

统信 UOS 系统 管理员手册

编写：培训认证部

版本：0.20.818



统信软件技术有限公司

目录

第一章 Linux 发展史	1
1.1、 Unix 操作系统简介	1
1.2、 GNU/GPL 的说明	2
1.3、 Linux 操作系统.....	3
1.4、 UOS 统一操作系统	6
第二章 UOS 服务器版本的安装	9
UOS 服务器版本用户界面	9
2.1 系统激活	9
2.2 虚拟机管理	11
2.3 UOS 系统安装.....	17
2.4 控制中心	22
第三章 命令与 Shell.....	25
3.1 shell 解释器	25
3.2 linux 控制台	26
3.3 linux 的基础命令	26
第四章 用户和组管理	31
4.1 用户管理	31
4.2 密码管理	36
4.3 组管理.....	36
第五章 文件属性与权限.....	39
概述:	39

5.1 文件类型的查看.....	39
5.2 文件权限	40
5.3 文件/文件夹的隐藏权限	41
5.4 文件访问控制列表与链接	42
第六章 目录管理.....	44
6.2 强制位 u+s.....	44
6.3 强制位 g+s.....	45
6.4 冒险位（或粘制位） o+t.....	45
6.5 umask：控制新建的文件或目录的权限	46
第七章 用户信息交互与 VI 编辑器.....	47
7.1 常用的交互工具.....	47
7.2 vim 编辑器：	47
7.3 修改 vim 配置	49
第八章 文件查找与压缩归档.....	50
8.1 文件查找.....	50
8.2 文件内容的查找.....	51
8.3 归档与压缩	51
第九章 输入输出重定向.....	53
9.1 重定向	53
9.2 管道.....	54
9.3 文件处理	54
第十章 软件包的安装与使用.....	55

10.1 使用 dpkg 管理软件包.....	55
10.2 使用 apt 管理软件包	55
10.3 获取 deb 包 （用外网提取 deb 包，内网安装）	56
10.4 源码包安装	56
第十一章 进程管理与系统监控.....	57
11.1 系统状态查询	57
11.2 进程管理.....	57
11.3 用 lsof 恢复文件	60
第十二章 服务与计划任务.....	61
12.1 一号进程 systemd.....	61
12.2 设置系统运行级别	61
12.3 systemctl 控制服务	61
12.4 ssh 远程及 scp 远程拷贝	62
12.5 ssh 免密码登录.....	62
12.6 cron 计划任务	62
第十三章 LINUX 的启动流程.....	64
13.1 Linux 的启动流程	64
13.2 破解 root 密码.....	64
13.3 防止破密码：增加 grub 口令	64
13.4 启动修复.....	65
第十四章 网络管理	66
14.1 网络连接管理	66

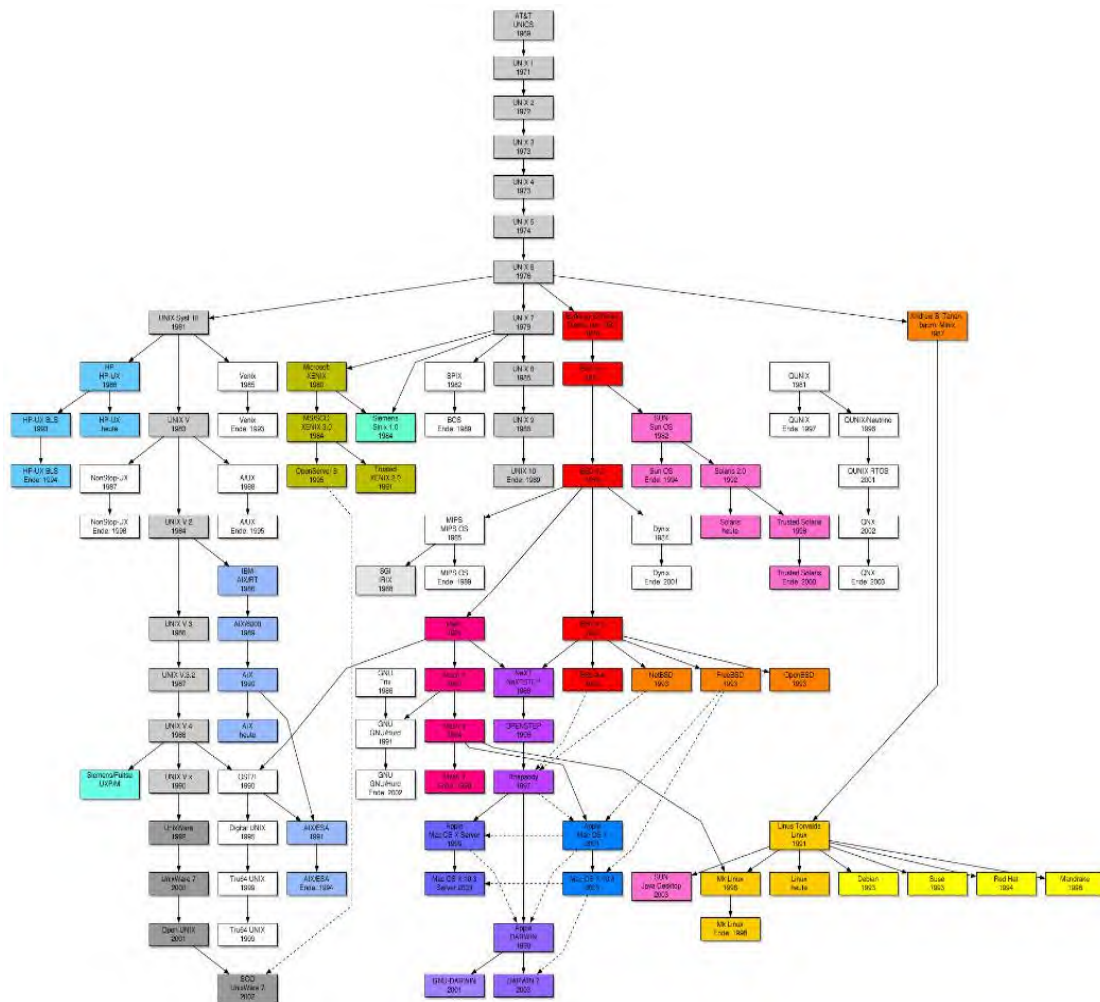
14.2 配置 ip 地址	67
14.3 ipv6 地址.....	68
第十五章 磁盘管理	69
15.1 基础理论.....	69
15.2 磁盘分区.....	70
15.3 swap 分区.....	71
15.4 lvm 逻辑卷.....	71
15.5 磁盘配额.....	72
15.6 raid.....	72

第一章 Linux 发展史

1.1、 Unix 操作系统简介

UNIX 历史

- UNIX 操作系统诞生于 60 年代末期的 Bell 实验室。
- 美国电话电报公司(AT&T)在 70 年代中期开始发行 UNIX 的非商业许可证。
- 从 70 年代末开始在市场上出现了不同的 UNIX 商品化版本。



UNIX 基本原则

- 所有对象都是以文件的方式体现；
- 配置数据以文本形式保存；
- 由短小的单目程序构成；
- 避免不必要的用户交互；
- 可使用多个程序合作完成复杂任务。



1.2、 GNU/GPL 的说明

GNU 项目介绍

- GNU Project 由 Richard Stallman 发起开始于 1984 年，由自由软件基金(FSF :Free Software Foundation)支持。GNU 是 GNU' s not UNIX 的缩写。

- GNU 的主旨在于发展一个类似 Unix，并且为自由软件的完整操作系统：GNU 系统。



GNU 通用公共许可证 (GPL)

一个法定的版权声明，但附带(或，在技术上去除了某些限制)，在条款中，允许对某项成果以及由它派生的其余成果的重用，修改和复制对所有人都是自由的。

非版权(copyleft)

copyleft 带有标准的 Copyright 声明，确认作者的所有权和标志。但它放弃了标准 copyright 中的某些限制。它声明：任何人不但可以自由分发该成果，还可以自由地修改它。但你不能声明你做了原始的工作，或声明是由他人做的。最终，所有派生的成果必须遵循这一条款(相当于继承关系)。

开源软件与自由软件

自由软件(Free SoftWare)

自由软件是关于自由以及社区的一场道德运动，是一种价值观，倡导者希望软件能够自由得使用，复制，修改，分发等等，有些共产主义的韵味，因此自由软件涉及到对与错的问题。

开源软件(Open SoftWare)

可以肤浅的认为开源软件=自由软件，他们几乎都开放源代码。开源软件的哲学并不涉及到对与错，它仅仅是一个实际的价值观。而自由软件是一种要求软件自由的文化现象，是一种哲理、伦理和社会价值观，而开源软件是一种开发模式

1.3、 Linux 操作系统

Linux 的起源

- Linux 操作系统可以说是 UNIX 操作系统的一个“克隆体”，
- Linux 的第一个内核(Kernel)编写于 1991/10/5。由 Linus Torvalds 于芬兰赫尔辛基大学发布的。



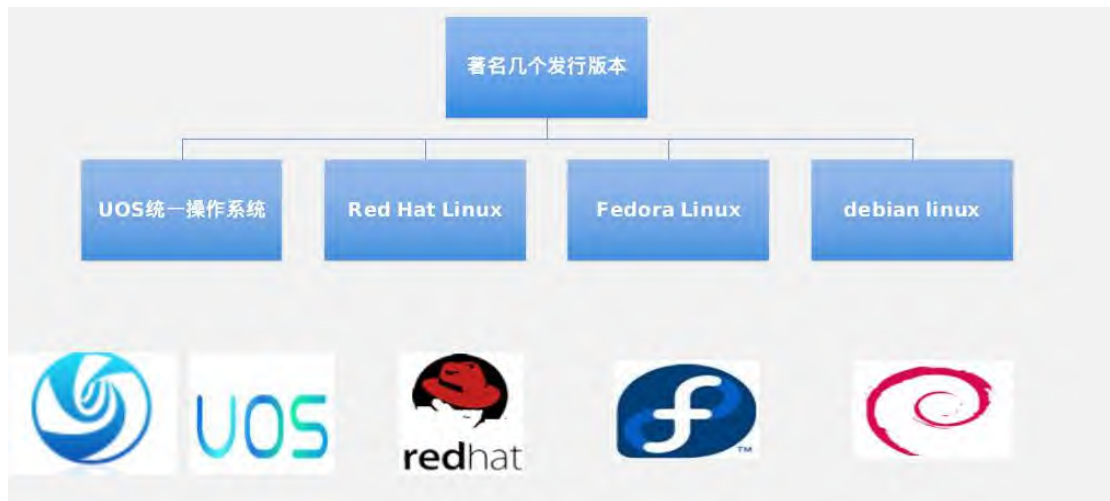
Linux 是一个自由的、遵循 GNU 和 GPL 原则，并且类似于 Unix 的一个的操作系统。

Linux 特点

- 遵循 GNU/GPL
- 开放性
- 多用户
- 多任务
- 良好的用户界面
- 供了丰富的网络功能
- 可靠的系统安全
- 良好的可移植性
- 稳定性



Linux 的发行版本 (1)



Linux 的发行版本 (2)



Linux 操作系统的展望

- 在面临着微软侵权说的强烈施压下 Linux 企鹅正以百折不挠的精神正在成长,更是在今年有很多人举手称赞 Linux 的进步。

- 他不仅仅是作为服务器的功能强大，在桌面上也有着突飞猛进的发展。
- 小企鹅 Linux 已经悄悄的进驻用户的电脑中，由于 Linux 标榜自由开放，所以我们即使在睡不醒的情况下也要相信 Linux 会成为近年内操作系统的新霸主。



1.4、 UOS 统一操作系统

产品概述



UOS统一操作系统是基于Linux内核,同源异构支持四种CPU架构(AMD64、ARM64、MIPS64、SW64)和六大CPU平台(鲲鹏、龙芯、申威、海光、兆芯、飞腾),提供高效简洁的人机交互、美观易用的桌面应用、安全稳定的系统服务,是真正可用和好用的自主操作系统。

UOS 通过对硬件外设的适配支持,对应用程序的兼容和优化,以及对应用场景解决方案的构建,完全满足项目支撑、平台应用、应用开发和系统定制的需求,体现了当今Linux操作系统发展的最新水平。

统一特性

1、统一的版本

同源异构,同一份源代码构建支持不同CPU架构的OS产品。

2、统一的支撑平台

UOS 桌面和服务器版产品提供统一的编译工具链,并提供统一的社区支持。

3、统一的应用商店和仓库

UOS 应用商店支持签名认证,提供统一安全的应用软件发布渠道。

CPU 厂商	CPU 架构	CPU 型号
--------	--------	--------

龙芯	MIPS64	龙芯 (3A3000/4000、3B3000/4000)
----	--------	------------------------------

申威	SW64	申威 (421、1621)
----	------	---------------

鲲鹏 ARM64 鲲鹏 (920s、916、920)

飞腾 ARM64 飞腾 (FT2000/4、FT2000/64)

海光 AMD64 海光 (31XX、51XX、71XX)

兆芯 AMD64 兆芯 (ZX-C、ZX-E 系列, KX、KH 系列)

Intel/AMD AMD64 主流型号 CPU

4、统一的开发接口

UOS 桌面版和服务器版产品提供统一版本的运行和开发环境，包括运行库、开发库、头文件；

应用开发厂商仅需在某 CPU 平台完成一次开发，即可在多种架构 CPU 平台完成构建。

5、统一的标准规范

UOS 符合规范的测试认证，为适配厂商提供高效支持，并提供软硬件产品的互认证。

6、统一的文档

UOS 桌面版和服务器版产品提供一致的开发文档、维护文档、使用文档，降低运维门槛。

核心版本	Kernel 4.19
	DDE 5.0
	Xorg 1.20.4.1
	Glibc 2.28
	GCC 8.3.0
开发库版本	JDK 11
	Qt 5.11.3

	Gtk 3.24.5
--	------------

完善的技术支持与售后服务



官方网址：

<https://www.uniontech.com/>

合作伙伴：

统信软件技术有限公司 ©版权所有





<https://www.chinauos.com/>

服务热线:

400-858-8488

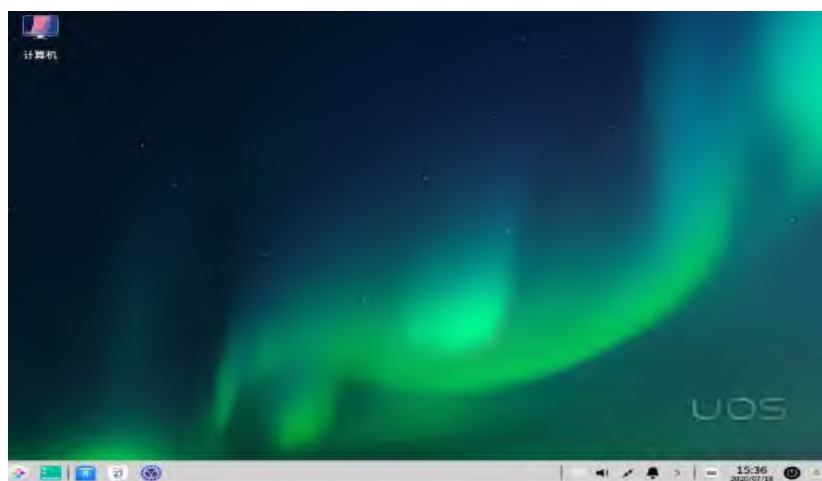
售后服务:

support@uniontech.com

第二章 UOS 服务器版本的安装

UOS 服务器版本用户界面

桌面环境



UOS 最小化安装的命令行界面

```
uos-PC login: root
Password:
Last login: Thu Jul 16 15:36:27 CST 2020 on tty2
Welcome to uos 20 GNU/Linux

 * Homepage: https://www.chinuos.com/
 * Bugreport: https://bbs.chinuos.com/

root@uos-PC:~#
root@uos-PC:~#
root@uos-PC:~#
root@uos-PC:~#
```

2.1 系统激活

系统激活

激活步骤：控制中心→系统信息→**激活**

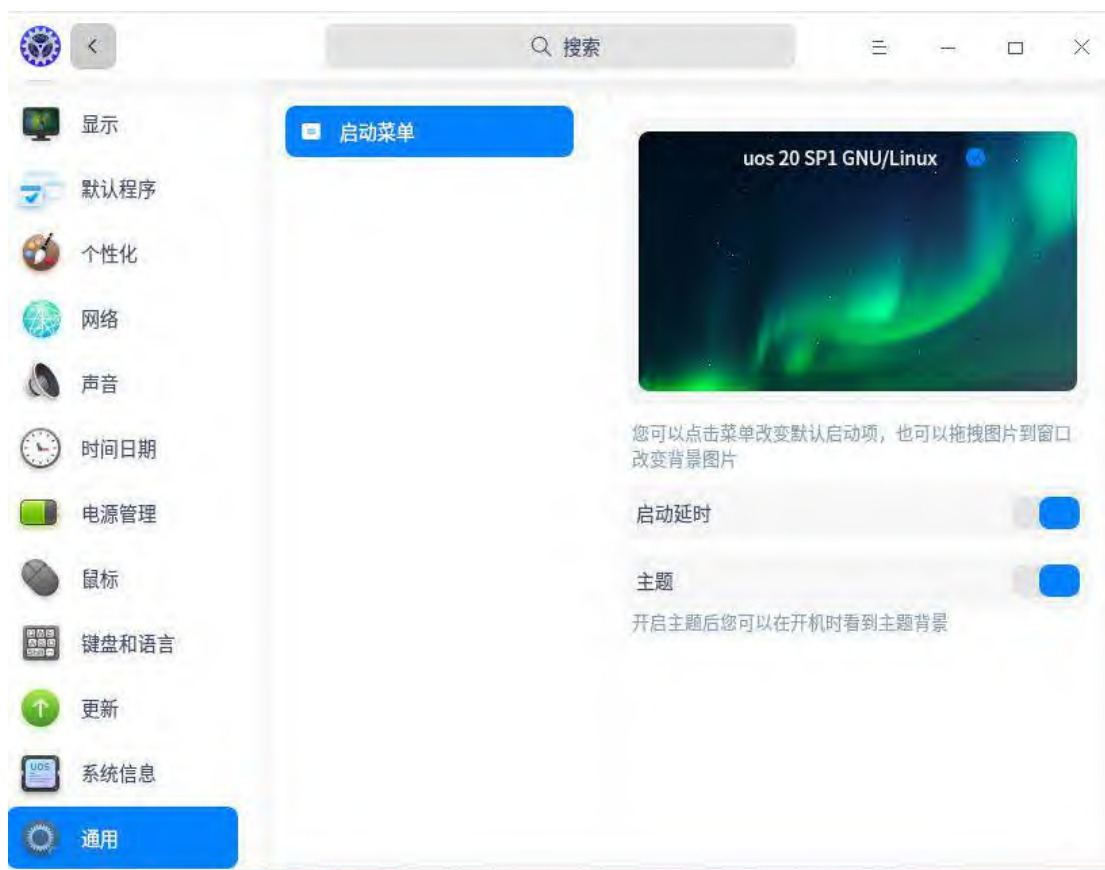


试用期激活：确认就可以激活只能使用 90 天。

激活码和激活文件就得需要通过官方渠道获取进行激活了。



服务器版本不需要开启开发者权限



2.2 虚拟机管理

KVM 是一个开源的系统虚拟化模块，uos 系统默认支持该虚拟化工具，只需要简单的命令行安装就可以使用，KVM 会将我们的系统资源充分利用，虚拟出多个 uos 系统方便办公使用和测试

1、 安装 KVM

在桌面右键【在终端中打开】，这个就是 UOS 的命令行模式，不再使用鼠标点点，使用命令就能完成大部分操作。

更新 apt 源：使用 sudo 获取 root 权限执行命令：sudo apt update

```
uos@uos-PC: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
uos@uos-PC:~/Desktop$ sudo apt update

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：

#1) 尊重别人的隐私。
#2) 输入前要先考虑(后果和风险)。
#3) 权力越大，责任越大。

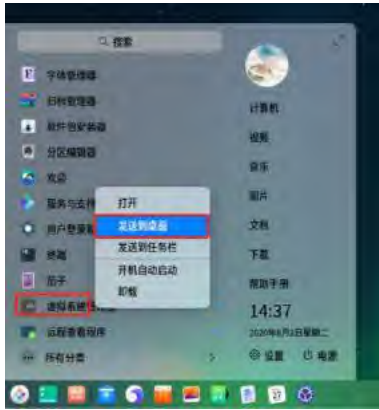
[sudo] uos 的密码：
获取:2 https://store.chinauos.com/appstore eagle InRelease [5,277 B]
命中:1 https://uos.deepin.cn/printer eagle InRelease
获取:3 https://uos.deepin.cn/uos eagle InRelease [20.0 kB]
获取:4 https://store.chinauos.com/appstore eagle/appstore amd64 Packages [359 kB]
获取:9 https://packages.chinauos.com/uos eagle/non-free amd64 Packages [206 kB]
获取:10 https://packages.chinauos.com/uos eagle/main i386 Packages [17.8 MB]
获取:11 https://packages.chinauos.com/uos eagle/contrib amd64 Packages [102 kB]
已下载 36.8 MB，耗时 8秒 (4,583 kB/s)
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
有 11 个软件包可以升级。请执行 'apt list --upgradable' 来查看它们。
uos@uos-PC:~/Desktop$
```

```
sudo apt -y install libvirt0 libvirt-daemon qemu virt-manager bridge-utils libvirt-clients
python-libvirt qemu-efi uml-utilities virtinst qemu-system
```

执行安装 KVM 命令：

```
uos@uos-PC: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
uos@uos-PC:~/Desktop$ sudo apt -y install libvirt0 libvirt-daemon qemu virt-manager bridge-utils libvirt-clients python-libvirt qe
mu-efi uml-utilities virtinst qemu-system
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件：
augeas-lenses genisoimage gir1.2-gtk-vnc-2.0 gir1.2-libosinfo-1.0 gir1.2-libvirt-glib-1.0
gir1.2-spiceclientglib-2.0 gir1.2-spiceclientgtk-3.0 gir1.2-vte-2.91 ipxe-qemu libaugeas0
libbrlapi0.6 libcacard0 libcapstone3 libfdt libgovirt-common libgovirt2 libgtk-vnc-2.0-0
libgunc-1.0-0 libnetcf1 libosinfo-1.0-0 libphodv-2.0-0 libphodav-2.0-common librdmacm1
libspice-client-glib-2.0-8 libspice-client-gtk-3.0-5 libspice-server1 libusbredirhost1
libusbredirparser1 libvdeplug2 libvirglrenderer0 libvirt-daemon-system libvirt-glib-1.0-0
libxencall1 libxendevicemodel1 libxenfvchn1 libxenforeignmemory1 libxengnttab1

正在处理用于 mine-support (3.62) 的触发器 ...
正在处理用于 hicolor-icon-theme (0.17-2) 的触发器 ...
正在处理用于 lastore-daemon (5.0.6-1) 的触发器 ...
正在处理用于 libglib2.0-0:amd64 (2.58.3-4+deepin) 的触发器 ...
正在处理用于 libc-bin (2.28-7-1+deepin) 的触发器 ...
正在处理用于 systemd (241.5+ct1+eagle) 的触发器 ...
正在处理用于 man-db (2.8.5-2) 的触发器 ...
正在处理用于 shared-mime-info (1.10-1) 的触发器 ...
正在处理用于 bamfdemon (0.5.4-1) 的触发器 ...
Rebuilding /usr/share/applications/bamf-2.index...
uos@uos-PC:~/Desktop$
```



启动器→虚拟系统管理器→右击发送到桌面

2、新建虚拟网络

编辑→连接详情→虚拟网络



default 是 KVM 安装时默认创建的虚拟网络

创建新的虚拟网络

创建虚拟网络

4 的步骤 1

为虚拟网络选择名称:

网络名称(N): public

① 示例: network1

取消(C) 后退(B) 前进(F)

创建新的虚拟网络

创建虚拟网络

4 的步骤 2

为虚拟网络选择 IPv4 地址空间:

☒ 启用 IPv4 网络地址空间定义

网络(N): 192.168.100.0/24

① 提示: 网络地址应在 IPv4 私有地址范围内选择。例如: 10.0.0.0/8 或 192.168.0.0/16

网关: 192.168.100.1

类型: ?

☒ 启用 DHCPv4

开始: 192.168.100.128

结束: 192.168.100.254

☐ 启用静态路由

取消(C) 后退(B) 前进(F)

创建新的虚拟网络

创建虚拟网络

4 的步骤 3

为虚拟网络选择 IPv6 地址空间:

☐ 启用 IPv6 网络地址空间定义

取消(C) 后退(B) 前进(F)

创建新的虚拟网络

创建虚拟网络

4 的步骤 4

连接到 物理网络:

☐ 隔离的虚拟网络(I)

☒ 转发到物理网络(W)

目的(D): 任意物理设备

模式(M): NAT

☐ Pool containing all of VFs of a SR-IOV device

Device List: No available device

☐ 启用 IPv6 内部路由/网络

如果 未 指定 IPv6 网络地址, 将会在虚拟机之间启用 IPv6 内部路由。默认启用 IPv4 内部路由。

DNS 域名: public

取消(C) 后退(B) 完成(E)

创建新的虚拟网络

创建虚拟网络

4 的步骤 4

连接到 物理网络:

☐ 隔离的虚拟网络(I)

☒ 转发到物理网络(W)

目的(D): 任意物理设备

模式(M): 物理设备 lo

☐ Pool containing all of VFs of a SR-IOV device

Device List: No available device

☐ 启用 IPv6 内部路由/网络

如果 未 指定 IPv6 网络地址, 将会在虚拟机之间启用 IPv6 内部路由。默认启用 IPv4 内部路由。

DNS 域名: public

取消(C) 后退(B) 完成(F)

创建新的虚拟网络

创建虚拟网络

4 的步骤 4

连接到 物理网络:

☐ 隔离的虚拟网络(I)

☒ 转发到物理网络(W)

目的(D): 任意物理设备

模式(M): NAT

☐ Pool containing all of VFs of a SR-IOV device

Device List: Open

☐ 启用 IPv6 内部路由/网络

如果 未 指定 IPv6 网络地址, 将会在虚拟机之间启用 IPv6 内部路由。默认启用 IPv4 内部路由。

DNS 域名: public

取消(C) 后退(B) 完成(F)

在步骤 4 中的网络模式：

- 隔离的虚拟网络 → Host Only:
- 转发到物理网络：
 - 目的：默认即可，通过本主机网卡转发
 - 模式：NAT、路由、Open
- SR-IOV：技术是一种基于硬件的虚拟化解决方案，可提高性能和可伸缩性

3、新建存储

编辑→连接详情→存储



Qcow2 优点：占用空间小、支持加密、快照、压缩等

4、新建虚拟机



第一步选择后，第二步浏览本地的 uos 系统镜像，手动选择正在安装的操作系统



选择合适的 cpu、内存、磁盘资源



最后确定虚拟机名称和加入的网络即可完成，会直接开启虚拟机进入到安装过程

2.3 UOS 系统安装

前面的操作为我们做了系统和虚拟化的铺垫,那么就体验一下通过虚拟化来安装操作系统

安装 UOS 系统，需要 2 核 cpu，2G 内存，最少 64G 存储的资源，网络选择 NAT

1、 安装 UOS 系统

打开新建的虚拟机，通过提示完成虚拟机系统的安装



选择安装基本环境（最小化，桌面版本），附加的服务器的软件



进入安装 UOS 20 desktop 系统，选择简体中文并勾选用户许可协议即可



全盘安装会分很多区，在使用时不是必须的分区，存储空间会被拆分，我们可以选择手

动分区，仅创建我们需要的分区即可，**先分/boot 分区**



/boot 分区: 2G

Swap 分区: 4G

/ 分区: 剩余全部

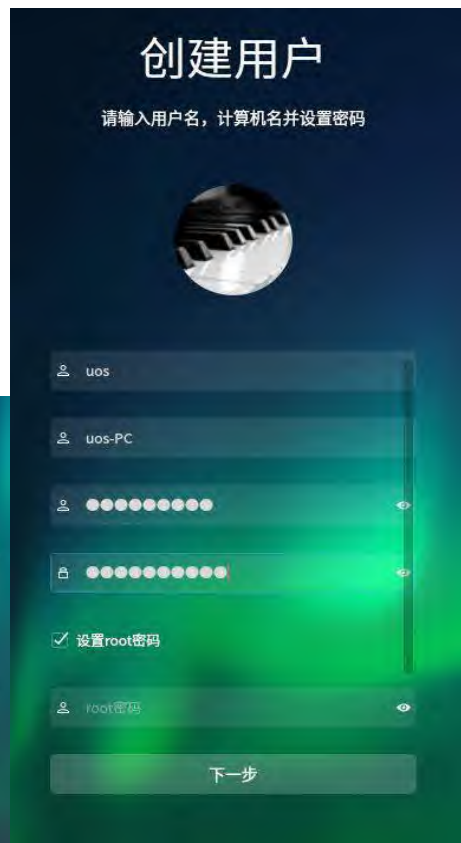
基本分区创建完成后确认分区继续安装系统，等待安装成功



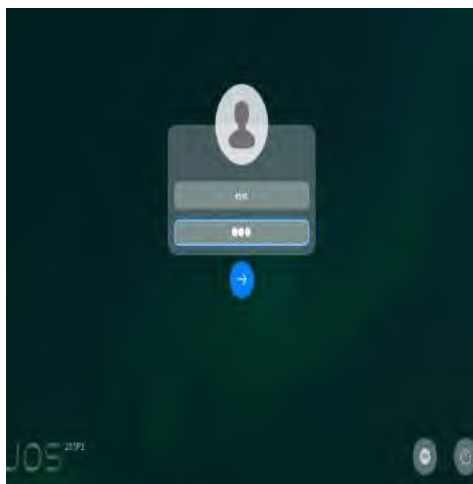
立即体验，选择 uos 20 启动

2、 初始化系统

系统启动后 选择时区(默认上海即可)



初始化,



创建使用用户→优化系统配置→登录体验新系统→根据机器性能选择特效模式



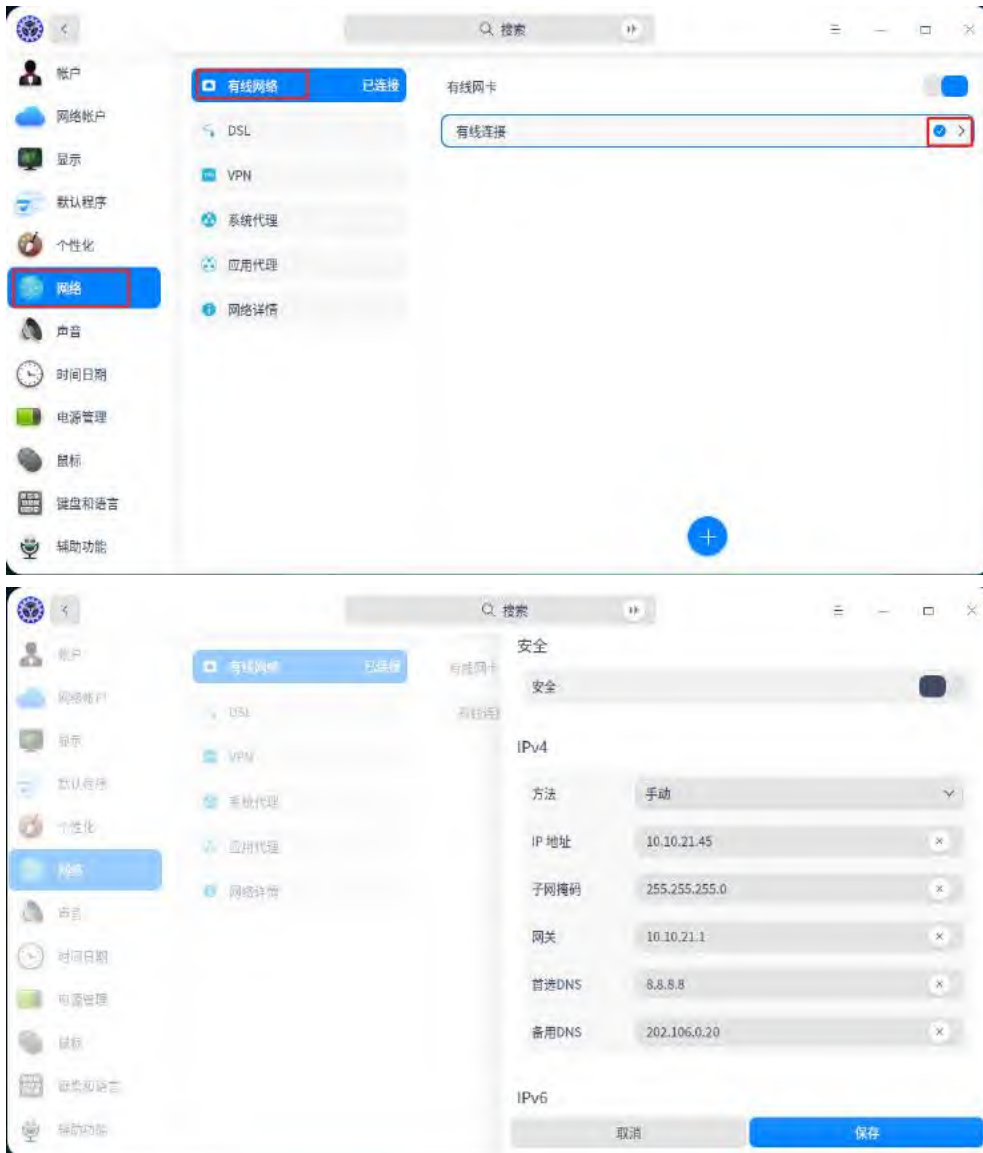
初始化完成，体验新系统

2.4 控制中心

控制中心是核心的一个功能，集中了 UOS 系统相关的配置，所需要的配置都能在这里找到并快速的进行配置。



1、网络配置



控制中心→网络→有线网络→有线连接

只需根据当前网路配置合适的 IPv4 地址保存即可

第三章 命令与 Shell

3.1 shell 解释器

1、Shell 解释器简介：

Shell 和 Shell 编程 不是一个概念；Shell 是操作系统的最外层，它可以合并编程语言来 控制进程和文件！从下图你可以清楚的看见，Shell 是负责 User 与 Linux OS 之间沟通的桥梁！Shell 为用户提供了一个操作界面，User 在这个界面输入指令，其实就是通过 Shell 向 Linux Kernel 传递过去，这也就是为什么 Shell 也叫 解释器的原因！



2、Linux shell 种类：

(1) Shell 的种类

bourne again shell (/bin/bash) —— 最常用

bourne shell (/use/bin/sh 或 /bin/sh)

C shell (/usr/bin/csh)

K shell (/usr/bin/ksh)

注意：我们最常用的 Shell 就是 /bin/bash ！它使用简单以及免费！也是大多数 Linux OS 默认的 Shell 环境！

还有就是，不同的 shell 语言的语法不同，一般不能交换使用！

(2) Shell 的命令

shell 命令分为两种：分别是内部命令和外部命令。

内部命令：在安装的时候嵌入系统内核。

外部命令：以文件的形式存在。

可以使用 type 命令查看是内部命令还是外部命令。

Linux 中，默认的 shell 是 bash

(3) 系统中的 shell

查看系统支持的 shell: cat etc/shells

系统默认的 shell: echo \$SHELL

3、Shell 、Shell 命令、Shell 编程、Shell 脚本的区别：

简单来说：

SHELL ： 是一个整体的概念！解释器！

SHELL 编程 and SHELL 脚本： 可以统称为 SHELL 编程！

SHELL 命令： 是 SHELL 编程 底层具体的语句 和 实现方法！

3.2 linux 控制台

1、控制台切换

(1)、terminal： 键盘输入， 显示器输出。

(2)、physical console： 键盘和显示器。

(3)、 virtual console： Linux 的一个功能， 物理控制台支持多个虚拟控制台，
使用 Ctrl+Alt+(F2~F6)

控制台终端 (/dev/tty)和虚拟终端(/dev/pts/n) ---远程终端

图形--字符 Ctrl+Alt+(F2~F6)

字符--字符 Alt+(F2~F6)

字符--图形 Alt+F1

w： 显示登陆终端信息

2、控制台的快捷键

(1)、查看虚拟机 ip 命令 ifconfig

(2)、在控制台下： 复制命令： Ctrl + Insert 组合键或用鼠标选中即是复制。粘贴命令： Shift
+ Insert 组合键或单击鼠标滚轮即为粘贴。

(3)、ctrl+l 清空屏幕或者 clear

(4)、ctrl+c 退出某个正在执行中的操作

(5)、ctrl+d 退出 shell 或者 exit 退出

(6)、ctrl+a 将光标移到行首

(7)、ctrl+e 将光标移到行尾

(8)、ctrl+u 删除光标前的字符

(9)、ctrl+k 删除光标后的字符

(10)、ctrl+w 删除光标前空格为界线的单词

(11)、ctrl+左右箭头 以单词为单位移动光标

(12)、ctrl+r 搜索历史命令

(13)、tab 键补全

3.3 linux 的基础命令

1、命令的帮助

(1)、man 帮助

man ls

(2)、help 查找命令的帮助

ls --help

2、基础命令

(1)、who 查看当前用户的登陆信息

(2)、pwd 查看当前的位置

语法格式: pwd [参数]

常用参数:

-L 显示逻辑路径

Pwd

(3)、cd 切换目录

cd [参数] [目录名]

常用参数:

- P 如果切换的目标目录是一个符号链接, 则直接切换到符号链接指向的目标目录
- L 如果切换的目标目录是一个符号链接, 则直接切换到符号链接名所在的目录
- 仅使用" - "选项时, 当前目录将被切换到环境变量" OLDPWD" 对应值的目录
- ~ 切换至当前用户目录
- .. 切换至当前目录位置的上一级目录

参考实例

cd / 切换到根目录

cd ~ 返回当前用户的家目录

Cd .. 返回上一级的目录

(4)、ls 列出当前目录下内容

语法格式: ls [选项] [文件]

常用参数:

- a 显示所有文件及目录 (包括以 "." 开头的隐藏文件)
- l 使用长格式列出文件及目录信息
- r 将文件以相反次序显示(默认依英文字母次序)
- t 根据最后的修改时间排序
- A 同 -a , 但不列出 "." (当前目录) 及 ".." (父目录)
- S 根据文件大小排序
- R 递归列出所有子目录

参考实例

ls -l 详细的列出

ls -a 列出所有

(5)、mkdir 创建目录

语法格式: mkdir [选项] 目录....

常用参数:

-m, --mode=模式, 设定权限<模式> (类似 chmod), 而不是 rwxrwxrwx 减 umask

-p, --parents 可以是一个路径名称。此时若路径中的某些目录尚不存在,加上此选项后,系统将自动建立好那些尚不存在的目录,即一次可以建立多个目录;

-v, --verbose 每次创建新目录都显示信息

--help 显示此帮助信息并退出

--version 输出版本信息并退出

参考实例:

创建一个空目录

```
mkdir test1
```

递归创建多个目录

```
mkdir -p /a/b/c/d
```

(6)、touch

语法格式

touch [参数] [文件]

常用参数：

-a 改变档案的读取时间记录

-m 改变档案的修改时间记录

-r 使用参考档的时间记录，与 --file 的效果一样

-c 不创建新文件

-d 设定时间与日期，可以使用各种不同的格式

-t 设定档案的时间记录，格式与 date 命令相同

--no-create 不创建新文件

--help 显示帮助信息

--version 列出版本讯息

参考实例

touch xing.txt 创建 txt 文件

touch a b c 创建多个文件

touch file{1..90}.txt 批量创建文本文件

(7)、cat 查看文档内容

语法格式

cat [-AbeEnstTuv] [--help] [--version] fileName

参数说明：

-n 或 --number：由 1 开始对所有输出的行数编号。

-b 或 --number-nonblank：和 -n 相似，只不过对于空白行不编号。

-s 或 --squeeze-blank：当遇到有连续两行以上的空白行，就代换为一行的空白行。

-v 或 --show-nonprinting: 使用 ^ 和 M- 符号, 除了 LFD 和 TAB 之外。

-E 或 --show-ends: 在每行结束处显示 \$。

-T 或 --show-tabs: 将 TAB 字符显示为 ^I。

参考实例:

查看内容

```
cat xing.txt
```

把 textfile1 的文档内容加上行号后输入 textfile2 这个文档里:

```
cat -n textfile1 > textfile2
```

(8)、cp 复制

语法格式

```
cp [参数] [文件]
```

常用参数:

-f 若目标文件已存在, 则会直接覆盖原文件

-i 若目标文件已存在, 则会询问是否覆盖

-p 保留源文件或目录的所有属性

-r 递归复制文件和目录

-d 当复制符号连接时, 把目标文件或目录也建立为符号连接, 并指向与源文件或目录连接的原始文件或目录

-l 对源文件建立硬连接, 而非复制文件

-s 对源文件建立符号连接, 而非复制文件

-b 覆盖已存在的文件目标前将目标文件备份

-v 详细显示 cp 命令执行的操作过程

-a 等价于 “dpr” 选项

参考实例

复制目录：

```
cp -R dir1 dir2/
```

将文件 test1 改名为 test2：

```
cp -f test1 test2
```

复制多个文件：

```
cp -r file1 file2 file3 dir
```

(9)、mv 移动或者剪切

语法格式

```
mv [参数]
```

常用参数：

-i 若存在同名文件，则向用户询问是否覆盖

-f 覆盖已有文件时，不进行任何提示

-b 当文件存在时，覆盖前为其创建一个备份

-u

当源文件比目标文件新，或者目标文件不存在时，才执行移动此操作

参考实例

将文件 file_1 重命名为 file_2:

```
mv file_1 file_2
```

将文件 file 移动到目录 dir 中 :

```
mv file /dir
```

将目录 dir1 移动目录 dir2 中 (前提是目录 dir2 已存在, 若不存在则改名):

```
mv /dir1 /dir2
```

将目录 dir1 下的文件移动到当前目录下:

```
mv /dir1/* .
```

(10) 、rm 删除

语法格式

```
rm [参数] [文件]
```

常用参数:

-f 忽略不存在的文件, 不会出现警告信息

-i 删除前会询问用户是否操作

-r/R 递归删除

-v 显示指令的详细执行过程

rm 删除文本文件

rm -r 删除文件

rm -f 强制删除

第四章 用户和组管理

4.1 用户管理

1、linux 用户管理命令

(1)useradd 添加用户命令

当使用 useradd 命令不加参数选项，后面直接跟所添加的用户名时，系统首先会读取配置文件/etc/login.defs 和/etc/default/useradd 文件中所配置的信息建立用户的家目录，并复制/etc/skel 中的所有文件（包括隐藏的环境配置文件）到新用户的家目录中。

语法格式:

useradd 选项 用户名

useradd -m 新用户名

-c 添加备注信息

-d 指定有效家目录

-g 指定用户组

-G 指定附加组

-n 取消以用户为名的组

-s 指定登录 shell

-u 指定用户 ID

例：

```
useradd -m tx
```

```
useradd -m -d /home/haha -s /bin/bash -u 1010 yanwj
```

```
id yanwj
```

(2) 删除用户

如果一个用户的账号不再使用，可以从系统中删除。删除用户账号就是要将/etc/passwd等系统文件中的该用户记录删除，必要时还删除用户的主目录。删除一个已有的用户账号使用 userdel 命令，其格式如下：

语法格式：

```
userdel 选项 用户名
```

常用的选项是-r，它的作用是把用户的主目录一起删除。

```
userdel -r tx
```

(3) 更改用户信息

修改用户账号就是根据实际情况更改用户的有关属性，如用户号、主目录、用户组、登录Shell 等。

修改已有用户的信息使用 `usermod` 命令，其格式如下：

代码：

`usermod` 选项 用户名

常用的选项包括 `-c`, `-d`, `-m`, `-g`, `-G`, `-s`, `-u` 以及 `-o` 等，这些选项的意义与 `useradd` 命令中的

选项一样，可以为用户指定新的资源值。另外，有些系统可以使用如下选项：

语法格式：

`-l` 新用户名

这个选项指定一个新的账号，即将原来的用户名改为新的用户名。

例子：

```
usermod -s /usr/sbin/nologin yanwj
```

(4) 检查用户身份

`who` 查询当前在线用户

`w` 查询当前在线用户的详细信息

`groups` 查询用户所属的组

`id` 显示用户 ID 信息

2、用户的配置文件

`/etc/passwd`: #用户的配置文件，保存用户账户的基本信息

`/etc/shadow` #用户影子口令文件

`/etc/group` #用户组配置文件

/etc/gshadow #用户组的影子文件

/etc/default/useradd # 使用 useradd 添加用户时需要调用的一个默认的配置文

/etc/login.defs #定义创建用户时需要的一些用户的配置文件

/etc/skel/ #存放新用户配置文件的目录

(5)/etc/passwd 详解

/etc/passwd 文件中每行定义一个用户账号，有多少行就表示多少个账号，在一行中可以清晰的看出，各内容之间又通过“:”号划分了 7 个字段，这 7 个字段分别定义了账号的不同属性，passwd 文件实际内容如下：

```
# head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

- 字段 1：帐号名，这是用户登陆时使用的账户名称，在系统中是唯一的，不能重名
- 字段 2：密码占位符 x；早期的 unix 系统中，该字段是存放账户和密码的，由于安全原因，后来把这个密码字段内容移到/etc/shadow 中了。可以看到一个字母 x，表示该用户的密码 etc/shadow 文件中保护的。
- 字段 3：UID；范围是 0-65535
- 字段 4：GID；范围是 0-65535；当添加用户时，默认情况下会同时建立一个与用户同名且 UID 和 GID 相同的组。
- 字段 5：用户说明；这个字段是对这个账户的说明

- 字段 6: 宿主目录; 用户登陆后首先进入的目录, 一般与"/home/用户名"这样的目录
- 字段 7: 登录 Shell 当前用户登陆后所使用的 shell, 在 centos/rhel 系统中, 默认的 shell 是 bash; 如果不希望用户登陆系统, 可以通过 usermod 或者手动修改 passwd 设置, 将该字段设置为/sbin/nologin 即可。大多数内置系统账户都是/sbin/nologin, 这表示禁止登陆系统。这是出于安全考虑的。

(6)linux 系统用户角色划分

用户在系统中是分角色的, 在 Linux 系统中, 由于角色的不同, 权限和所完成的任务也不同; 值得注意的是用户的角色是通过 UID 和 GID 识别的; 特别是 UID, 在运维工作中, 一个 UID 是唯一标识一个系统用户的账号。

用户账户:

超级用户 root (0)

程序用户 (1~1000)

普通用户 (1001~65535)

(7)用户组

每个用户都至少属于一个用户组

每个用户组可以包括多个用户

同一用户组的用户享有该组共有的权限

(8)用户的影子口令文件/etc/shadow

由于 passwd 文件必须要被所有的用户读, 所以会带来安全隐患。而 shadow 文件就是为了解决这个安全隐患而增加的。

来看一下/etc/shadow 的权限：

```
# ls -l /etc/shadow-----. 1 root root 1059 4 月 11 14:13 /etc/shadow
```

其文件内容为：

```
# head -5 /etc/shadow
```

```
root:$6$brYJfSgOsd8kAu5m$5iAUHQZJKWLP1Bx7VNEV6layO4UJyOekk  
BJWr9tmqjHBbtPCttYC2d8OK.Jt7eH2/oxNq82Bc5v8iHNYfnX2d1:18407:0  
:99999:7:::
```

```
daemon*:18348:0:99999:7:::
```

```
bin*:18348:0:99999:7:::
```

```
sys*:18348:0:99999:7:::
```

```
sync*:18348:0:99999:7:::
```

和/etc/passwd 一样，shadow 文件的每一行内容，也是以冒号(:)作为分隔符，共 9 个字段，其各个字段的意义如下表：

- 字段 1：帐号名称
- 字段 2：加密的密码
- 字段 3：最近更改密码的时间；从 1970/1/1 到上次修改密码的天数
- 字段 4：禁止修改密码的天数；从 1970/1/1 开始，多少天之内不能修改密码，默认值为 0
- 字段 5：用户必须更改口令的天数；密码的最长有效天数，默认值为 99999
- 字段 6：警告更改密码的期限；密码过期之前警告天数，默认值为 7；在用户密码过期前多少天提醒用户更改密码
- 字段 7：不活动时间；密码过期之后账户宽限时间 3+5；在用户密码过期之后到禁用账户的天数
- 字段 8：帐号失效时间，默认值为空；从 1970/1/1 日起，到用户被禁用的天数

- 字段 9: 保留字段 (未使用) , 标志

密码过期: 一旦超过密码过期日期, 用户成功登陆, Linux 会强迫用户设置一个新密码, 设置完成后才开启 Shell 程序

账户过期: 若超过账户过期日期, Linux 会禁止用户登陆系统, 即使输入正确密码, 也无法登陆

(9) 用户组配置文件/etc/group

/etc/group 文件是用户组的配置文件, 内容包括用户与用户组, 并且能显示用户归属哪个用户组, 因为一个用户可以归属一个或多个不同的用户组; 同一用户组的用户之间具有相似的特性。如果某个用户下有对系统管理有最重要的内容, 最好让用户拥有独立的用户组, 或者是把用户下的文件的权限设置为完全私有; 另外 root 用户组一般不要轻易把普通用户加入进入。

/etc/group 文件就是记录 GID 与用户组的文件。/etc/group 文件同 /etc/passwd 类似, 其文件权限也是 644:

```
# ls -l /etc/group
```

```
-rw-r--r-- 1 root root 1304 8月 3 20:07 /etc/group
```

/etc/group 的文件内容为:

```
# head -5 /etc/group
```

```
root:x:0:
```

```
daemon:x:1:
```

```
bin:x:2:
```

```
sys:x:3:
```

adm:x:4:

group 文件各个字段的详细说明：

字段 1：组账户名称

字段 2：密码占位符 x；通常不需要设置该密码，由于安全原因，该密码被记录在/etc/gshadow 中，因此显示为'x'。这类似/etc/shadow

字段 3：组账户 GID 号,用户组 ID

字段 4：本组的成员用户列表；加入这个组的所有用户账号

4.2 密码管理

用法：passwd [选项] [登录]

选项：

-a, --all	报告所有帐户的密码状态
-d, --delete	删除指定帐户的密码
-e, --expire	强制使指定帐户的密码过期
-h, --help	显示此帮助信息并推出
-k, --keep-tokens	仅在过期后修改密码
-i, --inactive INACTIVE	密码过期后设置密码不活动为 INACTIVE
-l, --lock	锁定指定的帐户
-n, --mindays MIN_DAYS	设置到下次修改密码所需等待的最短天数 为 MIN_DAYS
-q, --quiet	安静模式
-r, --repository REPOSITORY	在 REPOSITORY 库中改变密码
-R, --root CHROOT_DIR	chroot 到的目录

- | | |
|--------------------------|----------------------------------|
| -S, --status | 报告指定帐户密码的状态 |
| -u, --unlock | 解锁被指定帐户 |
| -w, --warndays WARN_DAYS | 设置过期警告天数为 WARN_DAYS |
| -x, --maxdays MAX_DAYS | 设置到下次修改密码所需等待的最多天数
为 MAX_DAYS |

例 1:

```
#passwd uos
```

新的 密码: uos@uos

重新输入新的 密码: uos@uos

4.3 组管理

1、添加用户组

语法格式:

```
groupadd 选项 用户组
```

可以使用的选项有:

代码:

-g GID 指定新用户组的组标识号 (GID) 。

-o 一般与-g 选项同时使用, 表示新用户组的 GID 可以与系统已有用户组的 GID 相同。

```
groupadd 添加一个组
```

例 1:

```
# groupadd group1
```

此命令向系统中增加了一个新组 group1，新组的组标识号是在当前已有的最大组标识号的基础上加 1。

```
#groupadd -g 101 group2
```

此命令向系统中增加了一个新组 group2，同时指定新组的组标识号是 101。

2、gpasswd 管理组

gpasswd 命令 管理组

语法：

格式：gpasswd [选项] 组名

选项：

- A 定义组管理员列表
- a 添加组成员，每次只能加一个
- d 删除组成员，每次只能删一个
- M 定义组成员列表，可设置多个，用 “,” 分开——定义的组成员必须是已存在

用户的

- r 移除密码

例子

gpasswd 将用户添加组命令

```
gpasswd -a uos tx
```

3、groupdel 删除一个已有的用户组

语法格式:

```
groupdel 用户组
```

```
#groupdel group1
```

此命令从系统中删除组 group1。

4、改用户组的属性使用 groupmod 命令

语法格式:

```
groupmod 选项 用户组
```

常用的选项有：

-g GID 为用户组指定新的组标识号。

-o 与-g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。

-n 新用户组 将用户组的名字改为新名字

例 1：

```
# groupmod -g 102 group2
```

此命令将组 group2 的组标识号修改为 102。

例 2:

```
# groupmod -g 10000 -n group3 group2
```

此命令将组 group2 的标识号改为 10000，组名修改为 group3。

5、chage 修改用户密码有效期

修改用户密码有效期：chage

语法：

格式:chage [选项] 用户名

常用命令选项：

复制代码

-d:将最近一次密码设置时间设置为"最近时间"

-E:指定账号过期时间，YYYY-MM-DD

-l:指定当密码失效后多少天锁定账号；

-l:列出密码有效期信息

-m:指定密码的最小天数

-M:指定密码的最大天数

-W:将过期警告天数设置为"警告天数"

例子:

chage -l uos3 #查看用户信息

chage -d 0 uos3 #切换控制台用 uos3 登录进行测试

chage -E 2015-10-1 uos3

第五章 文件属性与权限

概述：

- ◆ 在 Linux 系统中，一切皆文件（目录也是一个特殊的文件）；
- ◆ 因此对系统的管理，说到底就是对文件进行管理。

5.1 文件类型的查看

ls -l 查看文件的详细信息

```
-rw-r--r-- 1 root lonny 12 Apr 19 19:50 file1
```

文件类型、文件权限、硬连接数、文件属主、文件属组、文件大小、更动时间、文件名。

文件类型：第一位表示文件类型

-：表示一般文件

d：表示目录

l：表示软链接

p：进程间相互通信的文件，Linux 拥有一些机制来允许进程间的互相通信，这些机制称为进程间通信机制，管道 (pipe)、命名管道 (FIFO)、共享缓冲区、信号量、sockets 信号等都是进程间常用通信机制，pipe 是用于父进程和子进程之间通信。FIFO 是一个文件，允许运行在同一台计算机的进程间进行通信。

s: 表示 socket 通信套接字文件 (通常用于网络数据连接)

c: 表示字符设备文件 (如: 键盘、鼠标、终端等, 通常放在/dev 下)

b: 表示块设备文件 (存储数据设备文件, 如: 硬盘)

file /etc/passwd 查看文件类型, text

file /dev/sda 块设备

file /boot/vmlinuz-4.19.0-6-amd64 内核启动镜像

file /usr/bin/touch 可执行文件

stat test 查看文件详细信息, 更改时间等

隐藏文件与转义符

touch /home/uos/Desktop/.uos 创建隐藏文件, 以 "." 开头的文件为隐藏文件

ls -la /home/uos/Desktop/.uos -a 显示所有子目录和文件信息, 包括隐藏文件

touch /home/uos/Desktop/fi\\e \ 转义符

touch /home/uos/Desktop/'fi\e' 双, 单引号, 表示之内的字符串, 没有其他含义

文件名中不能包括 Linux 特殊字符如 "\"、"/" 等(如果在文件中使用这些特殊符号可通过转义符 "\" 将其转义)。

5.2 文件权限

文件权限

读权限 r: 允许查看文件内容, 使用 4 表示

写权限 w: 允许修改文件内容, 使用 2 表示

可执行 x: 允许运行程序, 使用 1 表示

无权限 -: 使用 0 表示

例如:

drwxr-xr-x 意思是一个权限为 755 的目录

-rw-r--r-- 意思是一个权限为 644 的文件

所有权

属主: 拥有该文件的用户账号

属组: 拥有该文件的组账号

更改文件权限

chmod 设置文件或目录的权限

格式:

chmod [-R] [ugoa] [+=] [rwx] 参数

chmod [-R] [nnn] 参数

常用选项：

-R：表示以递归的方式设置目录及目录下的所有子目录及文件的权限。

u 属主

g 属组

o 其他人

a 所有人

+ 添加

- 删除

= 重置

nnn 数字权限，如：777 666 644 755 等

示例：

touch 1234

ls -l 1234

chmod u+x 1234 属主加可执行权限

chmod g=rwx 1234 重置属组权限为可读可写可执行

chmod o+rwx 1234 给除了属主属组的其他人增加可读可写可执行

chmod 755 1234 给文件改为 755 权限

更改文件属主

chown 设置文件的归属(只有 root 用户才可以用“chown”指令来改变文件的拥有者)

chown 属主:属组 文件名

chown 属主.属组 文件名

示例：

groupadd pxb 创建组 pxb

useradd -g pxb uos1 创建用户 uos1 并指定他的组

useradd -g pxb uos2

echo 12345678 > /test 创建一个文件并写入

chown uos1:pxb /test 更改属主属组

chmod 640 /test 权限修改为 640 “rw-r-----” , uos1 权限可读可写, 组权限可读,

其他人没有权限

su - uos1 切换到 uos1 用户

cat /test 可查看

echo 123 >> /test 可写

su - uos2 切换到 uos2 用户

cat /test 能读

echo 123 >> /test 不能写

useradd uos3 创建 uos3

su - uos3 切换用户

cat /test 不能读

echo 456 >> /test 不能写

5.3 文件/文件夹的隐藏权限

管理 Linux 系统中的文件和目录，除了可以设置上述读写执行权限外，还可以使用 **chattr**

设置文件/文件夹的隐藏权限，chattr 只有 root 用户可以使用

命令格式为：

chattr [+ -=] [属性] 文件/目录名

+ ： 在原有参数设定基础上，追加参数。

- ： 在原有参数设定基础上，移除参数。

= ： 更新为指定参数设定。

属性：

i:如果对文件设置 i 属性，那么不允许对文件进行删除、改名，也不能添加和修改数据;如果对目录设置 i 属性，那么只能修改目录下文件中的数据，但不允许建立和删除文件

a:如果对文件设置 a 属性，那么只能在文件中增加数据，但是不能删除和修改数据;

如果对目录设置 a 属性，那么只允许在目录中建立和修改文件，但是不允许删除文件

u:设置此属性的文件或目录，在删除时，其内容会被保存，以保证后期能够恢复，
常用来防止意外删除文件或目录

s和 u 相反, 删除文件或目录时, 会被彻底删除 (直接从硬盘上删除, 然后用 0 填充所占用的区域), 不可恢复

lsattr 文件/目录 查看文件或目录的隐藏属性

示例

```
touch uosfile
```

```
lsattr uosfile
```

```
chattr +i uosfile    不能修改, 不能删除
```

```
lsattr uosfile
```

```
chattr -i +a uosfile
```

```
lsattr uosfile
```

```
echo 123 > uosfile
```

```
echo 123 >> uosfile    只能追加, 不能删除
```

```
chattr -a uosfile
```

```
rm uosfile
```

5.4 文件访问控制列表与链接

文件访问控制列表

ACL 可以为某个文件单独设置该文件具体的某用户或组的权限。

命令格式为：

`getfacl <文件名>` 获取文件的访问控制信息

`setfacl` 设置文件的 acl

`-m` 修改文件的 acl

`-x` 取消用户或组对文件的权限

示例：

`setfacl -m u:用户名:权限 <文件名>` //设置某用户名的访问权限

`setfacl -m g:组名:权限 <文件名>` //设置某个组的访问权限

`setfacl -x u:用户名 <文件名>` //取消某用户名的访问权限

`setfacl -x g:组名 <文件名>` //取消某个组的访问权限

实验：

`touch /var/log/uos.log`

`getfacl /var/log/uos.log`

`su - uos`

`vim /var/log/uos.log` //不能更改

`setfacl -m u:uos:rwx /var/log/uos.log`

`su - uos`

`vim /var/log/uos.log` //可以进行更改

`setfacl -x u:uos /var/log/uos.log`

软链接(符号链接)与硬链接

为源文件创建了一个新的指针。当对软链接操作时，系统就会找到原文件并对原文件进行操作（与 Windows 的快捷方式类似）。

语法：

`ln -s <源文件> <新建链接名>`

硬链接

硬链接文件完全等同于原文件，原文件名和连接文件都指向相同的物理地址

语法：

`ln <源文件> <新建链接名>`

硬链接的描述

不可跨文件系统创建硬连接，也不可为目录建立硬链接。

文件在磁盘中的数据是唯一的，这样就可以节省硬盘空间

由于只有当删除文件的最后一个节点时，文件才能真正从磁盘空间中消除，因此可以防

止不必要的误删除。

硬连接占用 inode

实例：

```
touch uosfile
```

```
cp uosfile uosfile1
```

```
ln -s uosfile uosfile2    #软连接
```

```
ln uosfile uosfile3       #硬连接
```

```
echo uosystem > uosfile
```

第六章 目录管理

6.1 目录结构

Linux 系统的目录组成类似一个倒置的树型结构, 它以一个名为根（“/”）的目录开始向下延伸。

它不同与其它操作系统。例如 windows, 在 windows 中, 它有多少分区那么就有多少个根, 而这些根之间是并列的, 而在 Linux 中无论有多少个分区都有一个根。

```
root@yanght-PC: /# tree -L 1 -d
.
├── bin -> usr/bin
├── boot
├── data
├── dev
├── etc
├── gitlabdata
├── home
├── lib -> usr/lib
├── lib32 -> usr/lib32
├── lib64 -> usr/lib64
├── libx32 -> usr/libx32
├── lost+found
├── media
├── mnt
├── nonexistent
├── opt
├── proc
├── recovery
├── recovery_live
├── root
├── run
├── sbin -> usr/sbin
├── srv
├── sys
├── tmp
├── usr
├── var
└── 27 directories
```

在 Linux 系统中对应的文件类型字符为 “d” 。

目录是一种特殊的文件

目录上的读写执行权限与普通文件有所不同。

读：用户可以读取目录内的文件

写：单独使用没有作用。它与读和执行权限连用可以在目录内添加与删除任何文件。

执行：用户可以进入目录，调用目录内的资料

常见目录：

/bin 存储常用用户指令。

/boot 存放用于系统引导时使用的各种文件

/dev 存放设备文件

/etc 存放系统、服务的配置目录与文件

/home 存放用户家目录

/lib 存放库文件，如内核模块、共享库等

/usr 存放系统应用程序目录 i

6.2 强制位 u+s

setuid

设置使文件在执行阶段具有文件所有者的权限(例如：/usr/bin/passwd);如果一般用户执行该文件，则在执行过程中，该文件可以获得 root 权

语法：

```
chmod u±s <文件名>
```

u+s 实验:

```
su - uos1
```

```
touch /root/file      #权限禁止
```

```
chmod u+s /usr/bin/touch  #给 touch 设置强制位 setuid 之后就可以以 root 身
```

份去执行

```
su - uos
```

```
touch /root/file      #权限允许
```

切回 root 查看

6.3 强制位 g+s

setgid

该权限只对目录有效, 目录被设置该位后, 任何用户在此目录下创建的文件都具有和该目录所属的组相同的组

语法:

```
chmod g±s <目录名>
```

g+s 实验:

```
mkdir uos1 //创建目录/uos 并设置所属组为 uos
```

```
chown :uos /uos1/
```

```
touch /uos1/1.txt    //新创建文件、可以看到其属组还是属于创建者的
```

```
chmod g+s /root/uos1/    //为属组添加强制位 setgid
```

```
touch /uos/2.txt    //可以看到新建的文件会自动继承此前目录的属组
```

6.4 冒险位（或粘制位） o+t

Sticky 冒险位或称粘制位

- ◆ 普通文件的 sticky 位会被 linux 内核忽略。
- ◆ 目录的 sticky 位表示这个目录里的文件只能被 owner 和 root 删除。

o+t 实验：

```
useradd -m -s /bin/bash uos1
```

```
mkdir /tx
```

```
chmod o=rwx /tx/
```

```
touch /tx/uos1
```

```
su - uos1
```

```
rm /tx/uos1    //是可以删除的
```

```
exit    注销
```

```
chmod o+t /tx/
```

```
touch /tx/uos2
```

```
su - uos1
```

```
rm /tx/uos2
```

rm: 无法删除'/tx/uos1': 不允许的操作 //加上 t 权限之后就不可以删除其他人的文件

6.5 umask: 控制新建的文件或目录的权限

umask 值与新建文件、目录权限对照表

umask 值	目录权限值	文件权限值
0	7	6
1	6	6
2	5	4
3	4	4
4	3	2
5	2	2
6	1	0
7	0	0

例如: umask 值为 0000, 则目录权限值 777, 文件权限值 666

umask 值为 0022, 则目录权限值 755, 文件权限值 644

实验:

umask

0022 //当前 umask 掩码是 0022, 新创建文件的权限是 644, 目录的权限是 755

touch file1

mkdir di1

ls -l

umask 0027 //将 umask 掩码修改为 0027 之后, 新建文件的权限是 640, 目录的权限是 750

touch file2

mkdir di2

ls -l

第七章 用户信息交互与 VI 编辑器

7.1 常用的交互工具

实验要求:

除管理员 root 用户外, 新建普通用户 uos1

使用 tty1 登录 root 用户, tty2 登录 uos1

write 命令 指定一个在线用户发送短消息

//使用 root 用户对 uos1 用户发送消息

```
root@uos1-PC:~# write uos1
```

write: write: you have write permission turned off.

aaaaaaaaaaaa

aaaaaaaaaaaa

ctrl+d 结束

//去 uos1 用户验证查看

wall 命令 向所有在线用户广播

//使用 root 用户对所有用户发送消息

```
root@uos1-PC:~# wall 123456789
```

//客户端验证接收

7.2 vim 编辑器:

vi 编辑器是所有 Unix 和 Linux 系统下标准的编辑器, 相当于 Windows 的记事本一样, 它工作在字符模式, 由于不需要图形界面, 所以效率高; vim 是 vi 编辑器的升级版本, 除了兼容 vi 的所有指令以外, 还添加了许多重要的特性, 例如支持正则搜索, 语法高亮, 对 C 语言的自动缩进等;

vim 的三种模式:

命令模式(其它模式->ESC):

控制光标移动, 删除字符, 段落复制

编辑/插入模式(命令模式->a、i、o、A、I、O)

新增文字及修改文字

末行模式(命令模式->:)

保存文件, 离开 vi, 以及其他设置

vim 基础操作

vim --help 查看帮助

命令模式:

yy 复制当前行正行

nyy 复制从光标所在行开始的 n 行

dd 剪切当前光标所在行

ndd 剪切从光标所在行开始的 n 行

p 粘贴光标位置之后

G 跳转至尾行

gg 跳转至首行

dw 删至词尾

ndw 删除后 n 个词

d\$ 删至行尾

nd\$ 删除后 n 行（从光标当前处开始算起）

u 撤销上一次修改

U 撤销一行内的所有修改

末行模式

:r /etc/passwd 读文件内容进 vim

:r! ls -l / 读命令结果保存到文件中

:set number 行号

:set nonumber 去除行号

:s/old/new/g 在当前行中查找到的所有字符串 old 替换为 new

:2,6s/old/new/g 2-6 行替换

:%s/old/new/g 在整个文件范围内替换

:X 加入密码

:q 不保存退出

:q! 强制退出不保存

:wq 保存退出, 同 x

:wq! 强制保存退出

进入编辑模式

a 光标后插入

i 当前光标前插入

o 在当前光标下插入空行

A 在光标所在行尾插入

I 在光标行首插入内容

O 在当前光标上插入空行

7.3 修改 vim 配置

```
echo ":set number" >> /etc/vim/vimrc
```

```
vim /etc/passwd
```

实验-非正常关闭 vim 的恢复

```
vim uos 直接关闭终端
```

```
ls -la
```

```
vim uos 会有提示信息
```

删除同目录下面同名的隐藏文件即可

其它配置：

```
:set mouse-=a
```

```
:set tabstop=4
```

```
:set number
```

```
:set backspace=indent,eol,start
```

```
:set nocompatible
```

第八章 文件查找与压缩归档

8.1 文件查找

8.1.1 可执行文件的搜索

`which <指令>` 显示一个指令的完整路径与别名。

`whereis <指令>` 搜索一个指令的完整路径以及其帮助文件

8.1.2 Locate 更快的查找

`locate` 用于查找文件或目录，比 `find` 命令快，是因为不回去搜索目录，而是搜索一个数据库 `/var/lib/mlocate/mlocate.db`，这个库有本地所有文件信息，Linux 会自动创建这个数据库，每天自动更新一次，因此，我们在用 `whereis` 和 `locate` 查找文件时，有时会找到已经被删除的数据，或者刚刚建立文件，却无法查找到，原因就是数据库文件没有被更新。

为了避免这种情况，可以在使用 `locate` 之前，先使用 `updatedb` 命令，手动更新数据库

```
apt-get install locate -y
```

```
updatedb
```

```
locate passwd
```

8.1.3 find 在指定目录下查找文件

特点:

从指定路径下递归向下搜索文件。

支持按照各种条件方式搜索。

支持对搜索得到的文件再进一步的使用指令操作(例如: 删除、统计大小、复制等)

常用选项:

-name 根据文件名寻找文件

-user 根据文件拥有者寻找文件

-group 根据文件所属组寻找文件

-perm 根据文件权限寻找文件

-size 根据文件大小寻找文件[±Sizek]

-type 根据文件类型寻找文件, 常见类型有:

f(普通文件)、c(字符设备文件)、b(块设备文件)、l(连接文件)、d (目录)

-o 表达式或

-a 表达式与

示例:

```
find / -name uos1
```

```
find / -user hehe
```

```
find / -group xiaolizi
```

```
find / -perm 644
```

```
find / -size +10k
```



```
find /etc -size -10k
```

```
find /etc -type f/c/b/l/d
```

对文件进一步操作

```
find [路径] [参数] [表达式] -exec 指令 {} \;
```

{}代表 find 找到的文件

; 命令结束标志，由于各个系统中的;会有不同的意义，所以前面加\转义

示例：

```
find /tmp/ -type f -exec rm -rf {} \;
```

 用 exec 选项执行 cp 命令

8.2 文件内容的查找

grep 查找文件里符合条件的字符串。

常用选项：

- c 计算匹配关键字的行数
- i 忽略字符大小写的差别
- n 显示匹配的行及其行号
- s 不显示不存在或不匹配文本的错误信息
- h 查询多个文件时不显示文件名
- l 查询文件时只显示匹配字符所在的文件名
- v 取反

--color=auto

示例：

```
grep root /etc/passwd
```

```
grep ^root /etc/passwd
```

```
grep bash$ /etc/passwd
```

```
grep -i ROOT /etc/passwd
```

8.3 归档与压缩

tar 文件归档压缩

功能说明：用于建立、还原备份文件的工具，它也可以加入或解开备份文件内的文件。有时也称之为“打包”。

常用选项：

c 建立新的备份文件

v 显示指令执行过程

x 解包

t 显示 tar 文件的内容

z 用 gzip 压缩或解压文件，以.tar.gz 或.tgz 为后缀

j 用 bzip 压缩或解压文件，以.tar.bz2 为后缀

f 指定要操作的文件

-C 指定解压目录

示例:

```
tar czvf `date +%F`.log.tar.gz /var/log/      #以时间为名称打包
```

```
tar xzvf 2016-03-09.log.tar.gz -C /tmp/      #-C 指定解压目录
```

gzip 压缩

gunzip 或 gzip -d 解压

只能压缩文件，不能压缩目录；后缀名为.gz，而且不保留原文件

bzip2 压缩

bunzip2 或 bzip2 -d 解压

压缩文件，-k 参数可以保留原文件；后缀为.bz2

第九章 输入输出重定向

9.1 重定向

Linux 标准输入设备是键盘，标准输出设备是显示器，标准错误输出指的是显示器

设备	设备名	文件描述符	类型
键盘	/dev/stdin	0	标准输入
显示器	/dev/stdout	1	标准输出
显示器	/dev/stderr	2	标准错误输出

输出重定向：把要输出到显示器的内容，输出到文件中；>表示覆盖写入 >>追加写入

示例：

```
echo uos > test
```

```
cat test
```

```
echo bing > test
```

```
cat test
```

```
echo txuos >> test
```

```
cat test
```

错误重定向：2>错误重定向

示例：

```
wadwadwad 2> test
```

```
cat test
```

双重输出重定向

```
find / -user uos > test
```

```
cat test
```

```
find / -name passwd >file 2> test#将正确的输出结果与错误输出结果一次性单独
```

地送到不同的地方

```
cat test
```

```
cat file
```

如果用户将不管是正确输出还是错误输出结果都送到同一个指定的地方则可使用 “&>

或 &>>” 来完成。

```
find / -name passwd &> test
```

```
cat test
```

```
find / -name passwd > test 2>&1 #脚本里使用，将错误结果也输入进文件里
```

```
cat test
```

输入重定向：不使用标准输入端口输入文件，而是使用指定的文件作为标准输入设备；使用

“<” 来重定向输入源；使用 << 让系统将一次键盘的全部输入，先送入虚拟的 ‘当前文档’，然后一次性输入。

cat > ok << EOF#交互式

123

456

EOF #结束符

cat ok

9.2 管道

管道符号|：将前一个命令的执行结果作为后一个命令的执行参数

cat /etc/passwd | grep root

cat /etc/passwd | grep ^root

9.3 文件处理

文件操作命令

more 文件分页查看，空格向下一屏，Ctrl+B 返回上一屏

less 文件分页查看，使用[pageup] [pagedown]来往前往后翻看文件，回车一行，

空格翻页，q 退出

head /etc/passwd 默认前 10 行

head -5 /etc/passwd

tail /etc/passwd 默认后 10 行

`tail -n 5 /etc/passwd`

`tail -f /var/log/message` 实时监测文件

`wc -l /etc/passwd` 显示行数

`sort -rnk 3 -t : /etc/passwd`

`-r` 以相反的顺序来排序

`-n` 依照数值的大小排序

`k` 是指按照那一列进行排序

`-t <分隔字符>`指定排序时所用的栏位分隔字符。

`uniq -c uosfile`

`-c` 在每行旁边显示该行重复出现的次数

`sort uosfile | uniq -c`

`-r` 以相反排序

`df -Th | grep sda2 | tr -s " " | cut -d " " -f 6 | cut -d "%" -f 1` `cut -d` 自定义分

隔符，`-f` 分隔符后第几行，`tr -s` 把连续重复的字符以单独一个字符表示

`paste file1 file2 file3` 把每个文件以列对列的方式，一列列地加以合并

第十章 软件包的安装与使用

10.1 使用 dpkg 管理软件包

`dpkg`: Debian Linux 系统用来安装、创建和管理软件包的实用工具

`dpkg -l` #显示安装软件列表

`dpkg -l vsftpd` #查看是否安装 vsftpd

`cd /home/uos/Desktop`

`wget`

http://ftp.de.debian.org/debian/pool/main/v/vsftpd/vsftpd_3.0.3-12_amd64.deb

桌面版可直接双击安装

`dpkg -l /home/qin/Desktop/vsftpd_3.0.3-12_amd64.deb` --info 查看软件包

信息

`dpkg -c /home/qin/Desktop/vsftpd_3.0.3-12_amd64.deb` --contents 列出

deb 包的内容

`dpkg -i /home/qin/Desktop/vsftpd_3.0.3-12_amd64.deb` 安装软件包

`dpkg -L vsftpd` 查询 vsftpd 安装的文件

`dpkg -s vsftpd` 查询 vsftpd 的详细信息

`dpkg -r vsftpd` --remove 删除软件包，不清理配置文件

`dpkg -P vsftpd` --purge 删除软件包，清理配置文件

`wget`

http://mirrors.163.com/deepin/pool/main/a/apache2/apache2_2.4.33-4deepin_amd64.deb

`dpkg -i apache2_2.4.33-4deepin_amd64.deb` 报错，存在依赖关系

10.2 使用 apt 管理软件包

apt (Advanced Packaging Tool) 包管理工具

update - 更新可用软件包列表

upgrade - 通过 安装/升级 软件来更新系统

apt install apache2 报错，因为之前安装 apache2 和 apache2-bin 的残留

dpkg -P apache2

apt install -y apache2

apt list 列出所有可安装软件

apt search uos 查询软件

apt show apache2 查询 apache2 的详细信息

apt remove apache2 删除软件包，不清理配置文件

apt purge apache2 删除软件包，清理配置文件

apt autoremove 自动删除不需要的包，例如 apache2 删除后的依赖包

apt install -f 不解决依赖关系，强制安装

10.3 获取 deb 包 （用外网提取 deb 包，内网安装）

apt purge -y apache2 #会同时清除软件包和软件的配置文件

apt autoremove -y #自动删除不需要的包，例如 apache2 删除后的依赖包

```
apt clean          #删除包缓存中的所有包

ls -l /var/cache/apt/archives/ #包缓存已清空

apt install -d -y apache2 #把包下载到缓存中而不安装

mkdir /home/qin/Desktop/apache2

cp /var/cache/apt/archives/*.deb /home/qin/Desktop/apache2

dpkg -i /home/qin/Desktop/apache2/*.deb  #安装软件包

systemctl start apache2

systemctl status apache2
```

10.4 源码包安装

```
wget http://nginx.org/download/nginx-1.14.2.tar.gz

tar xzvf nginx-1.14.2.tar.gz -C /tmp

./configure --help

./configure --prefix=/usr/local/nginx --without-http_rewrite_module

--without-http_gzip_module

make && make install

/usr/local/nginx/sbin/nginx

netstat -upant | grep nginx
```


第十一章 进程管理与系统监控

11.1 系统状态查询

查询系统信息：

uname 显示系统信息

uname -r 输出内核发行号

uname -a 详细显示操作系统信息

查看/修改主机名：

hostname 查看主机名

hostname uosystem 临时修改主机名

vim /etc/hostname 永久修改主机名

hostnamectl -set-hostname uosystem 修改主机名及相关配置

查看最近用户登录：

last 查看最近登录的用户

lastlog 查看所有用户的最近登录情况

11.2 进程管理

查看进程：

ps aux 查看进程信息

a 表示所有关联到终端的进程，如果同时使用 x 则代表所有进程；

u 表示列出进程的用户

ps aux 返回详解：

USER 该进程是哪个用户运行的

PID 进程 ID 号

%CPU 占用的 CPU 百分比

%MEM 占用的内存百分比

VSZ 占用的虚拟内存 (KB)

RSS 占用的物理内存大小 (KB)

TTY 在哪个终端运行的；

tty1-tty6 是本地字符界面终端；

pts/0~255 表示虚拟终端，一般是远程连接的终端；

? 表示不属于任何终端，是由系统启动的

STAT 进程状态

START 进程的启动时间

TIME 进程占用 CPU 的运算时间，注意不是系统时间

COMMAND 所执行的指令

Linux 进程的五种状态

R 运行状态：正在运行或在运行队列中等待

S 休眠状态，可中断：在等待某个条件的形成或接收到信号

D 不可中断状态：指的并不是 CPU 不响应外部硬件的中断，而是不响应异步信号；

通常是长时间 I/O 等待引起

Z 僵死状态：进程已终止，但进程描述符存在，直到父进程调用 wait4()系统调用后释放

T 停止状态：处于后台挂起或暂停状态，如：使用 ctrl+z 后台挂起就会产生该状态
进程

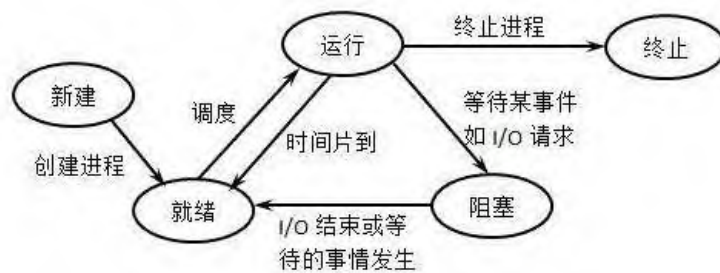


图 2. 进程的五态模型

状态后缀：

<：优先级高的进程

N：优先级低的进程

L：有些页被锁进内存

s：进程的领导者（在它之下有子进程）

l：多线程

+: 位于后台的进程组

free 查看系统内存使用状况

`free -m` 以 MB 为单位显示内存使用情况；`-g` 是以 GB 为单位；默认是按照 KB 为单位

`total` 物理内存总量

`used` 已用物理内存量

`free` 空闲物理内存量

`shared` 被共享使用的物理内存大小

`buff/cache` 表示磁盘缓存的大小

`available` 还可以被应用程序使用的物理内存大小， $\text{available} = \text{free} + \text{buffer} + \text{cache}$

top 系统监视器

- 第一行：系统当前时间 系统运行时间 当前用户登陆数 CPU 平均负载（这里有三个数值，分别是系统最近 1 分钟，5 分钟，15 分钟的平均负载）
- 第二行：total 进程总数、running 正在运行的进程数、sleeping 睡眠的进程数、stopped 停止的进程数、zombie 僵尸进程数。
- 第三行：CPU 使用率
 - %us 用户空间占用 CPU 百分比
 - %sy 系统内核占用 CPU 百分比
 - %ni 调整过用户态优先级的进程的 CPU 时间占比
 - %id 空闲的 CPU 百分比
 - %wa 等待 I/O 完成的 CPU 时间百分比

%hi 指的是 cpu 处理硬件中断的时间

%si 指的是 cpu 处理软中断的时间

%st 当 Linux 系统是在虚拟机中运行时，等待 CPU 资源的时间占比

通常 id%值可以反映一个系统 cpu 的闲忙程度

- 第四、五行：物理内存和 swap 使用情况
- 第六行：进程详细信息

PID 进程 PID 号

USER 运行进程的用户

PR 从系统内核角度看的进程调度优先级

NI 进程的 nice 值，即从用户空间角度看的进程优先级。值越低，优先级越高

VIRT 进程申请使用的虚拟内存量

RES 进程使用的物理内存量

SHR 进程使用的共享内存量

S 进程状态

%CPU CPU 占用百分比

%MEM 物理内存占用百分

TIME+ 进程创建后至今占用的 CPU 时间

COMMAND 运行进程使用的命令

top 常用快捷键

h 查看帮助信息

q 退出 top

- k 杀掉进程
- N 按照 PID 对进程排序。
- M 按%MEM 对进程排序。
- P 按%CPU 对进程排序。
- T 按 TIME+对进程排序

kill 终止进程

kill -l 查看所有信号

常用信号

- 1 (HUP) : 重新加载进程
- 9 (KILL) : 强制杀死进程
- 15 (TERM) : 正常停止一个进程

前台进程和后台后台进程

前台进程：一个命令执行后，独占 shell 终端，并拒绝其他输入

后台进程又称为守护进程，不受终端控制

示例：

```
ping www.baidu.com
```

ctrl+z 将当前正在运行的命令放入后台并挂起（暂停）

ctrl+c 终止前台进程

ping www.baidu.com & &表示将当前执行的命令放入后台并继续运行

jobs 查看后台进程

fg 1 fg [%作业号] 前台恢复运行

kill -9 %2 kill [%作业号] 给对应的进程发送终止信号

11.3 用 lsof 恢复文件

lsof: 查看进程打开的文件或打开文件的进程

示例:

lsof -p 1359 查看某个进程打开的所有文件

lsof -u root 查看某个用户打开的所有文件

lsof -u ^root 查看不是某个用户打开的文件，也就是取反，在用户名前加^

lsof /dev/null 查看某个文件被哪些进程打开使用

lsof -i:22 查看某个端口的使用情况

恢复内存中打开过的文件:

lsof |grep <filename> #查询打开过的文件描述符，取第 2 列

```
cp /proc/<fid>/fd/8 <filename>
```

第十二章 服务与计划任务

12.1 一号进程 systemd

systemd 是一个专用于 Linux 操作系统的系统与服务管理器；当作为启动进程(PID=1)运行时；它将作为初始化系统运行，也就是启动并维护各种用户空间的服务（pstree 查看）

systemctl 是 Systemd 的主命令，用于管理系统。

查看系统服务

systemctl list-units 查看所有系统服务

systemctl list-units --all | grep ssh 单独查看 apache2 服务

12.2 设置系统运行级别

systemctl get-default 查看默认运行级别

systemctl isolate runlevel3.target 启动运行等级 3

systemctl isolate runlevel5.target 图形界面

systemctl list-unit-files --type target | grep runleve 查看所有 type target

systemctl set-default graphical.target 设置默认启动到图形界面

systemctl set-default multi-user.target 设置默认启动到多用户字符界面

12.3 systemctl 控制服务

apt install apache2

systemctl status apache2

cat /lib/systemd/system/apache2.service apache2 服务启动文件

systemctl start apache2

ps aux | grep apache2 ps aux 查看进程

systemctl stop apache2

systemctl restart apache2

systemctl enable apache2 开机自启动

/etc/systemd/system/multi-user.target.wants/apache2.service enable 其实是是

建立 apache2 服务软链接

systemctl disable apache2 取消开机自启动

systemctl mask apache2 屏蔽服务，永远不能启动

systemctl unmask apache2 取消屏蔽

12.4 ssh 远程及 scp 远程拷贝

dpkg -l | grep ssh #默认安装 openssh-client

vim /etc/ssh/sshd_config

PermitRootLogin yes 开启 root 用户远程连接权限

```
systemctl restart sshd
```

```
ssh 192.168.200.202    测试, 不加用户名默认你当前什么身份连的以什么身份链接
```

```
ssh root@192.168.200.202
```

```
touch /tmp/uosfile
```

```
scp /tmp/uosfile 192.168.200.202:/root/Desktop/qintest
```

```
scp -r 192.168.200.202:/home /tmp/
```

12.5 ssh 免密码登录

server1:

```
ssh-keygen    一直回车
```

```
ssh-copy-id root@192.168.200.202
```

server2 验证

```
cat ~/.ssh/authorized_keys    server1 的公钥已添加
```

server1

```
ssh root@192.168.200.202    无密码可登录
```

12.6 cron 计划任务

格式:

```
* * * * * command
```

分(0-59) 时(0-23) 日 (一个月中的第几天 1-31) 月 (1-12) 星 (星期几 0-7,0/7

都表示周日) 需要执行的命令

符号含义

* 表示该范围内的任意时间

, 表示间隔的多个不连续时间点, 例如, "1,2,5,7,8,9"

- 表示一个连续的时间范围, 例如 "2-6" 表示 "2,3,4,5,6"

/ 指定间隔的时间频率, 例如 "0-23/2" 表示每两小时执行一次

示例:

```
0 17 * * 1-5 周一到周五每天 17:00
```

```
30 8 * * 1,3,5 每周一、三、五的 8 点 30 分
```

```
0 8-18/2 * * * 8 点到 18 点之间每隔 2 小时
```

```
0 * */3 * * 每隔 3 天
```

实验:

```
touch /tmp/uosfile
```

```
crontab -e -u root 第一次使用 crontab -e 时需选择编辑器, vim.basic 完整版的
```

vim; vim.tiny 是 tiny 版本的 vim,功能比较少

```
*/1 * * * * echo "UOS SYSTEM" >> /tmp/uosfile
```

crontab -l -u root 查看某用户的计划任务列表, 不指定用户则查看当前用户

第十三章 LINUX 的启动流程

13.1 Linux 的启动流程

- ① 开机自检：设备开机时，会有滴一声，自检开始，主要是检查计算机硬件，如：CPU，内存，主板，显卡等设备是否有故障
- ② 加载 BIOS：BIOS 自检，首先会在一个 Boot Sequence 程序中搜索可以让系统启动的引导设备（比如我们有时在 BIOS 中设置为从硬盘启动，或者从 CD-ROM 启动等等），如果 BIOS 找不到可以引导的设备及相关程序后，便会启动失败，如果找到相关硬盘设备，则 BIOS 将控制权交给启动设备中的 MBR 主引导记录
- ③ 读取 MBR 主引导记录：大小为 512 字节，存放预启动信息、分区表等信息
- ④ grub 引导菜单：系统读取内存中的 grub 配置信息，并依照此配置信息来启动不同的操作系统
- ⑤ 加载 kernel 内核：根据 grub 设定的内核映像所在路径，系统读取内存映像，并进行解压缩操作（内核文件都是以一种自解压的压缩格式存储以节省空间，存储于/boot 目录下），自解压完成，则加载 init(systemd)
- ⑥ init 初始化（systemd 所有进程的父进程）：初始化系统环境
- ⑦ 启动内核模块

13.2 破解 root 密码

开机按"e"键，进入 grub 引导菜单

找到 linux vmlinuz-* 开头的行在末尾添加 rw init=/bin/bash

按 ctrl+x 或 F10 继续启动

passwd root

reboot -f 或手动重启

13.3 防止破密码：增加 grub 口令

grub-mkpasswd-pbkdf2 # 输入密码，使用伪随机函数生成密钥

grub.pbkdf2.sha512.10000.....

vim /etc/grub.d/40_custom

set superusers="uos"

password_pbkdf2 uos

grub.pbkdf2.sha512.10000.6D67AA5F717AFE93D368A57804BC21B136DCA9D30C

BA6F74D9ABCBA32E76B59758E04A5D0A766734EDE586DEA2EC3E7F786F127FB55

C7922826455612B5406F5.30060B65CA1D5736364D109C7CB6459E5E2851EFF460

E43D72AAB8A592784B2D5353D9B51EE1EAFBC45AF739417CFC5EC12EE92D546E9

BC6A7BB58444BC27455

update-grub

重启验证

13.4 启动修复

```
rm -rf /boot/*
```

关机 BIOS 设置光盘启动，点击 install UOS 后，按 ctrl+alt+f2 进入命令行模式

```
sudo mount /dev/sda2 /mnt (随便挂)    挂载根分区
```

```
sudo mount /dev/sda1 /mnt/boot    挂载/boot 分区
```

```
sudo mount --bind /dev /mnt/dev    硬件设备，bind 绑定（目录挂载到目录）
```

```
sudo mount --bind /proc /mnt/proc
```

```
sudo mount --bind /sys /mnt/sys
```

```
sudo cp -rf /boot/* /mnt/boot/  将光盘引导/boot 复制到系统/boot
```

```
sudo chroot /mnt    把根目录换成指定的目的目录
```

```
grub-install /dev/sda    将 GRUB 安装到指定硬盘或分区上
```

```
update-grub    恢复 grub.conf
```

```
exit
```

```
reboot    选择硬盘启动
```


第十四章 网络管理

14.1 网络连接管理

查看 IP 的几种方法:

```
ip addr show
```

```
ifconfig
```

```
nmcli connection show 有线连接 查看网卡连接信息
```

device 管理

```
nmcli device show 查看网卡设备
```

添加一块网卡

```
nmcli device show | grep -i device | grep -v lo
```

```
nmcli device connect ens37 #链接网卡设备，物理连接
```

```
nmcli device disconnect ens37 #关闭网卡设备
```

修该网卡默认命名规则 (不想用 ens33 命名规则)

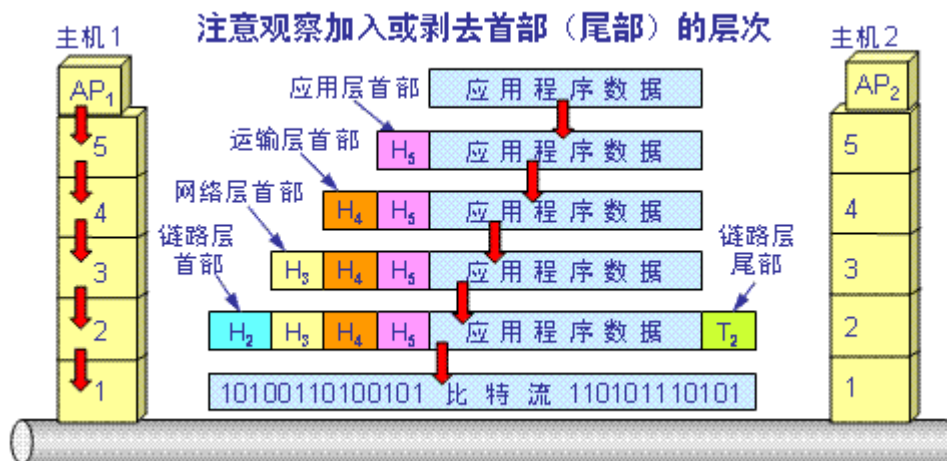
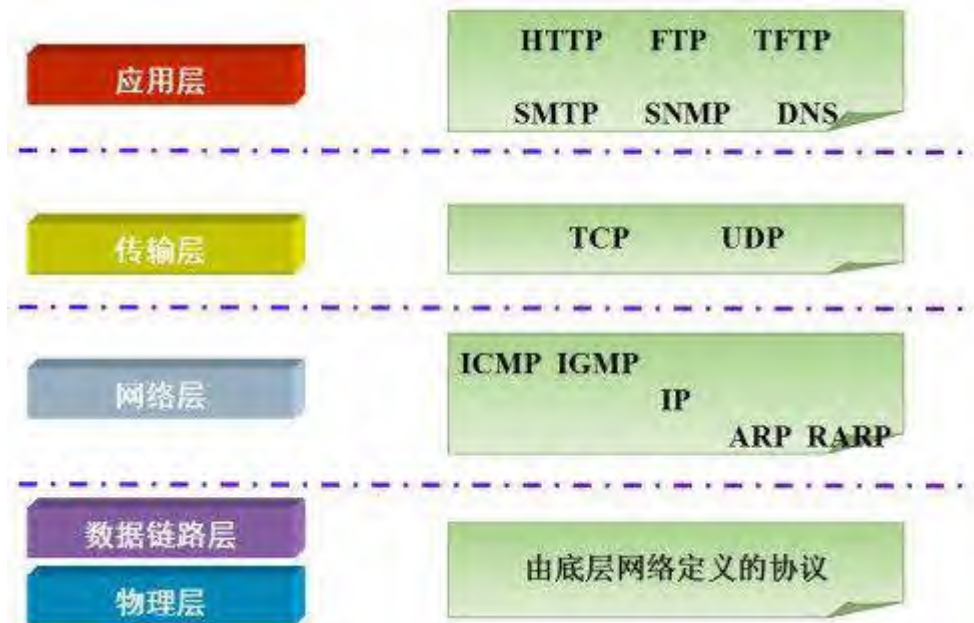
```
vim /etc/default/grub
```

```
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0" 更改为 eth0 的命
```

名规则

```
update-grub
```

shutdown -r now



14.2 配置 ip 地址

- uos 系统的网卡配置统一采用 nmcli 系列命令
- 配置文件位于 /etc/NetworkManager/system-connections/ 目录

nmcli connection show

nmcli connection delete ens33 #删除配置文件

```
nmcli connection add type ethernet con-name ens33 ifname ens33  
connection.autoconnect yes 重新生成配置文件并设置自动连接
```

#type 类型 ethernet 以太网卡, con-name 配置文件名字, ifname 设备名字,
connection.autoconnect 自动连接, 可加可不加, 一般都加上

```
nmcli connection modify ens33 ipv4.method manual ipv4.addresses  
192.168.200.201/24 ipv4.gateway 192.168.200.2 ipv4.dns 114.114.114.114  
connection.autoconnect yes
```

#modfiy 修改配置文件, manual 手动修改

```
nmcli connection up ens33 #启动配置文件
```

```
systemctl restart NetworkManager #重启服务查看
```

临时配置 IP 地址

```
ifconfig
```

```
ifconfig ens33 192.168.200.201/24
```

```
ifconfig
```

```
nmcli connection down ens33 #恢复
```

```
nmcli connection up ens33
```

一个网卡绑定多个 ip 地址

```
nmcli connection modify ens33 ipv4.method manual +ipv4.addresses  
"10.0.0.1/24  
  
nmcli connection down ens33  
  
nmcli connection up ens33  
  
ip addr show
```

14.3 ipv6 地址

```
nmcli connection delete ens33  
  
nmcli connection modify 有线连接 ipv6.method manual ipv6.addresses  
2001::1/64  
  
nmcli connection down 有线连接  
  
nmcli connection up 有线连接  
  
ping6 2001::1
```


第十五章 磁盘管理

15.1 基础理论

硬盘主分区表格式： mbr vs gpt

- Mbr: 在磁盘的第一扇区, 支持 4 个主分区, 旧标准
- Gpt: 支持 128 个主分区, 支持 2T 以上硬盘, 新标准

硬盘类型与速度： ssd vs sata/sas

- Ssd: 固状硬盘, 速度快, 非机械
- Sas/sata: sas 即串行 scsi, 速度 sata 快, 较 sata 贵, 多用于服务器中较重要应用; sata 即串行 ata 硬盘, 容量大, 便宜, 多用于备份;

Bios 启动类型： Legacy vs uefi

- Legacy: 传统 BIOS
- Uefi: 新型 EFI BIOS, 方便扩展 BIOS, 需要在启动硬盘建 EFI 分区, 以扩展 BIOS 的硬件驱动



常用分区类型与代码：

0 Empty	24 NEC DOS	81 Minix / old Lin	bf Solaris
1 FAT12	27 Hidden NTFS Win	82 Linux swap / So	c1 DRDOS/sec
(FAT-			
2 XENIX root	39 Plan 9	83 Linux	c4 DRDOS/sec
(FAT-			
3 XENIX usr	3c PartitionMagic	84 OS/2 hidden or	c6 DRDOS/sec
(FAT-			
4 FAT16 <32M	40 Venix 80286	85 Linux extended	c7 Syrinx
5 Extended	41 PPC PReP Boot	86 NTFS volume set	da Non-FS
data			
6 FAT16	42 SFS	87 NTFS volume set	db CP/M /

CTOS / .

7	HPFS/NTFS/exFAT 4d	QNX4.x	88	Linux plaintext de	Dell Utility
8	AIX	4e	QNX4.x 2nd part 8e	Linux LVM	df BootIt
9	AIX bootable	4f	QNX4.x 3rd part 93	Amoeba	e1 DOS access
a	OS/2 Boot Manag 50	OnTrack DM	94	Amoeba BBT	e3 DOS R/O
b	W95 FAT32	51	OnTrack DM6 Aux 9f	BSD/OS	e4

SpeedStor

c	W95 FAT32 (LBA) 52	CP/M	a0	IBM Thinkpad hi ea	Rufus
---	--------------------	------	----	--------------------	-------

alignment

e	W95 FAT16 (LBA) 53	OnTrack DM6 Aux a5	FreeBSD	eb	BeOS fs
f	W95 Ext'd (LBA) 54	OnTrackDM6	a6	OpenBSD	ee GPT
10	OPUS	55	EZ-Drive	a7	NeXTSTEP ef EFI

(FAT-12/16/

11	Hidden FAT12	56	Golden Bow	a8	Darwin UFS f0
----	--------------	----	------------	----	---------------

Linux/PA-RISC b

12	Compaq diagnost 5c	Priam Edisk	a9	NetBSD	f1 SpeedStor
14	Hidden FAT16 <3 61	SpeedStor	ab	Darwin boot	f4 SpeedStor
16	Hidden FAT16	63	GNU HURD or Sys af	HFS / HFS+	f2 DOS

secondary

17	Hidden HPFS/NTF 64	Novell Netware	b7	BSDI fs	fb VMware
----	--------------------	----------------	----	---------	-----------

VMFS

18	AST SmartSleep 65	Novell Netware	b8	BSDI swap	fc VMware
----	-------------------	----------------	----	-----------	-----------

VMKCORE

1b Hidden W95 FAT3 70 DiskSecure Mult bb Boot Wizard hid fd Linux raid
auto

1c Hidden W95 FAT3 75 PC/IX bc Acronis FAT32 L fe LANstep

1e Hidden W95 FAT1 80 Old Minix be Solaris boot ff BBT

分区: 把一块硬盘分成几个小块来使用

Fdisk、parted

格式化: 把一个分区格式化成某一种格式，以便被操作系统识别来读写数据

Mkfs

Mount: 格式化后的硬盘分区，挂载到某个目录来使用

RAID: 把一组硬盘组合到一起，像一块硬盘一样来使用

15.2 磁盘分区

分区:

添加一块 5G 的磁盘

fdisk -l

fdisk /dev/sdb

n p 1 回车 +1G

w

#最多创建 4 个主分区，要想更多创建扩展分区

mkfs.xfs /dev/sdb1 格式化为 xfs 格式

mkdir /sdb1

mount /dev/sdb1 /sdb1

df -h

mount 磁盘挂载

lsblk #树型结构显示分区信息

blkid /dev/sda3 #已格式化分区的 UUID 和文件系统

mkdir /cipan

mount /dev/sda3 /cipan/

df -h

15.3 swap 分区

fdisk /dev/sdc

mkswap /dev/sdc2 #建立交换分区

swapon -a #将/etc/fstab 文件中所设置为 swap 的设备开启

```
swapon /dev/sdc2 #开启交换分区
```

```
swapon -s #显示交换分区的信息
```

```
vim /etc/fstab
```

```
UUID="246cb701-407c-4410-8fc0-106f224ac884" none swap defaults 0 0
```

要自动开启交换区，可以把新的分区加入到/etc/fstab 中

15.4 lvm 逻辑卷

新建两个空白分区

```
pvs #查看
```

```
pvccreate /dev/sdb2 /dev/sdb3 #建立物理卷
```

```
vgs
```

```
vgcreate uosvg /dev/sdb2 /dev/sdb3 #建立卷组
```

```
vgs
```

```
lvcreate -L 1.9G uosvg -n uoslv #分割 LV 逻辑卷出来
```

```
lvs
```

```
mkfs.xfs /dev/uosvg/uoslv
```

```
mkdir /uos
```

```
mount /dev/uosvg/uoslv /uos/
```

`df -h`

15.5 磁盘配额

查看配额状态：

`mount`

安装配额管理包：

`apt install quota`

修改/etc/fstab:

`uuid= <uuid> /dev/qinlv /mnt/qinlv xfs`

`defaults,usrquota,grpquota 0 0`

激活配额检测：(xfs 不用)

ext4 执行: `quotacheck -cvug /mnt/qinlv`

对用户 qin 设置最高 20M，最多 6 个文件的限制：

`setquota -u qin 10240 20480 5 6 /mnt/qinlv`

- 10240 软空间限制
- 20480 硬空间限制
- 5 软文件数限制
- 6 硬文件数限制

测试：

`touch /mnt/qinlv/qinfile{1..9}`

`dd if=/dev/zero of=/mnt/qinlv/qinfile1 bs=1M count=22`

15.6 raid

15.6.1 磁盘阵列：保护硬盘和固态硬盘上应用程序数据的常用方法

RAID0 磁盘条带化（将存储数据的负载分散到了更多的物理驱动器上）：

最少需要两块磁盘

数据条带式分布

没有冗余，性能最佳(不存储镜像、校验信息)

不能应用于对数据安全性要求高的场合

RAID1 磁盘镜像（所有数据都被写入两个独立的物理磁盘。磁盘本质上是彼此的镜像。

如果一个磁盘出现故障，可以使用另一个磁盘来检索数据）：

最少需要 2 块磁盘

提供数据块冗余

所需磁盘空间增加了一倍

RAID5 奇偶校验：

最少 3 块磁盘

数据条带形式分布

以奇偶校验作冗余

适合多读少写的情景，是性能与数据冗余最佳的折中方案

RAID 10(又叫 RAID 1+0):

最少需要 4 块磁盘

先按 RAID0 分成两组，再分别对两组按 RAID 1 方式镜像

兼顾冗余(提供镜像存储)和性能(数据条带形分布)

在实际应用中较为常用

15.6.2 mdadm 选项:

mdadm 选项:

- C: 创建模
- n #: 使用几个块设备来创建此 RAID
- l #: 指明要创建的 RAID 的级别
- a {yes|no}: 自动创建目标 RAID 设备的设备文件
- c CHUNK_SIZE: 指明块大小,单位 k
- x #: 指明空闲盘的个数
- D: 显示 raid 的详细信息

管理模式:

- f: 标记指定磁盘为损坏
- a: 添加磁盘

-r: 移除磁盘

15.6.3 实验-raid0

添加 2 块磁盘，并分区

```
fdisk -l | grep "Disk /dev/sd"
```

```
mdadm -C /dev/md0 -a yes -l 0 -n 2 /dev/sdb1 /dev/sdc1 #创建新的软 raid
```

```
mkfs.xfs /dev/md0
```

```
mkdir /mnt/md0
```

```
mount /dev/md0 /mnt/md0
```

```
dd if=/dev/zero of=/mnt/md0/md0test bs=1M count=500
```

```
df -Th
```

```
mdadm -D /dev/md0 #输出指定 raid 详细信息
```

15.6.4 实验-raid1

```
mdadm -C /dev/md1 -a yes -l 1 -n 2 /dev/sdb2 /dev/sdc2
```

```
mkfs.xfs /dev/md1
```

```
mkdir /mnt/md1
```

```
mount /dev/md1 /mnt/md1
```

```
dd if=/dev/zero of=/mnt/md1/md1test bs=1M count=500
```

```
df -Th
```

```
mdadm -D /dev/md1
```

15.6.5 实验-raid5

```
mdadm -C /dev/md5 -a yes -l 5 -n 3 /dev/sdb3 /dev/sdc3 /dev/sdd3
```

```
mkfs.xfs /dev/md5
```

```
mkdir /mnt/md5
```

```
mount /dev/md5 /mnt/md5
```

```
dd if=/dev/zero of=/mnt/md5/md5test bs=1M count=500
```

```
df -Th
```

```
mdadm -D /dev/md5
```

15.6.6 实验-raid5 坏 1 块硬盘

```
dd if=/dev/zero of=/mnt/md5/md5file bs=1M count=1000
```

```
df -Th
```

```
umount /mnt/md5
```

```
mdadm /dev/md5 -f /dev/sdd3 #模拟/dev/sdd3 为坏盘
```

```
mdadm -D /dev/md5
```

```
mount /dev/md5 /mnt/md5/
```

```
ls -l /mnt/md5/
```

```
df -Th
```

```
mdadm /dev/md5 -r /dev/sdd3 #热拔/dev/sdd3
```

```
mdadm -D /dev/md5 #显示仅剩 2 块硬盘
```

```
mdadm /dev/md5 -a /dev/sdd3 #热插/dev/sdd3
```

```
mdadm -D /dev/md5 #快速查看可看到同步数据百分比变化
```

```
mount /dev/md5 /mnt/md5
```

```
ls -l /mnt/md5
```

15.6.7 实验-raid5 热备

```
mdadm -C /dev/md5x -a yes -l 5 -n 3 -x 1 /dev/sdb4 /dev/sdc4 /dev/sdd4
```

```
/dev/sde4 #x 指明空闲盘个数
```

```
mkfs.xfs /dev/md5x
```

```
mkdir /mnt/md5x
```

```
mount /dev/md5x /mnt/md5x
```

```
dd if=/dev/zero of=/mnt/md5x/md5xtest bs=1M count=500
```

```
df -Th
```

```
mdadm -D /dev/md6 #自动更换
```

