

linux 操作系统基础

唐玄、马瑞祥、姚颂词

安徽兮克电子科技有限公司

2018.08.18



目录

- 1 系统安装
- 2 数据存储
- 3 程序控制
- 4 脚本编程
- 5 系统配置



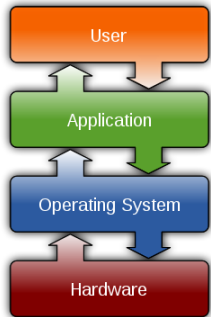
1 系统安装

- 基本术语
- 发展历史
- 安装流程
- 快速使用
- 帮助手册



什么是操作系统 I

操作系统：位于计算机硬件之上，方便用户使用计算机的接口程序



操作系统软件架构

Unix 操作系统 = Unix 系统内核及 Unix 实用工具

Linux 操作系统 = Linux 系统内核 + Linux 实用工具



操作系统软件架构——系统内核

- 系统内核：它是操作系统的核心，伴随着操作系统一直运行，并在需要时提供基本的服务
- 主要功能：
 - 内存管理
 - 进程管理
 - 设备管理
 - 文件管理
- linux 系统内核源码：<https://www.kernel.org/>



操作系统软件架构——实用工具

- 实用工具：方便用户管理与使用计算机的辅助程序
- 例：程序编辑器、数据库、文档处理软件等



1 系统安装

- 基本术语
- 发展历史
- 安装流程
- 快速使用
- 帮助手册



商业公司操作系统 I

- Microsoft
 - DOS
 - Windows 95/98/ME/2000/XP/Vista/7/8/8.1/10
 - Windows Server 2003/2008/2012/2016
 - Windows 10 S
- Apple
 - MAC OS X
 - macOS
 - iOS
- Google
 - Android
 - Chrome OS
 - Fuchsia (采用的比较新的 Magenta 内核)



商业公司操作系统 II

- Redhat
 - Red Hat Enterprise Linux Server (RHEL Server)
 - Red Hat Enterprise Linux Workstation (RHEL Workstation)
- Novell
 - Netware
 - SUSE



开源社区 linux 操作系统 I

linux 发行版排名: <http://www.distrowatch.org/>

- Manjaro

特点: 资源, 基于 Arch

<https://manjaro.org/>

- Mint

特点: 媒体, 基于 Ubuntu

<https://www.debian.org>

- Ubuntu

特点: 兼容, 基于 Debian

<https://www.debian.org>



开源社区 linux 操作系统 II

- Debian

特点：自由，deb 格式软件包，apt 包管理器

<https://www.debian.org>

- fedora

特点：激进，rpm 格式软件包，yum 包管理器，基于 RHEL Workstation

<https://www.getfedora.org>

- OpenSUSE

特点：华丽，基于 SUSE

<https://www.opensuse.org>

- Centos

特点：稳定，rpm 格式软件包，yum 包管理器，基于 RHEL Server

<https://www.centos.org>



开源社区 linux 操作系统 III

- Arch
特点：更新，pacman 包管理器
<https://www.archlinux.org>
- Gentoo
特点：定制，源码编译软件包
<https://www.gentoo.org>
- LFS
特点：专家，从 0 编译 linux 系统
<http://www.linuxfromscratch.org>



常见开源社区 BSD 操作系统

- FreeBSD

特点：高效，被 macOS 借鉴

<https://www.freebsd.org>

- NetBSD

特点：移植，被移植到多种硬件设备上

<http://www.netbsd.org>

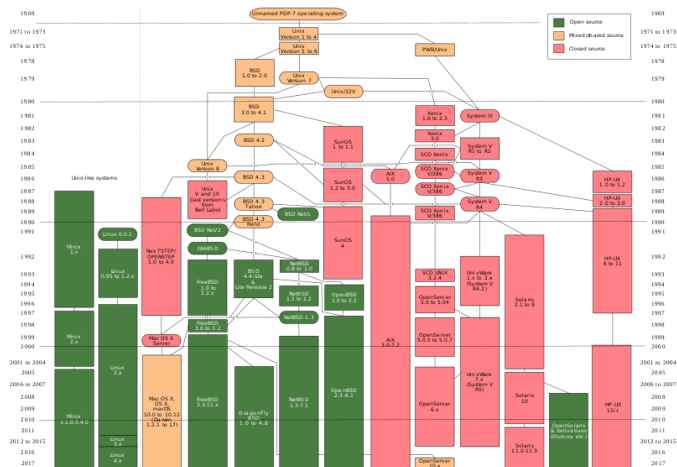
- OpenBSD

特点：安全，拒绝安装 FBI 后门，几乎没有安全漏洞

<http://www.openbsd.org>



操作系统发展历史



1 系统安装

- 基本术语
- 发展历史
- 安装流程
- 快速使用
- 帮助手册



安装 FreeBSD

- 网站

<https://www.freebsd.org/>

- 文档

<https://docs.freebsd.org/doc/>

- 安装 (v10.4)

[https:](https://docs.freebsd.org/doc/10.4-RELEASE/usr/local/share/doc/freebsd/zh_CN.UTF-8/books/handbook/bsdinstall.html)

[//docs.freebsd.org/doc/10.4-RELEASE/usr/local/share/doc/freebsd/zh_CN.UTF-8/books/handbook/bsdinstall.html](https://docs.freebsd.org/doc/10.4-RELEASE/usr/local/share/doc/freebsd/zh_CN.UTF-8/books/handbook/bsdinstall.html)



安装 Debian

- 网站

<https://www.debian.org/>

- 文档

<https://www.debian.org/doc/>

- 安装 (v8.x, jessie)

<https://debian-handbook.info/>



安装 Debian——准备

- 下载系统镜像：下载 amd64 架构 DVD-ISO 格式的系统镜像
- 刻录系统镜像（非虚拟机安装）：使用 win32 image writer(或 linux 下 dd 命令) 将系统镜像写入 U 盘
- 设置开机启动（非虚拟机安装）：开机按 F2（或 DEL、F12）进入 BIOS 并将 BOOT 优先级选项设置为 U 盘



安装 Debian——使用虚拟机 virtualbox

- 网站

<https://www.virtualbox.org/>

- 文档

<https://www.virtualbox.org/wiki/documentation/>

- 全版本下载

<http://download.virtualbox.org/virtualbox/>



安装 Debian——使用虚拟机 qemu

- 网站

<https://www.qemu.org/>

- 文档

<https://www.qemu.org/documentation/>

- 命令

```
qemu-img create sss.img 20G
```

```
qemu-system-x86_64 -enable-kvm -file sss.img -cdrom debian.iso  
-netdev user,id=net01 -device rtl8139,netdev=net01
```

```
qemu-img snapshot -c backup000 sss.img
```



安装 Debian——安装过程

- 设置语言、位置、本地化、键盘布局
- 检测 CDROM 并加载安装组件
- 检测与设置网络、主机名、域名
- 设置启用密码加密存储、启用 root 用户、root 用户密码、普通用户的用户名及密码
- 设置时钟
- 设置磁盘分区方案
- 安装基本系统
- 设置软件包管理工具
- 设置软件包方案
- 设置引导分区 (MBR 或 UEFI)



作业

使用 qemu 创建 debian 虚拟机，安装 Debian 操作系统，要求如下：

- 系统盘大小为 16G，预分配空间
- 添加 16 块硬盘，大小不限
- 添加 4 块虚拟网卡
- 安装完毕后，为系统盘创建快照
- 拓展内容
 - 配置软件源
 - 配置中文输入法



1 系统安装

- 基本术语
- 发展历史
- 安装流程
- 快速使用
- 帮助手册



登录注销

- 多用户登录（切换用户）

login [username]

su

- 多用户登录（切换终端）

Alt+F[1-6]

- 显示当前用户名

whoami

- 注销

exit



重启关机

- 重启

`reboot`

`shutdown -r now`

`systemctl reboot`

`init 6`

- 关机

`halt`

`shutdown -H now`

`systemctl halt`

- 关机并关闭电源

`poweroff`

`shutdown -h now`

`systemctl poweroff`

`init 0`



重启关机——运行级别

/sbin/init <runlevel>

- 0: Halt
- 1: Single-user mode
- 2: Multi-user mode
- 3: Multi-user mode with networking
- 4: Preserved
- 5: GUI
- 6: Reboot



1 系统安装

- 基本术语
- 发展历史
- 安装流程
- 快速使用
- 帮助手册



相关命令

- man
- whatis (man -f)
- apropos (man -k)
- which
- whereis

手册内容

- 名称 (name) !
- 语法 (synopsis) !
- 描述 (description) !
- 作者 (author) !
- 缺陷 (reporting bugs)
- 版权 (copyright) !
- 参考 (see also) !



2 数据存储

- 文件系统
- 目录文件
- 编辑工具
- 格式过滤



文件系统结构

- 文件系统层次结构标准
(fhs, Filesystem Hierarchy Standard)
- 网站: <http://www.pathname.com/fhs/>
- 特点: 树形结构

Directory	Description
bin	Essential command binaries
boot	Static files of the boot loader
dev	Device files
etc	Host-specific system configuration
lib	Essential shared libraries and kernel modules
media	Mount point for removable media
mnt	Mount point for mounting a filesystem temporarily
opt	Add-on application software packages
sbin	Essential system binaries
srv	Data for services provided by this system
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data



常用目录 I

- /: 根目录
- /bin: 基本程序
- /boot: 系统引导相关文件
- /dev: 设备文件
- /etc: 系统配置文件
- /home: 所有用户主目录的父级目录
- /lib: 基本共享库和内核模块
- /media: 可移动介质挂载点
- /mnt: 临时文件系统挂载点
- /opt: 第三方应用软件安装位置
- /proc: 进程目录, 存放现有硬件及当前进程的相关信息



常用目录 II

- /root: root 用户主目录
- /sbin: 基本系统程序
- /run: 运行目录
- /srv: 系统服务数据
- /sys: 系统目录
- /tmp: 临时文件
- /usr: 用户层次结构树形目录
- /var: 可变数据



特殊目录

- .: 当前目录
- ..: 父级目录
- ~: 用户主目录



文件路径

- 绝对文件路径

以根目录 (/) 作为起始目录

如执行命令 “/sbin/reboot” 时，搜索顺序为：

根目录——“sbin” 目录——“reboot” 文件

- 相对文件路径

以当前目录 (.) 作为起始目录

如执行命令 “../sbin/reboot” 时，搜索顺序为：

当前目录——父级目录——父级目录——“sbin” 目录——“reboot” 文件



2 数据存储

- 文件系统
- 目录文件
- 编辑工具
- 格式过滤



查看当前工作目录

- 显示当前目录（包括符号链接目录）

`pwd`

`pwd -L`

- 显示物理目录（不包括符号链接目录）

`pwd -P`

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
nrx@mrx:~/linux$ pwd
/home/mrx/linux
nrx@mrx:~/linux$ pwd -L
/home/mrx/linux
nrx@mrx:~/linux$ pwd -P
/home/mrx/linux
nrx@mrx:~/linux$ ls -l
总用量 4
lrwxrwxrwx 1 mrx mrx 4 4月 29 14:38 abc -> test
drwxr-xr-x 2 mrx mrx 4096 4月 29 14:38 test
nrx@mrx:~/linux$ cd abc
nrx@mrx:~/linux/abc$ pwd
/home/mrx/linux/abc
nrx@mrx:~/linux/abc$ pwd -P
/home/mrx/linux/test
nrx@mrx:~/linux/abc$
```



查看目录 I

- 默认显示当前目录下的文件和目录: `ls`
- 以详细列表格式显示: `ls -l`
 - 首项 1: 文件类型 (-普通文件 d 目录 l 链接 b 特殊文件 c 特殊文件 p 命名管道)
 - 首项 2: 文件权限
 - 第二项: 该文件有多少个链接的数字
 - 第三项: 所有者
 - 第四项: 所属组
 - 第五项: 文件大小
 - 第六 - 第八项: 最近一次修改的

```
mrx@mrx:~/linux$ ls
abc a.sh test
mrx@mrx:~/linux$ ls -l
总用量 4
lrwxrwxrwx 1 mrx mrx    4 4月 29 14:38 abc -> test
-rwxr-xr-x 1 mrx mrx    0 4月 29 14:54 a.sh
drwxr-xr-x 2 mrx mrx 4096 4月 29 14:53 test
mrx@mrx:~/linux$ ls -l
abc
a.sh
test
mrx@mrx:~/linux$ ls -F
abc@ a.sh* test/
mrx@mrx:~/linux$ ls -a
. .. abc a.sh test
mrx@mrx:~/linux$ ls -R
.:
abc a.sh test

./test:
abc
```



改变工作目录

- 切换指定目录

`cd ..`

`cd /bin`

`cd ../sbin`

- 切换到上一次目录

`cd -`

- 切换到用户主目录

`cd`

`cd ~`



创建目录

- 创建目录
`mkdir sss`
- 创建目录下的子目录
`mkdir sss/doc`
- 直接创建包含不存在目录在内的所有目录
`mkdir -p sss/bin/test`
- 以绝对路径方式创建目录
`mkdir /home/xk/sss/src`



复制、剪切、删除目录

- 复制目录

```
cp sss sss01
```

- 重命名、移动目录

```
mv sss sss02
```

- 删除空目录

```
rmdir sss01/doc
```

- 删除非空目录

```
rm -r sss01
```



创建文件

- 使用编辑器

nano readme01

vi readme02

gedit readme03

- 使用 touch

touch readme04 readme05

- 使用重定向技术

echo tang >readme06

cat >readme07

(退出: Ctrl+D)

ls >readme08



查看文件内容 I

- 查看单个或多个文件内容

```
cat /etc/passwd
```

```
cat -n /etc/passwd
```

```
cat -b /etc/passwd
```

```
cat -T /etc/passwd
```

- 查看指定行数

```
tail -n 5 /etc/passwd
```

```
head -n 5 /etc/passwd
```

- 分屏查看

```
more /etc/passwd
```

```
less -N /etc/passwd
```

```
less /etc/passwd /etc/group
```

(c) 切换文件 (c) 前一个文件

more 和 less 的故事

一群工程师在使用 more 查看错误时，发现一行内容显示出现错误，这个时候需要移动到代码的前几行，查看原因，然后不得不推出 more，重新启动它。一些人抱怨道：我们需要一个想后退的 more，另一个人说：是，我们需要 LESS，这使得每个人都发出的笑声



查看文件相关属性

- 查看文件基本类型、所有者、权限等属性
`ls -l readme01`
-: 普通文件, d: 目录, l: 链接, b: 块设备, c: 字符设备
r: 可读, w: 可写, x: 可执行

- 查看文件内容类型

`file /bin/ls`

`file /dev/sda`

`file /vmlinuz`

`file -b /etc/passwd`

`file -i install.log`

执行程序

`/bin/which which`



复制、剪切、删除文件

- 复制文件

```
cp readme01 readme02
```

- 重命名、移动文件

```
mv readme00 readme03
```

- 删除文件

```
rm -i readme02
```



访问权限

- 修改文件所有者、所属组

`chown tx:tx readme01`

`chgrp tx readme01`

- 修改文件权限

`rwX r-X r-X`

`421 421 421`

`421 4 1 4 1`

`chmod 644 readme01`

`chmod g+x,o+x readme02`

- 文件权限掩码

`umask`

`op_file = 0666 & ~mask`

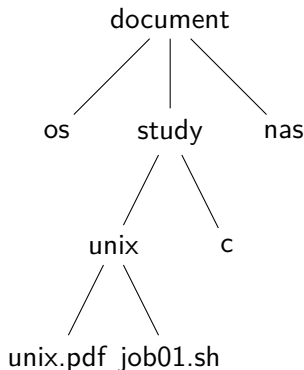
`op_dir = 0777 & ~mask`



作业

1 文件操作

- 1 创建右图所示目录结构，其中 `unix.pdf` 和 `job01.sh` 为文件，其他为目录
- 2 进入 `nas` 目录，创建文件 `readme`，并将该文件复制到 `os` 目录下
- 3 在 `nas` 目录中，给 `study` 目录创建子目录 `android`
- 4 将 `job01.sh` 权限改为可以所有用户均可以读取和执行，所有者和同组用户具有写权限



2 数据存储

- 文件系统
- 目录文件
- 编辑工具
- 格式过滤



常见文本编辑器

- vi, vim (<https://www.vim.org>)
编辑器之神
- emacs (<https://www.gnu.org/software/emacs>)
神之编辑器
- Visual Studio Code (<https://code.visualstudio.com>)
Microsoft
- Atom (<https://atom.io>)
GitHub
- Sublime Text (<https://www.sublimetext.com>)
(C)Sublime HQ Pty Ltd. Personal License: \$80



vi 能做什么

- 编辑文本
- 制作计算机程序
- 制作办公文档
- 制作电子表格
- 制作演示文稿
- 制作 2D 图形、图像
- 制作 3D 模型
-



vi 优点

- 资源占用低：图形化程序只有 2.4MB
- 操作速度快：不使用鼠标，全键盘操作
- 运行效率高：占用内存少
- 平台兼容好：Linux、Unix 默认安装，Windows、MAC OS X 全平台通用



使用 vi

- 无需安装

- 命令

vi [选项] [文件名]

例：vi +3 /etc/passwd



vi 的 3 种模式

启动后默认进入“命令模式”

- 命令模式 (command mode)
 - 提供复制、剪切等编辑功能
 - 按” i “、” a “、” o “进入“插入模式”
 - 按” : “进入”底行模式“
- 插入模式 (insert mode)
 - 提供输入文本功能
 - 按” ESC “进入”命令模式“
- 底行模式 (last line mode)
 - 提供设置、退出等管理功能
 - 按” ESC “进入”命令模式“



常用命令 - 模式切换

- i: 进入”插入模式“
- a: 进入”插入模式“，光标后移 1 位
- o: 进入”插入模式“，光标下方插入 1 行
- l: 进入”插入模式“，光标移至行首
- A: 进入”插入模式“，光标移至行尾
- O: 进入”插入模式“，光标上方插入 1 行



常用命令 -移动光标

- 0 或 \$: 光标移动到当前行开头或结尾
- ^: 光标移动到当前行第 1 个非空白字符
- gg 或 G: 光标移动到文件开头、结尾
- H、M、L: 光标移到屏幕开头、中间、结尾
- h、j、k、l: 光标向左、下、上、右移动 1 个字符
- w 或 W: 光标向右移动 1 个单词或 1 个以空白字符分隔的单词
- b 或 B: 光标向左移动 1 个单词或 1 个以空白字符分隔单词
- (或): 光标移到上一句或下一句
- {或}: 光标移到上一段或下一段
- » 或 «: 当前行向左或右缩进



常用命令 - 查找替换

- `/或?`: 向下或向上查找，按回车键确定
- `n 或 N`: 向下或向上查找下一个
- `:%s/x/y/g`: 将所有的 `x` 替换 `y`
- `:10,20s/x/y/g`: 将第 10 行到第 20 行之间所有的 `x` 替换 `y`
- `r`: 使用 1 个字符替换当前 1 个字符
- `s`: 使用 1 个或多个字符替换当前 1 个字符
- `c$`: 替换至当前行结尾
- `R`: 从当前字符开始替换多个字符
- `S 或 cc`: 替换当前行
- `C`: 从当前字符开始替换至当前行结尾



常用命令 -剪贴板

- dd: 剪切当前行
- d\$ 或 D: 剪切至行尾
- yy 或 Y: 复制当前行
- y\$: 复制至行尾
- p 或 P: 向后或向前粘贴



常用命令 - 撤销重做

- u: 撤销
- Ctrl + r: 重做
- .: 重复上一次命令
- 3dd: 重复删除 3 行
- 3iHello: 重复插入 3 次 Hello



常用命令 - 常用选项

- :set number
- :set encoding=utf-8
- :set fileencoding=utf-8
- :set fileformat=unix
- :set tabstop=8



常用命令 -其他操作

- :q: 退出
- :w: 保存
- :sh: 运行 shell
- :sp: 水平分屏显示文件
- :vsp: 垂直分屏显示文件
- Ctrl + w: 分屏切换



作业

- <http://www.vim-adventures.com>
- 使用 vi 编辑文章” test.sh “并执行该脚本
#!/bin/sh
sudo apt-get -y install espeak
espeak "the quick brown fox jumps over a lazy dog."
espeak -g 10 -p 10 -vzh " 中华人民共和国"



2 数据存储

- 文件系统
- 目录文件
- 编辑工具
- 格式过滤



显示输出

- 按行反序显示、按字符反序显示 `cat /etc/passwd /etc/hostname`
`tac`
`rev`
- 十六进制显示
`hexdump -C`
`od -x`
`xxd`



行列处理

- 合并文件

```
paste /etc/hostname /etc/passwd
```

- 删除列

```
colrm 2 4
```

- 提取列 (-d 指定分割符, -f 输出指定字段)

```
cut -d : -f 1-3 /etc/passwd
```

- 排序显示

```
sort -t : -k 3n /etc/passwd
```

- 无重复行输出

```
unique
```

- 按行连接两个文件

```
unique
```



制表符转换

- 制表符转为空格
expand
- 空格转为制表符
unexpand



统计

- 显示行号

`nl /etc/passwd`

- 计算行数

`wc -l /etc/passwd`



格式化

- 格式化列

`column -s : -t /etc/passwd`

- 格式化行 (-s 不分割单词, -w 指定宽度)

`fold -s -w 40 /etc/passwd`

- 格式化段 (将段落的各行连接在一起, 使得段落更紧凑)

`fmt`

- 格式化页

`pr`



查找替换

- 行查找

```
grep -Hn tx /etc/passwd
```

- 文件查找

```
find / -type c -name tx -perm 600
```

- 字符替换

```
apt-get moo|tr oo M  
tr a-z A-Z
```

- 行替换

```
sed "s:/?/g" /etc/passwd
```



比较

- 按字节比较文件

```
cmp -l test001 test002
```

- 比较两个已排序文件

```
comm -3 test001 test002
```

- 按行比较文件

```
diff -c test001 test002
```

```
sdiff test001 test002
```



作业

- 统计登录到系统用户的数量
- 统计/etc 下目录数量
- 搜索/etc 下所有和网络（net）有关的文件
- 显示所有用户的 ID、用户名和主目录
- 显示当前用户的所有组名，并使用冒号分隔



3 程序控制

- 基本概念
- 查看列表
- 作业控制
- 优先级别
- 守护进程



程序和进程

- 程序
静态的概念
- 进程
动态的概念
- 程序和进程的关系
程序 -运行-进程



3 程序控制

- 基本概念
- 查看列表
- 作业控制
- 优先级别
- 守护进程



学习使用 ps 查看

Table: 显示哪些进程

ps	与用户和终端相关进程
ps -a	与终端相关的进程
ps -e	所有进程
ps -p pid	与指定进程 ID 相关的进程
ps -u uid	与指定用户 ID 相关的进程



学习使用 ps 查看

Table: 显示哪些数据列

ps	PID TTY TIME CMD
ps -f	UID PID PPID C TTY TIME CMD
ps -F	UID PID PPID C SZ RSS TTY TIME CMD
ps -l	F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
ps -ly	S UID PID PPID C PRI NI RSS SZ WCHAN TTY TIME CMD



学习使用 ps 查看

Table: 进阶

ps -ef	显示所有用户进程，完整输出
ps -ax	显示所有进程
ps -aux	显示所有进程完整输出



显示内容 I

- PID
进程 id
- PPID
父进程 id
- C
处理器利用率
- STIME
累计系统时间
- %CPU
处理器利用率
- %MEM
真实内存使用率



显示内容 II

- CMD
命令的完整名称
- TIME
累计 CPU 时间
- COMMAND
正在执行的命令完整名称
- STAT(O,R,S,T,Z)
 - O 运行态
 - R 就绪态
 - S 睡眠态 (等待事件结束, 通常是 I/O)
 - T 挂起态
 - Z 僵尸态, 终止后, 父进程没有等待



显示内容 III

- NI
程序优先级



其他命令

- 为啥不使用 ps
ps 是静态的快照, 是瞬间的进程状态, 不能动态显示
- top 命令
显示顶端进程动态信息
- pstree 和 ptree
查看进程关系



3 程序控制

- 基本概念
- 查看列表
- 作业控制
- 优先级别
- 守护进程



基础命令

Table: 显示哪些进程

jobs	显示作业列表
fg	将作业移动至前台
bg	将作业移动至后台
Z	挂起程序
kill	杀死进程
echo \$\$	显示当前 shell 的 PID
echo \$!	显示上一条移动至后台的命令的 PID
ls > test	后台运行某程序



3 程序控制

- 基本概念
- 查看列表
- 作业控制
- **优先级别**
- 守护进程



设置优先级 nice

- 原理

给系统的程序调度器传输一个值来改变程序的级别

- 命令

`nice [-n adjustment] command`

- 使用

`nice gcc test.c` 这个命令并不会在后台运行, 后台运行需要加上 `&`

- 例子

`nice -n 1 test.sh`

Listing 1: 设置 nice 的脚本

```
nice -n 19 ./test_bg_a.sh &  
nice -n 18 ./test_bg_b.sh &  
ps -l
```



设置优先级 nice

- 命令

`renice niceness -p processid`

`niceness` 为 `nice` 值 `processid` 为进程 id



3 程序控制

- 基本概念
- 查看列表
- 作业控制
- 优先级别
- 守护进程



定义

- 何为守护进程

守护进程是一种后台静悄悄的运行的，不与任何终端相连，提供服务的进程

英文是 daemon

ps 查看 TTY 列中间问号字符就是不受终端控制的



作业

- 用 vi 打开一个文件，查看这个程序的进程，查看他的父进程
- 用 vi 打开一个文件，挂起这个程序，查看挂起操作的进程号，
- 尝试编写一个死循环程序，后台运行，设置他优先级最低，然后查看他的优先级



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



概念 I

- 什么是 shell

shell 是一个程序，充当了用户界面和脚本解释器，允许用户输入命令及间接访问内核的服务。

- shell 家族

Bourne Shell: sh, ksh, bash

C-Shell: csh, tcsh

- 改变 shell

临时改变 shell: tcsh

永久改变 shell: chsh



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



概念 I

- 什么是变量
变量是一个用来存储数据的实体
- 变量名限制
必须由大写字母、小写字母、数字、下划线构成
变量名第一个字符必须由字母或下划线构成，不能是数字
- shell 变量种类
环境变量：可以理解为全局变量，但是不完全相当于全局变量，子进程对全局修改不会影响父进程
shell 变量：局部变量



环境变量 I

- 显示
env、printenv
- 显示及使用全局变量
echo \$PWD
变量名第一个字符必须由字母或下划线构成，不能是数字
- Bourne Shel 设置一个全局变量
export:export XK=xike
unset:unset XK
- C-Shell 设置一个全局变量
setenv,unsetenv,set,unset



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



输入输出

- 标准输入 (0:stdin)
cat (Ctrl+D 结束标准输入)
read
- 标准输出 (1:stdout)
echo hello world
echo -n "hello world"(不换行输出)
echo -e "hello\n world"(允许转义)
- 标准错误 (2:stderr)



重定向

- 重定向标准输出

```
echo tangx > tangx.txt
```

```
echo .cc » tangx.txt
```

- 重定向标准输入

```
cat</etc/hostname
```

```
cat«EOF
```

- 重定向标准错误 `mkdir tangx.txt 2» tangx.txt`

- 重定向标准输出和标准错误 `(pwd;mkdir tangx.txt) » tangx.txt 2>&1`



管道

- 管道

< 命令 1> | < 命令 2>

命令 1 的标准输出作为命令 2 的标准输入

例: cal 2003|less

- 分流

< 命令 1> | tee [文件列表] | < 命令 2>

命令 1 的标准输出存入“文件列表”并作为命令 2 的标准输出

例: cal 2003|tee readme01|less



作业

- 显示环境变量，按照字母排序，还可以滚动显示
- 输出计算机当前的年份
- 设置一个全局变量，之后将所有变量输出到一个文件中，之后删除这个全局变量



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



顺序结构

- 顺序执行



test

Table: 数值比较

eq	相等
ge	大于或等于
gt	大于
le	小于等于
lt	小于
ne	不等于



字符串比较

Table: 字符串比较

<code>str1=str2</code>	相同
<code>str1!=str2</code>	不同
<code>str1<str2</code>	1 是否比 2 小
<code>str1<str2</code>	1 是否比 2 大
<code>-n str1</code>	1 是否长度非 0
<code>-z str1</code>	1 是否长度为 0



test |

Table: 文件比较

-d file	file 是否存在且为目录
-e file	file 是否存在
-f file	file 是否存在且为文件
-r file	file 是否存在且为可读
-s file	file 是否存在且不为空
-w file	file 是否存在且为可写
-x file	file 是否存在且为可执行
-O file	file 是否存在且属当前用户所有
-G file	file 是否存在且属当前用户的所属组所有
file -nt file2	file1 是否比 file2 新
file -ot file2	file1 是否比 file2 旧



if 语法 I

- if 语法

if command1

then

command set 1

elif command 2

then

command set 1

fi

- 符合条件

且 &&

或 ||



if 语法 II

Listing 2: if 判断

```
#!/bin/bash  
a=2  
b=2  
if test ${a} -eq ${b}  
then  
    echo "equal"  
fi
```



case I

- case 语法

case variable in

pattern1 | pattern2) commands1;;

pattern3) commands2;;

esac



case II

Listing 3: case 判断

```
#!/bin/sh
a=3
case $a in
3|4)
    echo "3或4";;
5)
    echo "5";;
esac
```



while

- while 语法

```
while test command 1  
do  
other command  
done
```

Listing 4: while 循环

```
#!/bin/sh  
i=0  
while [ ${i} -lt 10 ]  
do  
    echo "i:  $\square$  ${i}"  
    i=$(expr $i + 1)  
done
```



for

- for 语法
for var in list
do
commands
done

Listing 5: for 循环

```
#!/bin/sh  
for test in Asia Africa Europe Africa  
do  
    echo "1.2 $test"  
done
```



控制循环

- 跳出循环
break
- 继续循环
continue

Listing 6: continue

```
#!/bin/sh
i=0
while [ ${i} -lt 10 ]
do
    echo "i:_${i}"
    if [ ${i} -eq 5 ]; then
        break
    fi
```



作业

- 计算 1-100 的和
- 99 乘法表
- 输出斐波那契数列直到 1597(1, 1, 2, 3, 5, 8, 13)
- 将一目录下所有的文件的扩展名改为 bak
- 杨辉三角输出到第五行, 将结果倒着输出



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



命令参数

- 命令行参数
输入到命令行的所有参数
- 读取参数
\$+ 参数位置, 0 为程序名
- 参数总数
\$+#



其他

- \$+?

最后运行的命令的结束代码（返回值）即执行上一个指令的返回值



作业

- 判断/etc 文件夹下是否存在某个文件
- 写一个脚本
 - 1、创建一个组 newgroup, id 号为 4000;
 - 2、创建一个用户 mageedu1, id 号为 3001, 附加组为 newgroup;
 - 3、创建目录/tmp/hellodirxyz
 - 4、复制/etc/fstab 至上面的目录中
 - 5、改变目录及内部文件的属主和属组为 mageedu1;
 - 6、让目录及内部文件的其它用户没有任何权限;



4

脚本编程

- shell 基础
- 变量使用
- 输入输出
- 标准流程
 - 顺序结构
 - 选择结构
 - 循环
- 命令参数
- 函数模块



读取文件

- 读取文件

cat 文件名 | while read line

- 继续循环



期中总结

- 写一个脚本, 实现配置文件的增查改删, 需求如下
 - 1、从文件中读取增加内容, 动态增加到配置文件里
 - 2、读取配置文件"key" 的数量, 输入指定位置, 获得对应 value 值 (要将对应的 flag 分隔符当参数)
 - 3、输入三个参数: "key" 及新"value" 及 key 对应个数, 将内容完成修改
 - 4、删除对应的 key 和 value
 - 5、每次添加/修改/删除之前都要备份, 如果添加/修改/删除失败, 需要从之前备份的版本还原



5 系统配置

● 远程登录

● 区域语言

● 网络接口

● 主机域名

● 软件安装

● 用户数据

● 计划任务

● 备份同步

● 日志记录

● 启动引导

● 磁盘存储

● 虚拟容器

● 性能监视

● 系统安全

● 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



日期时间

- 显示当前日期时间

`date`

`date -u`

`date "+`

- 显示日历

`cal`

`cal 1982`



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- **主机域名**
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5

系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- **软件安装**
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



管理用户组

- 查看所有用户组
`less /etc/group`
- 创建用户组
`groupadd users -g 2018`
- 修改用户组：改名
`groupmod administrators -n admins`
- 删除用户组
`groupdel admins`



登录列表

- 显示已登录用户列表

users

who

w

- 显示最近登录列表

last

lastb

登录列表信息

注：

LOGIN (登录时间)

IDLE(空闲时间)

JCPU(该终端所有非当前后台进程
CPU 时间)

PCPU(该终端当前进程 CPU 时间)

WHAT(当前的命令)



作业

- 创建用户组 `guests`
- 创建用户 `guest` 加入 `guests` 用户组
- 删除用户组 `guest`
- 修改用户 `guest` 的密码为 `123456`



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- **计划任务**
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- **备份同步**
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5

系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- **日志记录**
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- **启动引导**
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- **磁盘存储**
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- **虚拟容器**
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- **性能监视**
- 系统安全
- 文件共享



5

系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全

● 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5

系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享



5 系统配置

- 远程登录
- 区域语言
- 网络接口
- 主机域名
- 软件安装
- 用户数据
- 计划任务
- 备份同步
- 日志记录
- 启动引导
- 磁盘存储
- 虚拟容器
- 性能监视
- 系统安全
- 文件共享

