条件表达式 p65

引号 p63

**第一章 Unix 系统概述**

重点:

Unix 的发展及其分支

Linux 系统的诞生 Android 是基于 Linux 内核的操作系统

GPL许可证协议

Linux 发行版的组成

**第 2 章 Shell 命令**

重点:

1. 根目录结构 p9

2. 文件和目录的操作 p11

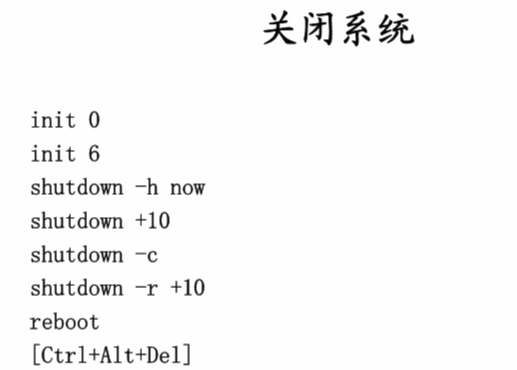
3. 用户和用户组 p16

4. 文件的权限管理 p19

5. 进程管理 p24

程序是保存在磁盘可执行映像中的机器代码指令和数 据的集合。进程可以被认为是执行中的计算机程 序。

Linux是多进程操作系统，每个程序启动时，可以创建 一个或几个进程，与其他程序创建的进程共同运行 在Linux内核空间。



6. 软件包管理 p32

7. Linux 系统的备份与恢复 p28

8. I/O 重定向和管道 p33

9. 元字符与正规表达式 p35

难点:

1. 决定进程是否能访问某个文件的因素。

进 程 的 凭 证 、 文 件 对 三 类 用 户 的 权 限 定 义 以 及 文 件 的 set-user-id ， set-group-id,Sticky 标记，其中，进程凭证包括**进程所属用户、用户组和附加用户组**。P20

2. 进程创建新文件和目录的默认权限的计算. P23

使用 open 和 mkdir 函数创建文件和目录所设置的 mode 值和进程的权限掩码。

目录的默认权限值 = 目录默认权限基准值 (rwx rwx rwx) 减去 UMASK权限位值

3. 在文本过滤器 grep 命令中使用元字符 p36

4. I/O 重定向和管道

5. 基于文件备份与恢复的命令 tar

6. 进程、进程组、会话、控制终端、作业、前台进程和后台进程之间的关系

**1）进程可以被认为是执行中的计算机程序。** Linux是多进程操作系统，每个程序启动时，可以创建一个或几个**进程**，与其他程序创建的进程共同运行在Linux内核空间。

**2）**Linux使用**会话**和**进程组**管理多用户进程，当用户在某个终端上登录时，系统创建一个新的**会话**，以会话ID的形式记录在进程中。一个会话中有一个**领头进程**和终端相连，负责从终端上接收输入。

**3）**将完成某种**作业**的相关进程定义为**进程组**，以进程组ID的形式记录在进程中。每个进程组中包含一个**领头进程**，领头进程的进程ID等于进程组ID。

**4）**一个终端会话可包含多个进程组，根据进程运行方式的不同，可将进程分为**前台进程和后台进程**。

**5）**前台进程和后台进程都称为**作业**，有前台、后台和挂起三种状态。Shell可使作业在这三种状态之间迁移。

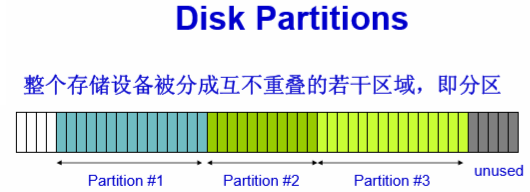
**6）终端**用于系统与用户间的交互，键盘作为终端的输入，终端显示器作为终端的输出。当会话的领头进程打开一个终端之后, 该终端就成为会话的**控制终端**。控制终端和会话是一对一的关系。

**第 3 章 Linux 系统的定制**

重点:

1. 磁盘结构 p39盘片 盘面 磁头 磁道 柱面 扇区

2. 磁盘分区的概念、格式和工具 p39

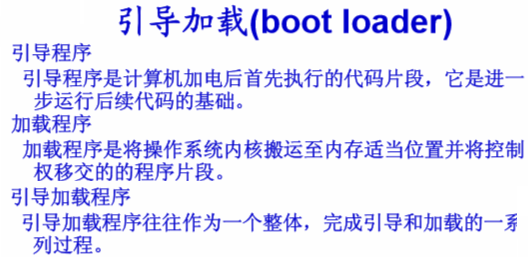


3. 磁盘分区的命名规则 p39

4. 文件系统的格式化 p41

5. 文件系统的挂载和卸载 p44

6. 引导加载程序 grub 的原理和配置 p46



GRUB引导配置：

etc/grub.conf 或者是 /boot/grub/menu.lst

内核安装过程已经在GRUB启动引导器的配置文件 “grub.conf”中对新内核的引导进行了适当的配置. P48

7. Linux 系统的启动过程 p45

难点:

1. 块设备、分区表与文件系统的关系

· 分区表记录（描述）了整个磁盘的分区情况，比如分区的起始位置，大小，类型。（MBR）

·而文件系统则是将某一分区格式化，“制作”成一种格式标准。

·块设备是是按块进行读写的随机存取设备。从块设备文件的外部表现来看，通常Linux内核将块设备文件存放在/dev目录下，用户像对常规文件一样来对其进行访问。从内部实现来看，在类Unix操作系统中，设备文件是一个设备驱动程序的界面，并被存储在文件系统中，就像常规文件.

**注：**文件系统参考p103

2. Linux 系统的引导过程

**第 4 章 Shell 程序设计** 熟练掌握和理解本章中的实例

重点:

1. shell 的概念、特点与种类 p56

2. shell 脚本的编写 p57

3. shell 变量 p58

4. shell 的控制语句 p65

难点:

1. 环境变量、位置变量和预定义变量

配置环境变量

可通过用户主目录下的配置文件，改变系统的启动参数,例如:可通过修改.bash\_profile修改环境变量PATH,具体操作如下:

$ cd ~ //进入用户的主目录

$ vi .bash\_profile //编辑配置文件

若文件中有下列行:

PATH= $PATH;$HOME/bin

则在PATH路径中增加当前路径,修改后外:

PATH= $PATH;$HOME/bin:.

$ exit //退出

2. 条件测试 p65

3. 条件语句和循环语句 p68, p73

**第5章 GNU C 开发环境** 熟练掌握和理解本章中的实例

重点:

1. 程序的编译和链接 p82

2. GCC 的概念和内容 p83

3. 项目管理工具 make p85

4. 静态函数库和共享函数库的概念、创建和使用 p93、p94，p96

5. Linux 环境下的 glibc 的特点 p98

难点:

1. 项目管理文件 Makefile 的编写 p87

2. 静态库和共享库的创建和使用

3. 应用程序、glibc 和 Linux 内核的关系 p99

**第 6 章 Linux 文件与目录** 熟练掌握和理解本章中的实例

重点:

1. 文件的基本 I/O 操作 p105

2. 文件的属性操作 p111

3. 目录操作 p116

4. 标准 I/O 函数库 p120

5. I/O 重定向 p129

**第 7 章 Linux 信号** 熟练掌握和理解本章中的实例

重点:

1. 信号的概念和分类 p133

2. 信号的处理过程 p136

3. 信号处理函数的设计原则 p139

4. 信号的发送 p143

5. 进程睡眠 p146

6. 间隔计时器 p150

难点:

1. 信号的处理过程

2. 信号处理函数的设计原则

3. 间隔计时器的类型

**第 8 章 Linux 进程** 熟练掌握和理解本章中的实例

重点:

1. 进程虚拟地址空间的结构 p156

2. 创建子进程 p162

3. 加载可执行程序 p167

难点:

1. 进程虚拟地址空间的结构

2. fork 函数

**实验五：**

1. 利用make工具进行项目管理有什么好处?

Make工具可以将编译和链接的步骤按一定的规则写入文本文件，而且当某个文件被修改，make工具只执行依赖于该文件的一系列规则，节约时间。

2. 进一步研究make工具对变量和隐含规则的处理.

Make工具支持四种变量：

①自定义变量：大小写敏感，变量名= 字符串 在Makefile中变量无数据类型。

②环境变量：make在运行过程中，将环境变量转化为同名同值的make变量，用户也可在Makefile中对这些变量进行重新定义。

③预定义变量：GNU make预定义了一些变量，在Makefile文件中可以直接使用。

④自动变量：由make工具预先定义，具有特定的意义。

潜规则：

①隐含规则：GNU make定义了内置的隐含规则，在不给出产生目标文件的命令时，由make自动添加。

②后缀规则：定义了将具有某后缀的文件转换为具有另外一后缀的文件。

③模式规则：是对具体规则的进一步抽象。