

# 安装 GNU/Linux



# 第一节 基础知识

## 1. 介绍计算机引导记录



BIOS：基本输入输出系统，固化在主板上

MBR:Master Boot Record, 主引导记录，位于硬盘的前 512 个字节，第一个分区的 0 扇区

# 第一节 基础知识

## 2. 硬盘相关概念的介绍

### IDE 硬盘

通常的计算机都是 IDE 硬盘

一台计算机最多有两条 IDE 总线

每条 IDE 总线最多可以挂接 2 个 IDE 设备

一台计算机最多有 4 个 IDE 设备

第一个 IDE 接口主设备 `=/dev/hda`

第一个 IDE 接口辅设备 `=/dev/hdb`

第二个 IDE 接口主设备 `=/dev/hdc`

第二个 IDE 接口辅设备 `=/dev/hdd`



# 第一节 基础知识

## 2. 硬盘相关概念的介绍

SCSI Small Computer System Interface  
每台计算机上可以有多个 SCSI 通道  
每通道上可有 16 个 SCSI 设备，包括 SCSI 卡。  
即最多可以挂接 15 块 SCSI 硬盘

第一个 SCSI 设备 = /dev/sda  
第二个 SCSI 设备 = /dev/sdb  
第三个 SCSI 设备 = /dev/sdc  
.....



# 第一节 基础知识

## 3. 活动硬盘 /U 盘 /SSD 硬盘

USB 活动硬盘属于热插拔技术。在 Linux 世界中将其认为 scsi 类型设备，因此对于 USB 活动硬盘也属于

`/dev/sdx`



# 第一节 基础知识

硬盘的容量、柱面数、磁头数、扇区数

C/H/S, Cylinder , Head , Sector

柱面 / 磁头数 / 扇区数

1. 磁头数表示硬盘总共有几个磁头，也就是有几面盘片，最大为 255
2. 柱面数表示硬盘每一面盘片上有几条磁道，最大为 1023
3. 扇区数表示每一条磁道上有几个扇区，最大为 63. 每个扇区一般是 512 个字节 ( 1M = 1048576Bytes)

所以磁盘最大容量为：

$$255 \times 1023 \times 63 \times 512 / 1048576 = 8024\text{MB}$$

# 第一节 基础知识

## 分区概念

为了让硬盘更好的组织数据，为了分开不同的数据，可以给硬盘分区 (partition)

基本分区 ( Primary Partition )

扩展分区 ( Extended Partition )

逻辑分区 ( Logical Partition )



# 第一节 基础知识

## 分区数量

在磁盘在基本状态之下，仅允许拥有 **4** 个分区的数量。主要原因：

在 **0** 扇区上的 **512** 字节中 **1-446** 字节分配给了 **MBR**

剩余的 **66** 字节中有 **64** 个字节可以存储分区表即： **$16 \times 4 = 64$**

最后 **2** 个字节是模数💬



# 第一节 基础知识

## Linux 分区命名法则

第一条 IDE 线上的第一个 IDE 硬盘的第一个 <b>基本</b> 分区	/dev/hda1	SCSI 的第一块硬盘的 第一个 <b>基本</b> 分区	/dev/sda1
第一条 IDE 线上的第一个 IDE 硬盘的第一个 <b>逻辑</b> 分区	/dev/hda5	SCSI 的第二块硬盘的 第一个 <b>逻辑</b> 分区	/dev/sdb5
第一条 IDE 线上的第二个 IDE 硬盘的第二个 <b>基本</b> 分区	/dev/hdb2	SCSI 的第三块硬盘的 第二个 <b>基本</b> 分区	/dev/sdc2
第二条 IDE 线上的第二个 IDE 硬盘的第二个 <b>逻辑</b> 分区	/dev/hdd6	.....	.....
.....	.....		

# 第一节 基础知识

## 2. 安装步骤

挂载点：

1. 目录树的概念
2. 从根开始，唯一的根目录
3. 每个文件系统（分区）安装到 Linux 目录结构树上的一个分支，安装点就是挂接点。
4. 安装时，至少给根分区指定一个挂接点
5. 交换分区不需要挂接，也不需要挂接点



# 第一节 基础知识

文件系统 File System:

用来组织、存储文件的信息、权限等数据的一种方式

每个文件系统都有自己的特点。格式化即完成文件系统的建立



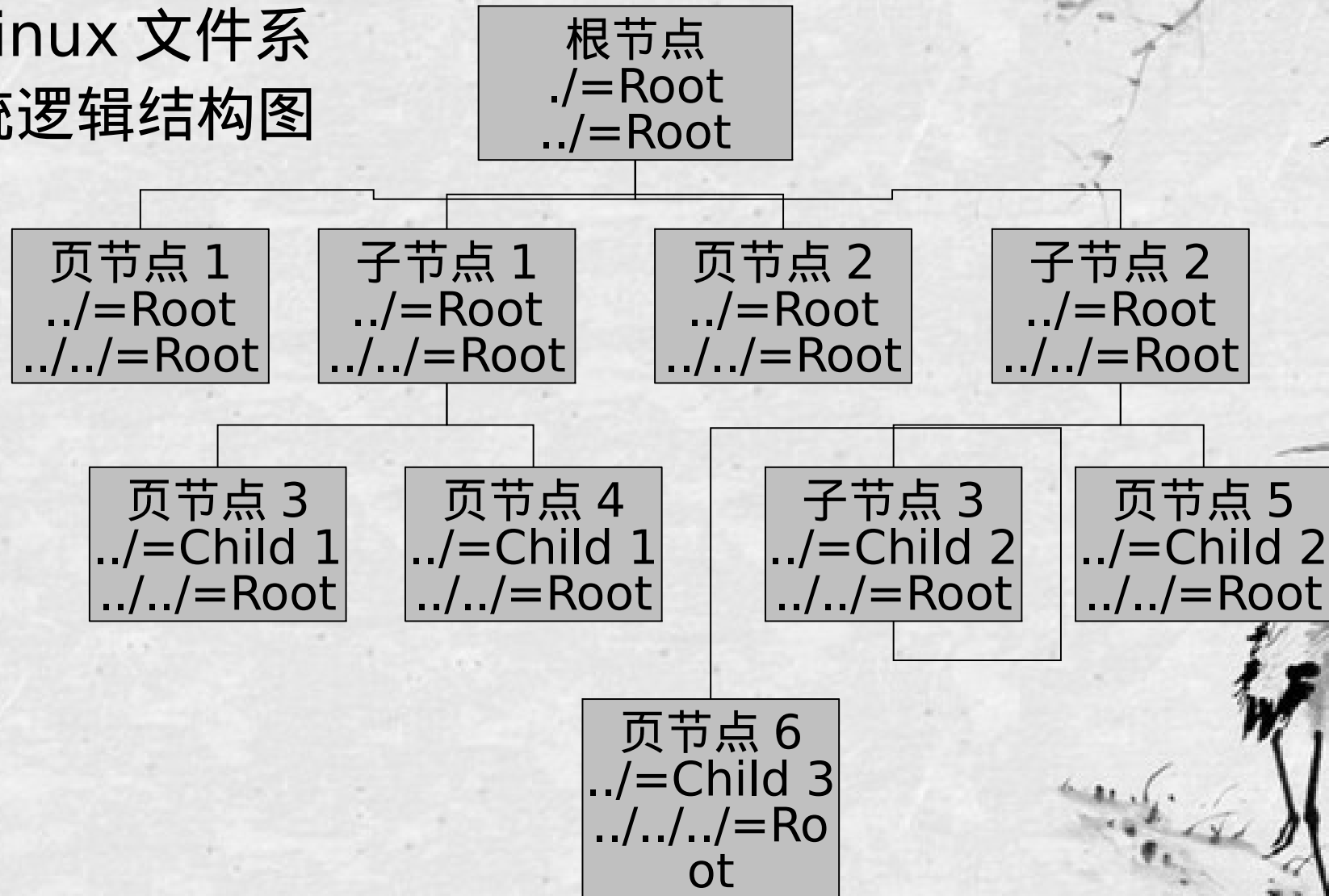
## 第一节 基础知识

文件系统是一种逻辑结构，它提供了一种框架，用于在随机访问存储设备中存储和获取信息。在 Linux 环境中，所有的信息都被视为文件，因此，基本上系统中的所有事务都要通过文件系统进行处理、存储和获取。

OS 要直接负责维护文件系统，因为文件系统是复杂的逻辑重叠，而不是简单的底层空间的划分。这还意味着就文件系统内部管理的信息种类来说，文件系统是相当灵活的且广泛的。文件系统存储了相当多的元信息（从字面上理解，就是关于信息的信息），它们不仅与其本身结构有关，而且还与所管理的文件有关。

# 第一节 基础知识

## Linux 文件系统逻辑结构图



## 第一节 基础知识

Linux 文件系统通常都是分层次的，它们是由在一个倒置的树型结构中所组织的多个目录组成。倒置的树型图使用核心节点或者顶部的根节点来描述。目录称为“节点”。节点可以包含子节点或其他称为“子节点”的子节点，不包含子节点的节点成为“页节点”。

每个目录都包含指向本身的“.”和指向其父目录的“..”的引用。即使是初始节点——或者目录结构的根——也有对其父目录的引用；在这种特殊的情况下，父引用只是指向其本身。

# 第一节 基础知识

文件系统创建时，预留了标记为“ Inode” 的区域。

Inode 即 i 节点，是一种特殊的文件，供内核读取文件信息用。这些信息包括文件允许访问的权限，所有者，创建日期等。



## 第一节 基础知识

Inode 结构本身就是在进行最小配置时进行实例化的。所有，即使在不用的情况下，i 节点都会存在，并是可用的。

虽然最后的结果是不能在创建了文件系统之后添加更多的 i 节点。

当然，分区中存储空间的数量是有限的——为 i 节点预留的空间越多，用于存储文件的空间就越少。反之，如果 i 节点用完了，就不能创建更多的文件了。即使磁盘空间仍然足够的多



## 第一节 基础知识

标记为“File Blocks”（文件块）是分区空间中剩余的部分。

尽管连续存放数据块的排列方式最有效，但文件的增减和删除（甚至修改）具有不确定的因素，所以加大了管理这一区域的难度。因此，文件块通常是不连续的，而且一个文件可能会分散存放在多个物理位置。

文件系统的任务是提供组织化的结构，供操作系统解释i节点列表、自由块（或称空闲空间）列表和超级块中的信息所有。一旦指定了结构，操作系统就能指定数据I/O，并管理所有后缀产生的信息，其中包括自由块、i节点列表和超级块信息等。

# 第一节 基础知识

## 超级块 (Super Block)

拥有掌握 Linux 王国的金钥匙。没有它的索引和列表。文件系统无法调用文件。

操作系统必须用超级块来判断哪些分区对应的块中包含着正在使用的文件系统，哪些分区可用于写入新的数据。

因此许多文件系统都有备份超级块的重要原因。

如果超级块发生冲突或被损坏，备份的超级块就可用于管理文件系统。

# 第一节 基础知识

## 块回收与利用

当文件被删除时，数据块和i节点只是简单地回到原来的剩余列表中。它们的内容不能变更、而且在改写之前可以恢复。



## 第一节 基础知识

Inode 作用：

系统上的每个文件都必须有惟一的标识。而且，系统上的每个文件的文件系统内部也如此。因此，文件系统是文件惟一标识符的第一部分，如通信地址中的街道名称一样。

i 节点编号是文件惟一标识符的第二部分，如通信地址中的房屋地址一样。它指向 i 节点，i 节点是实际保存文件物理位置及其其他相关信息的地方。

## 第一节 基础知识

Inode 作用：

目录只是某种特殊类型的文件。目录的特殊之处在于保存和 i 节点编号与所有文件名（特殊格式的和普通格式的）关联起来。此时，它充当的是一个容器。该信息保存在目录的数据块中，目录 i 节点的其余部分是一个标准问题。

位置条目列表表明哪些位置目前已填入了下一段（segment）。这是一个仅供内部使用的结构。类似于（用于增强存储性能和减少存取时间的）自由 i 节点列表。

# 第一节 基础知识

Inode 作用：

目录条目是目录文件具有特殊功能的关键。

尽管系统提示所有位置都已填满，但它们都是不连续的（这是因为文件已经被移走或删除）。

但对于目录而言，第一个条目始终为“.”，表示当前目录，可通过当前的目录内的（或下一级）相对路径访问。

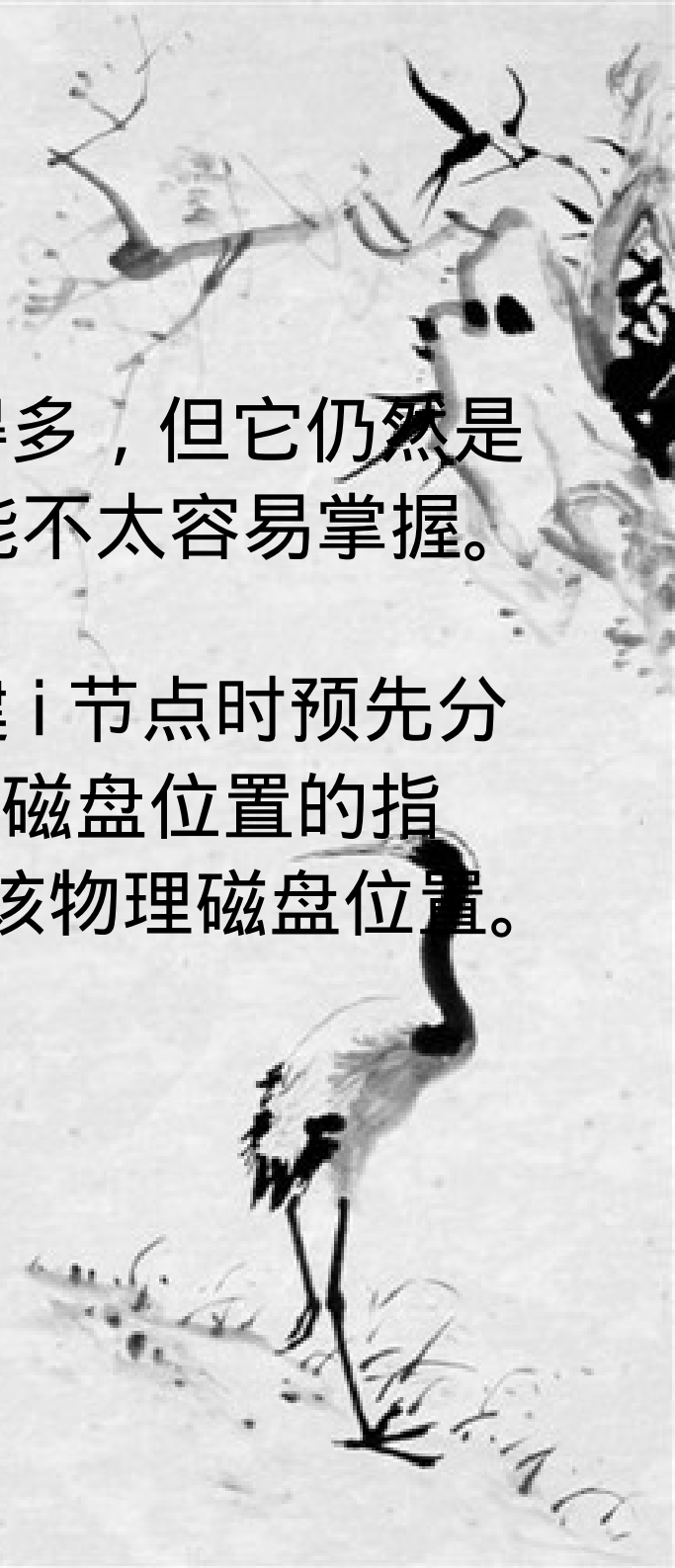
而第二个条目是始终为“..”，表示的是父目录，可通过当前目录的上一级相对路径访问。没必要列出其他条目（也就是说，它们不是按 i 节点编号或文件名进行排序的）。

# 第一节 基础知识

文件寻址：

虽然它比其他的文件系统结构复杂得多，但它仍然是较为简单的结构，但其物理定址方案可能不太容易掌握。

如果文件相当小，其地址可以用创建 i 节点时预先分配的“直接块”来处理。直接块是指向物理磁盘位置的指针，文件的直接（ literal ）位就保存在该物理磁盘位置。



# 第一节 基础知识

## 引导块 (boot block)

用于存储引导时的信息，存放操作引导信息的块

## 文件锁机制

锁定标志的存在是为了避免竞争条件 ( race condition )。i 节点必须在指定时间由特定的某个过程进行处理。锁定标志可用防止多个进程修改文件系统或同时修改 i 节点中的数据。没有锁定标志，某个进程就可能会要求 i 节点，但该节点已被另外一个进程占用，另一个进程在第一个进程返回执行写操作之前，已要求了该 i 节点并向其中写入了数据。这样做的后果：要么其中一个进程的数据消失，要么就是两个进程的数据冲突。



# 第一节 基础知识

## RH 中常见的 Linux 分区类型

ext2 文件系统类型  
ext3 文件系统类型  
ext4 文件系统类型  
Swap 文件系统类型



# 第一节 基础知识

ext2:

是一个非日志式的文件系统，但所有的 Linux 都可以非常好的支持



# 第一节 基础知识

ext3:

是一个日志型的文件系统，在 ext3 中多了几个名为 kjournald 的常驻程序在执行，它的功能主要是记录文件系统的改变情形，避免在不正常关机或卸载文件系统时造成数据的损失或数据的不连贯。因此，若不正常关机，再重新开机时，系统并不进行硬盘扫描。

## 第一节 基础知识

ext3 特性：

(1) 稳定性：不正常的关机或是卸载文件系统 `umount file system` 时，可通过 `kjournald` 快速复原，它不需要通过扫描整个文件系统或文件数目来确认文件的一致性。只有在硬件发生损坏时，才需要作一致性的确认。

(2) 数据的完整性在不正常关机或者卸载文件系统时，可确保数据的完整性。ext3 文件系统允许用户选择数据保护的种类以及等级。

# 第一节 基础知识

ext3 特性：

(3) 存取速度 :ext3 的 kjournald 可以调整硬盘磁头的动作。



# 第一节 基础知识

ext4:

一个 ext3 升级版日志型的文件系统



# 第一节 基础知识

ext4 特性：

(1) 直接从 ext3 升级，无需格式化

(2) ext3 仅支持 16TB 文件系统，仅支持 2TB 独立文件，但 ext4 支持 1EB( $EB > PB > TB > GB$ ) 文件系统，独立文件支持 16TB

(3) ext3 仅支持 32,000 子目录，ext4 则无限

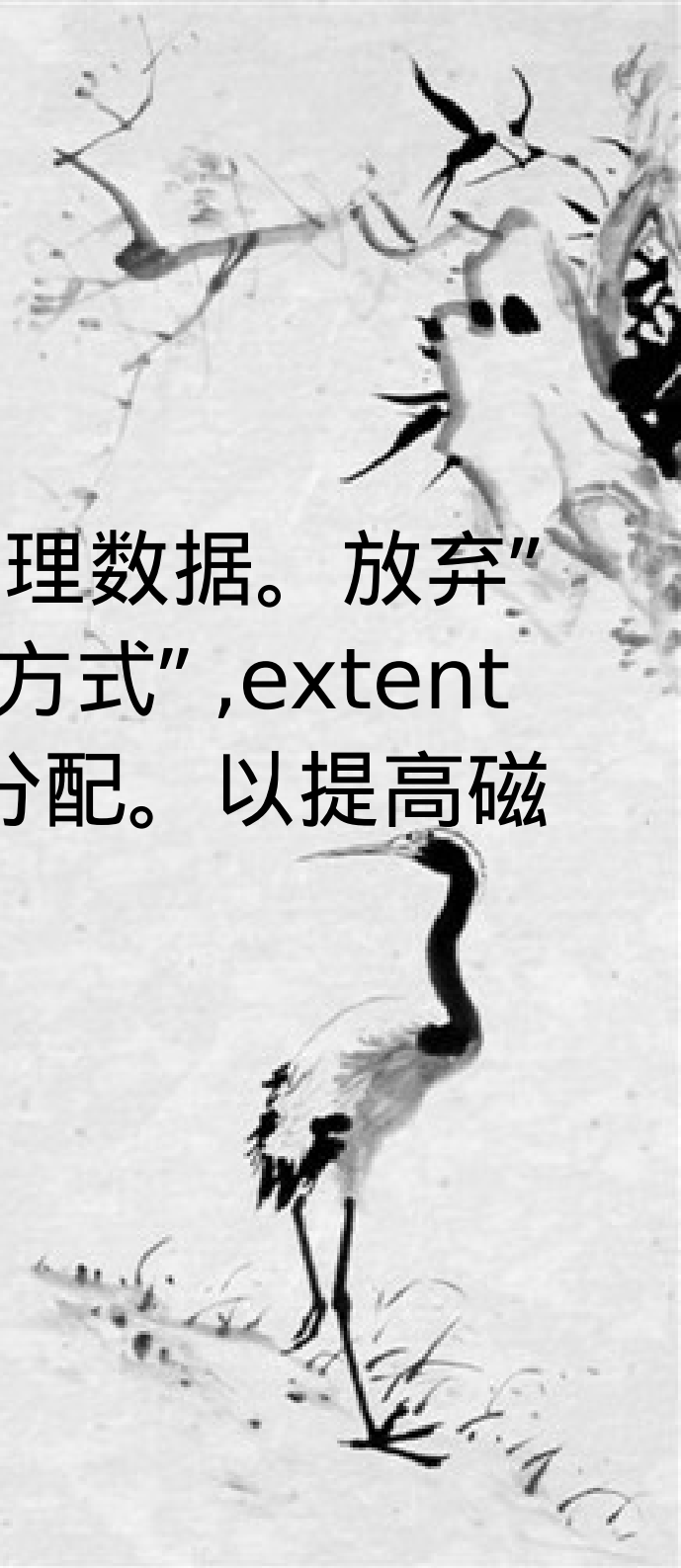
## 第一节 基础知识

ext4 特性：

(4) 引入 Extents( 区域 ) 技术，管理数据。放弃“文件写入时预先建立、分配块”存储方式”，extent 则是将一组块存在此区域中，直接分配。以提高磁盘效率

(5) 更好的利用 CPU 资源

(6) 快速 fsck 等等





## 第一节 基础知识

Swap:

swap 被成为虚拟内存，在 Linux 世界中称之为交换分区。

SWAP 可以是一个分区，也可以是一个文件



## 第二节 初步安装 **Linux**

RedHat Enterprise Linux

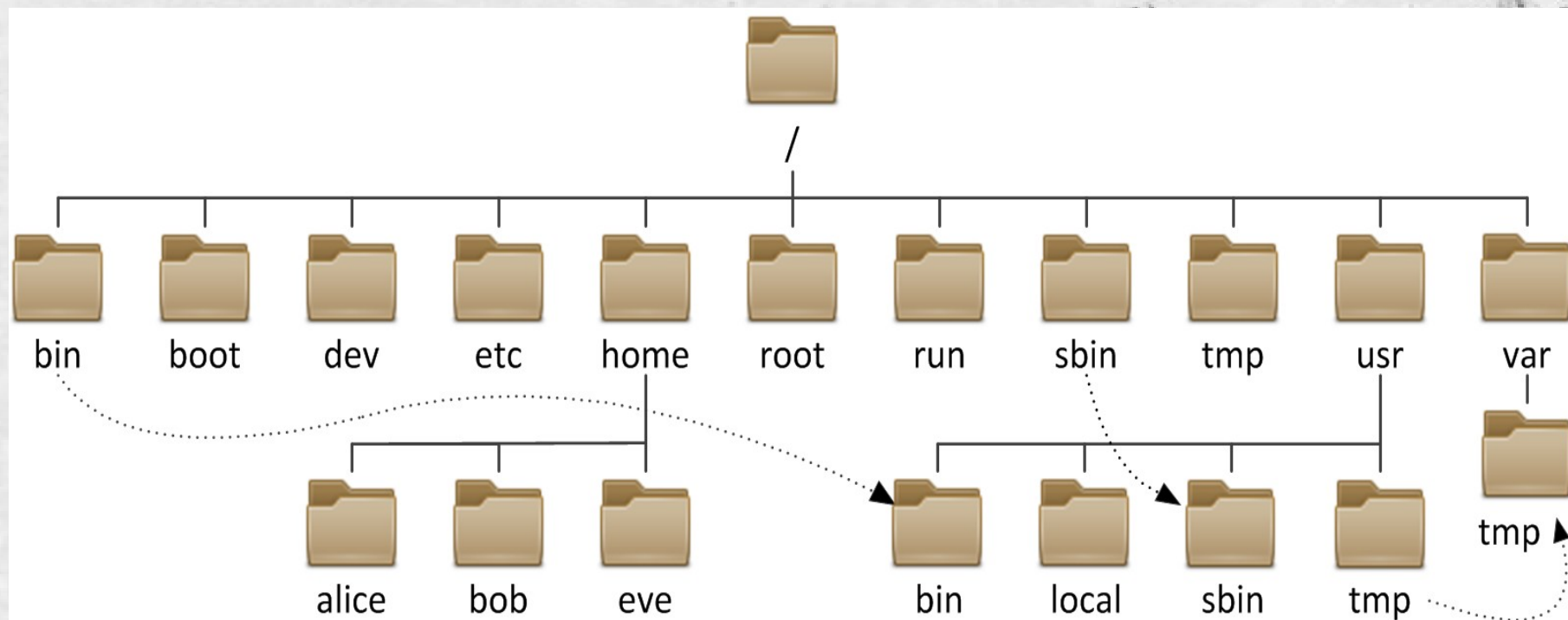
OR

CentOS



### 第三节 Linux 目录

Linux 目录：



## 第三节 Linux 目录

### Linux 重点目录介绍：

目录名称	功能介绍
/usr	第三方软件安装，共享库，包含文件及文件、静态只读数据。其中重要的子目录有： <ul style="list-style-type: none"><li>- /usr/bin: 用户命令</li><li>- /usr/sbin: 系统管理命令</li><li>- /usr/local: 本地制定软件</li></ul>
/etc	保存系统指定的配置文件
/var	在系统启动时所需要的变量数据，改变文件相关信息也在其目录下被规范。
/run	进程执行的信息保存在此目录中，并存有进程 ID 及锁文件。其在以前的版本中为 /var/run 及 /var/lock
/home	用户主目录

## 第三节 Linux 目录

### Linux 重点目录介绍：

目录名称	功能介绍
/root	管理员主目录
/tmp	保存大量的临时文件，存留 10 天及以上的文件将被从此目录中删除。 另外，系统中还有另一个临时目录：/var/tmp，此目录下的文件经过 30 天没有进程任何操作则将被自动删除
/boot	Linux 系统所需启动引导文件存放目录
/dev	包含特定的设备文件以可以使操作系统进程硬件控制
/var/log	存放系统及各服务程序等的日志信息

## 第四节 关闭防火墙 /**SELinux**

防火墙与 SELinux 都属于 Linux 的安全防御工具，为保障后续实验能够顺利完成，故先将两者关闭。

### 1. 关闭防火墙

```
#systemctl stop firewalld.service
```

### 2. 禁止防火墙开机启动

```
#systemctl disable firewalld.service
```

## 第四节 关闭防火墙 /**SELinux**

防火墙与 SELinux 都属于 Linux 的安全防御工具，为保障后续实验能够顺利完成，故先将两者关闭。

### 3. 关闭 SELinux

```
#nano -w /etc/selinux/config
```

将

```
SELINUX=enforcing
```

改为

```
SELINUX=disabled
```

保存并退出 : ^x->y-> 回车



## 第五节 系统关闭与重启

Linux 的启动分为 7 个运行级别

级别	说明
0	关机
1	单用户 
2	多用户级别 (无 NFS) 
3	多用户级别 
4	未定义 
5	图形化, 多用户 
6	重启



## 第五节 系统关闭与重启

自 RHEL7 后，采用了 systemd 的方式进行了系统的级别管理。因此它的运行级别将逐步消失。虽然保留了过去的级别代码，但实际上不在真实存在。取而代之的则是各种功能的目标文件来完成相关工作。

后续将会有详细介绍



## 第五节 系统关闭与重启

命令格式：halt

功能：关闭系统

语法格式：halt < 选项 >



## 第五节 系统关闭与重启

选项：

-p: 关闭机器的时候关闭电源

-n: 在关闭的计算机的时候不同步数据 

-w : 不做关机的操作，只是把这个操作记录到日志里

-d: 不把关机的信息记录到日志文件里 (-n 参数包含此项)

-f: 不调用 shutdown



## 第五节 系统关闭与重启

命令 :poweroff

功能：关闭系统

语法格式： poweroff [ 选项 ]



## 第五节 系统关闭与重启

选项 :-

-n -w -d -f -p 这些参数和 halt 的参数意义相同

-h: 在关机前将所有的硬件转到待机模式

注：这个选项是针对 IDE 硬盘



## 第五节 系统关闭与重启

命令 :shutdown

功能：关闭 / 重启系统

命令格式： shutdown < 选项 > [ 时间 ] [ 信息 ]



## 第五节 系统关闭与重启

选项：

- t：这个 -t 参数后面需要加数字比如 5，表示在发出关机信号 5 秒后开始执行关机工作
- k：不做真正的关机，只是通知一下所有的用户。
- r：重启计算机
- h：关机后关闭电源



## 第五节 系统关闭与重启

-f 在重新启动的时候跳过 fsck( 磁盘检查程序 )

-c 取消已在执行的关机动作





## 第五节 系统关闭与重启

命令 :reboot

功能：系统重启

命令格式： reboot [ 选项 ]

reboot 所有的选项同 halt 一样，除了 halt -p 以外其它的选项含义都是一样。只要记住 halt 的 -n -w -d -f 这几个参数的作用那 reboot 参数的功能想忘记也不行。

## 第五节 系统关闭与重启

命令 :init

功能：关闭或重启系统

语法结构：init [ 运行级别 ]

运行级别 :0-6



## 第五节 系统关闭与重启

shutdown/init 区别

简单来说

shutdown 可以发送信息，并提示用户

init 直接启动相应的运行级别，而没有提示信息

