例：
- (1) 查找当前用户主目录下的所有文件。

```
[user@localhost ~]$ find ~ -print  
/home/user  
/home/user/exittest.c  
/home/user/exittest  
/home/user/greeting.c~  
/home/user/.ssh  
/home/user/.xsession-errors
```



查找文件和目录命令

- (2) 查找当前用户主目录下最近3天存取过的文件。

```
[user@localhost ~]$ find ~ atime 3 -print  
/home/user  
/home/user/exittest.c  
/home/user/exittest  
/home/user/greeting.c~  
/home/user/.ssh  
/home/user/.xsession-errors  
/home/user/tt.c  
/home/user/pc.c
```



查找文件和目录命令

- (3) 查找当前目录下扩展名为c的文件并显示。

```
[user@localhost ~]$ find . -name "*.c" -print
./exittest.c
./tt.c
./pc.c
./switch.c
./gdbtest.c
./fork1.c
./bad.c
./sigtest2.c
./bubble.c
```



查找文件和目录命令

- (4) 查找当前目录下权限为755的文件，即文件属主可以读、写、执行，其他用户可以读、执行的文件。

```
[user@localhost ~]$ find . -perm 755 -print
./下载
./tt
./文档
./音乐
./config
./config/gnome-disk-utility
./config/gnome-disk-utility/ata-smart-ignore
./config/gnome-session
./config/gnome-session/saved-session
./nautilus
./gnome2
./gnome2/gedit
./gnome2/nautilus-scripts
```




查找文件和目录命令

- (5) 查找当前目录下文件属主为user的文件并显示。

```
[user@localhost ~]$ find . -user user -print
.
./exittest.c
./exittest
./greeting.c~
./ssh
./xsession-errors
./pc.c
./.gtk-bookmarks
```



文件操作命令 - 文件的压缩和备份

- 1) bzip2命令
- 格式: bzip2 [选项] [要压缩的文件]
- 说明: .bz2文件的压缩程序, 并删除原始的文件。
- 常用选项:
 - -c 或--stdout: 将压缩或解压缩的结果送到标准输出。
 - -d或--decompress: 执行解压缩。
 - -f 或--force: 压缩或解压缩时, 若输出文件与现有文件同名, 覆盖现有文件。
 - -k 或--keep: bzip2 在压缩或解压缩后, 保留原始文件。



文件的压缩和备份

- 示例：
- (1) 压缩当前目录下的文件memo.1。

```
[user@localhost test]$ bzip2 memo.1
```

用ls -l命令查看压缩后的文件。

```
[user@localhost test]$ ls -l
总用量 4
-rw-rw-r--. 1 user user 63  9月  7 10:33 memo.1.bz2
```



- (2) 解压缩当前目录下的文件memo.1.bz2，并保留原始文件。

```
[user@localhost test]$ bzip2 -d -k memo.1.bz2
```

用ls-l命令查看。

```
[user@localhost test]$ ls -l
总用量 8
-rw-rw-r--. 1 user user 21  9月  7 10:33 memo.1
-rw-rw-r--. 1 user user 63  9月  7 10:33 memo.1.bz2
```



文件的压缩和备份

- 2) gzip命令
- 格式: gzip [选项] [文件 ...]
- 说明: gzip压缩会产生“.gz”的压缩文件, 并删除原始文件。
- 常用选项:
 - -c或--stdout或--to-stdout: 压缩后文件输出到标准输出设备, 不改变原始文件。
 - -d 或--decompress 或--uncompress: 解开压缩文件。
 - -f或--force: 强行压缩文件。
 - -n 或--no-name: 压缩文件时, 不保存原来的文件名称及时间戳记。
 - -N 或--name: 压缩文件时, 保存原来的文件名称及时间戳记。
 - -r 或--recursive: 递归处理, 将指定目录下的所有文件及子目录一并处理。



文件的压缩和备份

- 示例:
- (1) 压缩当前目录下的文件memo.1。

```
[user@localhost test]$ gzip memo.1
```

用ls -l命令查看压缩后的文件。

```
[user@localhost test]$ ls -l
总用量 4
-rw-rw-r--. 1 user user 46  9月  7 10:33 memo.1.gz
```



文件的压缩和备份

- (2) 解压缩当前目录下的文件memo.1.gz。

```
[user@localhost test]$ gzip -d memo.1.gz
```

用ls-l命令查看。

```
[user@localhost test]$ ls -l
总用量 4
-rw-rw-r--. 1 user user 21  9月  7 10:33 memo.1
```



文件的压缩和备份

- (3) 压缩当前目录下test1目录中的所有文件。

```
[user@localhost test]$ gzip -r test1
```

改变目录到test1下，用ls-l命令查看文件，
该目录下的文件tt被压缩成tt.gz。

```
[user@localhost test]$ cd test1  
[user@localhost test1]$ ls  
tt.gz
```




文件的压缩和备份

- 3) tar(tape archive)命令
- 格式: tar [选项...] [文件或目录]...
- 说明: tar 是用来建立, 还原备份文件的工具程序, 它可以加入, 解开备份文件内的文件。
- 常用选项:
 - -A 或--catenate: 新增tar文件到已存在的备份文件。
 - -c或--create: 建立新的备份文件。
 - -f<备份文件>或--file=<备份文件>: 指定备份文件。
 - -t或--list: 列出备份文件的内容。
 - -v或--verbose: 显示指令执行过程。
 - -x 或--extract 或--get: 从备份文件中还原文件。
 - --delete: 从备份文件中删除指定的文件。



文件的压缩和备份

- 示例:
- (1) 将当前目录下的文件memo.1、memo.2、memo.3备份到文件memo.tar中，并显示备份指令执行过程。

```
[user@localhost test]$ tar -cvf memo.tar memo.1 memo.2 memo.3  
memo.1  
memo.2  
memo.3
```



文件的压缩和备份

- 用`ls -l`命令查看当前目录，生成了备份文件`memo.tar`。

```
[user@localhost test]$ ls -l
总用量 24
-rw-rw-r--. 1 user user    21  9月  7 10:33 memo.1
-rw-rw-r--. 1 user user    21  9月  8 10:04 memo.2
-rw-rw-r--. 1 user user    21  9月  8 10:04 memo.3
-rw-rw-r--. 1 user user 10240  9月  8 10:07 memo.tar
```



文件的压缩和备份

- (2) 将当前目录下的备份文件memo.tar还原。
先删除当前目录下的文件memo.1、memo.2、memo.3。

```
[user@localhost test]$ rm memo.1 memo.2 memo.3
[user@localhost test]$ ls -l
总用量 16
-rw-rw-r--. 1 user user 10240  9月  8 10:07 memo.tar
```



文件的压缩和备份

将备份文件memo.tar还原。

```
[user@localhost test]$ tar -xvf memo.tar  
memo.1  
memo.2  
memo.3
```



文件的压缩和备份

用 `ls -l` 命令查看当前目录，memo.tar 被还原成文件 memo.1、memo.2、memo.3。

```
[user@localhost test]$ ls -l
```

总用量 28

-rw-rw-r--.	1	user	user	21	9月	7 10:33	memo.1
-rw-rw-r--.	1	user	user	21	9月	8 10:04	memo.2
-rw-rw-r--.	1	user	user	21	9月	8 10:04	memo.3
-rw-rw-r--.	1	user	user	10240	9月	8 10:07	memo.tar



文件的压缩和备份

- (3) 将当前目录下的文件memo.4备份到文件memoapp.tar中，并新增memoapp.tar到memo.tar中。
将文件memo.4备份到文件memoapp.tar中。

```
[user@localhost tt]$ tar -cvf memoapp.tar memo.4  
memo.4
```



文件的压缩和备份

用tar命令列出备份文件memoapp.tar的内容。

```
[user@localhost tt]$ tar -tvf memoapp.tar  
-rw-rw-r-- user/user          21 2011-09-08 10:32 memo.4
```

用tar命令列出备份文件memo.tar内容。

```
[user@localhost tt]$ tar -tvf memo.tar  
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.1  
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.2  
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.3
```




文件的压缩和备份

将备份文件memoapp.tar新增到备份文件memo.tar中，并用tar命令列出新增后的备份文件memo.tar内容。

```
[user@localhost tt]$ tar -Avf memo.tar memoapp.tar
[user@localhost tt]$ tar -tvf memo.tar
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.1
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.2
-rw-rw-r-- user/user          21 2011-09-08 10:14 memo.3
-rw-rw-r-- user/user          21 2011-09-08 10:32 memo.4
```



文件的压缩和备份

(4) 用bzip2命令将备份文件memo.tar压缩。

```
[user@localhost tt]$ bzip2 memo.tar
```

用ls-l命令查看当前目录，备份文件memo.tar被压缩成文件memo.tar.bz2。

```
[user@localhost tt]$ ls -l  
总用量 4  
-rw-rw-r--. 1 user user 190  9月  8 10:36 memo.tar.bz2
```



文件的压缩和备份

- (5) 用gzip命令将备份文件memo.tar压缩。
压缩备份文件memo.tar。

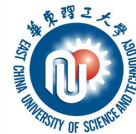
```
[user@localhost tt]$ gzip memo.tar
```

用ls-l命令查看当前目录，备份文件memo.tar被压缩成文件memo.tar.gz

```
[user@localhost tt]$ ls -l
总用量 4
-rw-rw-r--. 1 user user 216  9月  8 10:36 memo.tar.gz
```



改变文件或目录存取权限命令

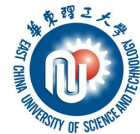


- 1) 用户和用户组

- Linux系统是一个多用户多任务的分时操作系统，任何一个要使用系统资源的用户，都必须首先向系统管理员申请一个账号，然后以这个账号的身份进入系统。Linux内部用UID标识各用户。
- 用户组（group）就是具有相同特征的用户集合体，Linux系统中每个用户都属于一个用户组，系统能对一个用户组中的所有用户进行集中管理。在Linux内部用户组用GID标识。
- Linux系统规定了4种不同类型的用户，分别是文件主、同组用户、其他用户、超级用户。



改变文件或目录存取权限命令



- 2) 存取权限

存取权限就是用来确定谁可以通过何种方式对文件和目录进行访问，Linux系统规定3种访问文件和目录的方式。

- (1) 读 (r)

对文件表示只允许指定用户读取该文件的内容，禁止做任何更改操作；对目录表示可以列出存储在该目录下的文件。

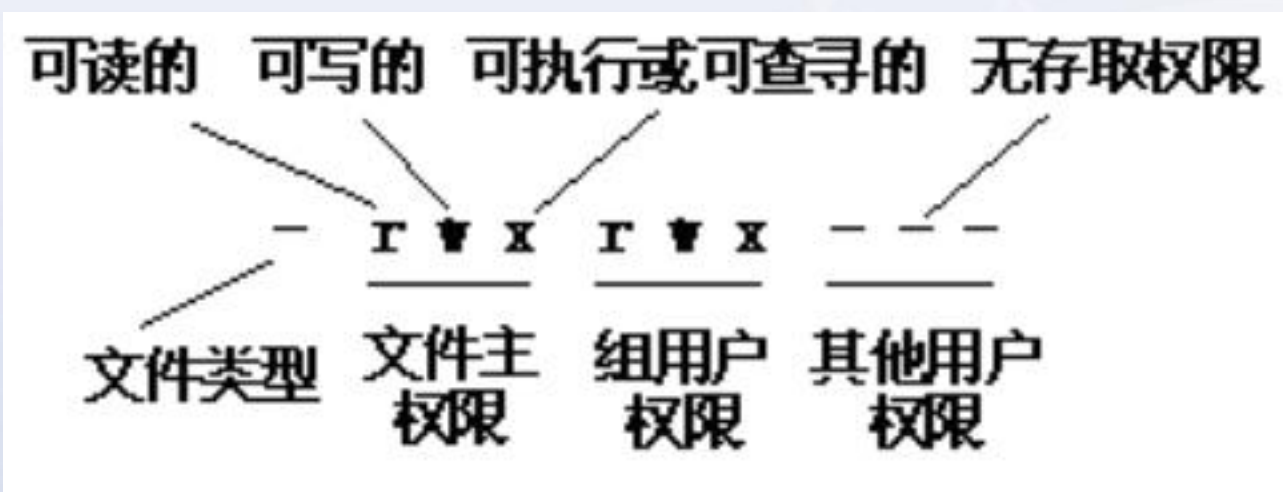
- (2) 写 (w)

对文件表示允许指定用户打开并修改该文件；对目录表示允许从该目录中删除或添加新的文件。

- (3) 执行 (x)

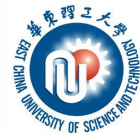
对文件表示允许指定用户执行该文件；对目录表示允许在该目录中进行查找，能用cd命令将工作目录改为该目录。

- 用 `ls -l` 命令可以显示文件或目录的详细信息，其中第一字段即为文件属性字段





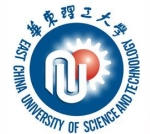
改变文件或目录存取权限命令



- 3) chmod命令
- **格式:** chmod [who] [opt] [mode] 文件或目录名...
- **说明:** 用于改变文件或目录的访问权限，是Linux系统管理员最常用到的命令之一。其中who表示对象，是以下字母中的一个或多个的组合。
 - **u:** 表示文件所有者。
 - **g:** 表示同组用户。
 - **o:** 表示其它用户。
 - **a:** 表示所有用户。



改变文件或目录存取权限命令

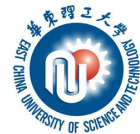


opt表示操作，可以为如下符号。

- **+**: 添加某个权限。
- **-**: 取消某个权限。
- **=**: 赋予给定的权限，并取消原有的权限。



改变文件或目录存取权限命令



mode表示权限，是以下字母一个或多个的组合。

- **r**: 可读。
- **w**: 可写。
- **x**: 可执行。



用数字设定法改变文件或目录的访问权限。

- **格式:** `chmod [mode] 文件或目录名...`
- **说明:** 将`rwX`看成二进制数, 如果该位有权限, 用1表示, 该位没有权限用0表示, 那么`rwX r-X r- -`可以表示为: 111 101 100, 再将其每三位转换成为一位八进制数, 就是754。



表 2.3 常见权限数字表示表

权限	对应二进制	八进制
<code>rwX rw- rw-</code>	111 110 110	766
<code>rw- r-- r--</code>	110 100 100	644
<code>rwX r-X r-X</code>	111 101 101	755
<code>r-- --- ---</code>	100 000 000	400
<code>r-X r-- r--</code>	101 100 000	540



- 示例:
- (1) 为当前目录下文件test.txt的主人同组用户增加读写权限。

用ls-l命令查看文件test.txt改变前权限，和文件主人user同组用户对文件test.txt没有权限。

```
[user@localhost ~]$ ls -l test.txt  
-rw-----. 1 user user 19 Aug 14 03:54 test.txt
```



用`chmod`命令为同组用户增加读写权限。

```
[user@localhost ~]$ chmod g+rw test.txt
```

用`ls-l`命令查看文件`test.txt`改变后的权限，和文件主人`user`同组用户增加了读写权限。

```
[user@localhost ~]$ ls -l test.txt  
-rw-rw---- 1 user user 19 Aug 14 03:54 test.txt
```



(2) 将当前目录下文件test.txt的权限改变为文件主可以读和写，同组用户可以执行，其它用户无权访问。

```
[user@localhost ~]$ chmod u=rw,g=x test.txt
```

用ls-l命令查看文件test.txt修改后的权限。

```
[user@localhost ~]$ ls -l test.txt  
-rw---x--. 1 user user 19 Aug 14 03:54 test.txt
```



- (3) 用数字权限方式将当前目录下文件test.txt的权限设为：文件主人可读写、同组用户可读写、其它用户只读。

权限串为“rw-rw-r--”，转换成二进制数是110 110 100，每三位转换成为一八进制数，结果为664。

```
[user@localhost ~]$ chmod 664 test.txt
```



用ls-l命令查看文件test.txt修改后权限。

```
[user@localhost ~]$ ls -l test.txt  
-rw-rw-r-- 1 user user 16 Aug 14 06:50 test.txt
```




- 1) chgrp
- **格式:** chgrp [选项]… 所属组 文件或目录…
- **说明:** chgrp命令改变指定文件所属的用户组，其中所属组可以是用户组的ID，也可以是用户组的组名。文件是以空格分开的要改变属组的文件列表，支持通配符。在Linux下一般只有超级用户才能改变该文件的属组。
- **常用选项:**
 - -R或--recursive: 递归式地改变指定目录及其下的所有子目录和文件的属组。



- 示例:
- (1) 将当前目录下文件test.txt的用户组改为teacher。

```
[root@localhost user]# chgrp teacher test.txt
```

用ls-l命令查看文件test.txt修改后的工作组。

```
[root@localhost user]# ls -l test.txt  
-rw-rw-r--. 1 user teacher 16 Aug 14 06:50 test.txt
```



- (2) 将目录/home/user/dest及其子目录下的所有文件的用户组改为teacher。

```
[root@localhost dest]# chgrp -R teacher /home/user/dest
```

用ls-l命令查看目录/home/user/dest修改后的工作组。

```
[root@localhost dest]# ls -l /home/user/dest
total 16
-rw-rw-r--. 1 user teacher 79 Aug 14 04:44 memo.1
-rw-rw-r--. 1 user teacher 20 Aug 13 03:13 memo.1~
-rw-rw-r--. 1 user teacher 20 Aug 13 03:13 memo.2
drwxrwxr-x. 3 user teacher 4096 Aug 14 04:25 test
```



- 2) chown命令
- **格式:** chown [选项] ... [用户][:[组]] 文件 ...
- **说明:** chown命令将指定文件的拥有者改为指定的用户或组。用户可以是用户名或用户ID。组可以是组名或组ID。文件是以空格分开的要改变权限的文件列表，支持通配符。在linux下一般只有超级用户才可以使用该命令。
- **常用选项:**
 - -R或--recursive: 递归式改变指定目录及其下的所有子目录和文件的拥有者。



- 示例:
- (1) 将当前目录下文件test.txt的所有者改为root。

```
[root@localhost user]# chown root test.txt
```

用ls-l命令查看文件test.txt修改后的所有者。

```
[root@localhost user]# ls -l test.txt  
-rw-rw-r--. 1 root teacher 16 Aug 14 06:50 test.txt
```

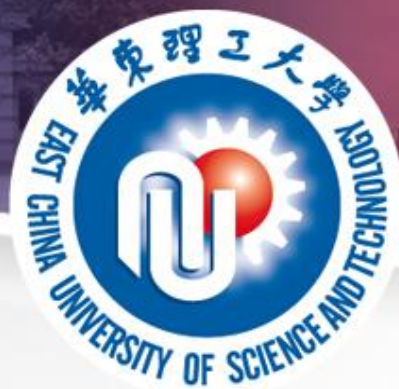


- (2) 将目录 `/home/user/dest` 及其下所有文件和目录的所有者改为root。

```
[root@localhost dest]# chown -R root /home/user/dest
```

用`ls-l`命令查看修改后`/home/user/dest`目录下文件和目录的所有者。

```
[root@localhost dest]# ls -l /home/user/dest
total 16
-rw-rw-r--. 1 root teacher 79 Aug 14 04:44 memo.1
-rw-rw-r--. 1 root teacher 20 Aug 13 03:13 memo.1~
-rw-rw-r--. 1 root teacher 20 Aug 13 03:13 memo.2
drwxrwxr-x. 3 root teacher 4096 Aug 14 04:25 test
```



THANKS!