1. Linux 内核的几大功能

管理硬件

管理内存

管理流（输入输出）

管理文件系统（文件系统负责为内存分区，合理且高性能的存取数据）

1. Linux 下的用户 Home
   1. Linux 是多用户服务系统
   2. Home 放着一个个文件夹代表一个个用户 ，自己操作自己的文件夹从而实现 多用户同时操作
2. Linux 如何管理分区 dev

注意： linux下 所有的东西的（打印机、键盘、硬盘等） 都是 文件（设备）

文件分类 ：

字符设备

块设备（二进制形式读取数据 ）

* 1. Windows 分区 CED 盘
  2. Dev :用于装硬件设备

4、Linux 的分区命名 /dev/xxyN

* 1. dev 是所有设备文件的所在分区目录
  2. xx 表示分区所在设备的类型 通常是 hd（ide） /sd（scsi）
  3. Y 表示 设备所在第几个磁盘 如：/dev/hda 第一个ide磁盘 /dev/sdb 第二个scsi磁盘
  4. N 代表分区数字 主分区或扩展分区 1-4，逻辑分区永远从5 开始

5、如何使用命令

* 1. 有些命令 需要参数 [] 表示可选的参数 不带[]表示必选项
  2. 不知道命令怎么用 xxx --help
  3. Man xxx 会详细显示 该命令怎么能用 退出 man 直接 q

6、常用的命令

Ctrl + c 结束执行命令

Bye 退出一个命令 如 ： ftp

Ls 目录名 列出当前目录下的 目录和文件

Ls -l 目录名 竖着列出目录

Ls -R（R大写）目录名 列出 树 型目录

Cd /xx 进入某一目录

Cd / 进入根目录

Cd .. 进入上层目录

Cd. 当前目录

Pwd 当前路径是什么

Mkdir 创建目录

Rmdir 只能删除目录为空 目录

Rm -rf XXX 删除目录和文件 remove -r（递归）f（force）

Touch xxx 创建文件 xxx 文件为空 （注意：在linux下 文件的扩展 名意义不大，大多可执行的文件 只有一个文件名 或.sh 结尾）

Vi xxx（文件名） 往文件中添加 内容（文本编辑器） 如果当前文件名存在直接进

入编译，如果没有 创建一个新的 进入编辑

Vi的两种模式 ： 命令模式 （默认刚创建完文件进入编辑页面）

dd 删除一整行

dw 删除一个 word

o 往下插入一行

O（大写） 往上插入一行

编辑模式 切换编辑模式 敲 a （append ） 切换到 命令模式下 按esc键盘

： 回到 文本编辑器的最下方

w 保存

然后 再 : 清空命令

然后 q保存 退出 编辑器 /q！ 不保存退出

Vim 打开文件

Wc 统计执行文本的行数、字数、字符数

Grep 在指定文本中查找 包含指定的字符串 那行数据 （检索目标行命令）

More xxx（文件名） 查询文件内容 （ vi xxx是进入编辑器）

Cat xxx（文件名） 查询文件内容(正序)

tac xxx（文件名） 查询文件内容(倒叙)

Head -3 xxx 用于 文件内容的 特定前3 条显示 （大文件）

Tail 用于 文件内容的 特定后3 条显示 （大文件）

Chkconfig 命令主要用来更新（启动或停止）和查询系统服务的运行级信息

Chkconfig --list 列出所有服务 有6列 代表 6 个启动级别

Find 按照文件名字/日期/大小等 查找文件在什么位置上

例： 按名字 找

find / -name \*关键字 / 表示从根目录开始找 （可以具体）

-name 表示按照名字查找

\* 模糊查找

Whereis 命令 解释当前命令 当前的状态 ，哪里来的 ，怎么使用

Echo $path (path : linux环境变量 linux下的分隔符是：)

Ln xxx（文件） xx 创建xx 链接 到 xxx文件 （硬链接 赋值文件 ）

Ln -s xxx（文件） xx 创建xx 链接 到 xxx文件 （软链接 相当于 windos下的快捷方式 ）

Cp n1 n2 n1 文件名 n2路径 ，将n1 拷贝到 n2路径下

Cp -r n1 n2 将1目录拷贝名字为2目录 （注意：当操作文件时要考虑 到递归）

Mv xxx 目录地址（相对或绝对） 移动（拿走） move

Ps -ef 查看linux下的所有进程

exit 退出终端

Shutdown 关机

Reboot 重启

在linux中有shell ， 用于编译命令 然后 将编译后的 命令传输至 linux 内核 在去执行

添加用户 删除用户（管理员）

Useradd 用户名（每当添加一用户 就会在home目录多一个对应的 目录，还会在etc 目录多一个 passwd文件 里面存了 所有的用户信息）

Passwd 用户名 指定密码

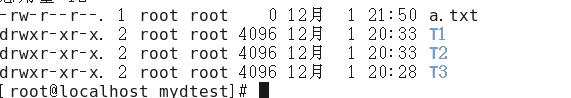
文件权限（很重要）

Chmod 权限 文件名 ： 修改权限 change modify

Chown 用户名 文件名 ： 修改 文件的所有者 change onwer

1. Linux 下 对权限的 分类有4种
   1. Read
   2. Write
   3. Execute 执行
   4. None - 不能 做 r w e 操作

B、



总共有10 个字母 表示 ：谁 对什么样的文件 有什么样权限

开头首字母 ：

- 开头 是文件

d开头是目录

L 开头表示是个链接

后面有9位 分成3 组，每组有3位字母

如 ： -rw-r--r-- ： 文件的所有这 rw ，同组的人 只r ，其他组的人 只r

第一组 ：表示文件的 所有者

第二组：同一组的人 对此文件有什么权限（用户可以分组）

第三组：剩下的人对 此文件 有什么权限



管道 | 表示 将上一个命令执行的结果 交给下一个命令去执行

命令替换 `命令`

Wall 通知所有人 warning all

重定向输出 > 、>>、2> 错误重定向 将一个命令输出的结果 ， 输入到 指定的文件中

如： ls > aa.txt 将当前目录下的文件 所有内容输入到 刚建的 aa.txt文件中

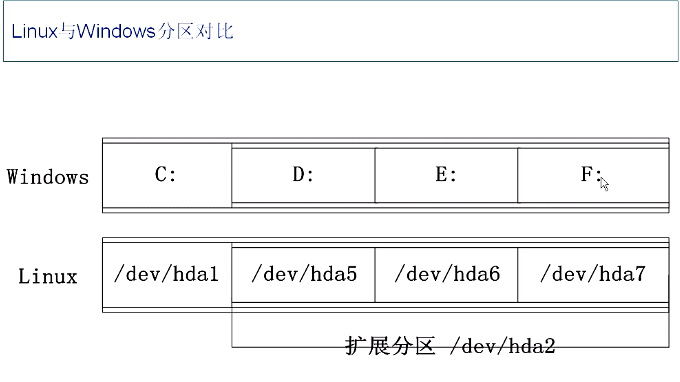
（一个> 号 输进去一份, 两个> 输进去两份）

2> 错误重定向 ：当一个命令错误时 ，也会执行 将错误提示信息 输入到 创建的文件中

重定向输入 <

6、如何使用命令

* 1. 有些命令 需要参数 [] 表示可选的参数 不带[]表示必选项
  2. 不知道命令怎么用 xxx --help
  3. Main xxx 会详细显示 该命令怎么能用 退出 man 直接 q



注意 ： linux 至少有两个分区

1. / 根分区
2. SWAP 交换分区（拿硬盘当内存使用） 虚拟分区 默认大小 是 实际物理内存的两倍

为什么 ？在 linux 下 有些特殊的程序会认为 物理内存不够使用 ，不管物理内存多大都会把数据放在swap中

8、 Linux 的启动程序顺序 boot sequence （重点）

* 1. Load BIOS 检测硬件
  2. Read mbr’s config to find out the os 读取物理磁盘的第一个扇区第一个磁道 中的OS 操作系统
  3. Load the kernel of the os 加载要启动系统的核心文件
  4. Init process starts 然后初始化第一个进程
  5. Execute etc 执行一个命令启动（etc/rc.d/sysinit目录下 有系统配置文件和启动序rc ：run command
  6. Start other modules 加载系统中的各个模块 （用户管理模块、硬盘管理模块等）
  7. Execute the run level scripts linux 的系统的启动 需要依次启动多个层次（最多有6 层）
     + 1. 6个层次 run level 命令执行 /etc/inittab（目录） 重要
          1. 0 系统停机状态
          2. 1用户工作状态
          3. 2多用户状态（没有nfs ：不可以联网）
          4. 3多用户状态（没有nfs ：可以联网）
          5. 4系统未使用 开始让用户使用
          6. 5图形界面
          7. 6系统正常关闭 重新启动
  8. Execute/etc/rc.d/rc.local（重要）执行这个文件程序 此脚本可以修改配置属性让tomcat自启动
  9. Execute /bin/login 登录程序启动
  10. Shell started 启动了

1. 使用FTP实现 windos 和 linux 之间 文件的共享
   1. FTP 是实现 windows 和 linux之间数据共享的服务
   2. Vsftpd、proftpd 是具体的服务器名子
   3. ftp 中的两中用户
      1. 本地用户
      2. 匿名用户
   4. 安装 FTP 步骤
      1. Yum search vsftpd
      2. Yum install nsftpd.版本号
      3. Yes
      4. Complete（完成） service vsftpd status 查看当前 ftp的状态
      5. Service vsftpd start
      6. Service vsftpd stop
   5. 设置 vsftpd的配置 （安装目录在 /etc/vsftpd目录下）
      1. Conf 主配置文件
      2. Ftpusers 指定哪些用户 不能访问 ftp服务器
      3. User\_list 指定哪些用户 不能访问 ftp服务器

注意：两个不同文件只是拒绝的时候出现的情况不同 “拒绝”是相同的，如果需要给某一用户 具有访问的权利 必须同时修改两个 配置文件

1. Windows 连接 linux
   1. Linux下查询IP地址 ifconfig
      1. Lo local loopback
      2. Eth0 网卡1
      3. Eth1 等 网卡2
   2. 找到 linux 的本机IP
   3. 用windows dos命令ping linuxIP 看是否ping通
   4. Dos 下 ： ftp 192.168.1.14 ping linuxIP 连接 linux上的 htp 服务
   5. 如果 不能ping 通， 可能被 linux 的 防火墙拦截 需要关闭linux的防

火墙， 获取开发端口

关闭 防火墙 ： service iptables stop

* 1. 登录 ： ftp 登录使用 用户名：anonymous 密码默认没有
  2. 登录 后 会有pub 目录 ，pub目录 用于放文件
     1. 位置在 个目录下/var/ftp下
  3. 设置某个用户（默认只有一个synonym可以登录）拥有 连接到ftp服务的 权限 如： 设置 root可以连接 etc/vfstpd /vfstpd .conf
     1. 将vsftpd/user\_list文件中 root 注释
     2. 将vsftpd/Ftpusers 文件中 root注释
  4. 设置 vsftpd 在系统刚启动的时候 就默认启动
  5. Window连接ftp服务器 ， doc命令 ftp eth0地址 netaddr 地址， 用户名 root 密码 mayadong

1. linux下安装 JDK
   1. 装rpm 类型的文件 执行命令： rpm -ivh 文件名.rpm
   2. 配置 环境变量
      1. JAVA\_HOME=绝对路径

注意：配置环境玩java\_home ,只能本用户可以使用 ，如果开放 执行 export JAVA\_HOME

1. Linux 安装Tomcat tar.gz (需要解压) 压缩了两次
   1. Gzip -d 问价名 将 tar.gz 解压成 tar 压缩文件
   2. tar -xzvf 文件名 将 tar 压缩包解压成 目录文件（解压后直接能用）
   3. 为目录文件 重命名 命令还是mv
   4. 启动 taomcat ./start.sh
   5. 关闭 tomcat 进程 杀出进程（ps -ef） kill 进程id
   6. 命令上网 wget http://localhost:8080
2. Linux 下安装 Redis
   1. 解压 redis.tar.gz tar -xzvf 文件名
   2. Make 安装
   3. 注意1： 可能会 没有 gcc 编译器 解决 ：装 gcc编译器
   4. 注意2： 可能会 报没有文件或目录的 错误 ， 需要清除 一些残存的文件 解决：make distclean 再make

I、Make 根据Makefile编译源代码，连接，生成目标文件，可执行文件

II、make distclean类似make clean，会清除上次的make命令所产生的object文件（后缀为“.o”的文件）及可执行文件，但同时也将configure生成的文件全部删除掉，包括.config文件等

* 1. 二次 Make 后会提示 make test （不要执行）
  2. 第一次装redis liunx下会缺少 tcl 插件

解决 ： wget http：//downloads.sourceforge.net/tcl/tcl8.6.4-html.tar.gz

* 1. 安装完成后 默认的安装目录在 usr/local/bin 目录下
  2. 装玩redis 后修改 redis.conf配置文件 general 的daemonize 为yes

默认位NO ： 类似的 tomcat的 run.bat 执行会出现 黑窗口 ，黑窗口 关掉 tomcat服务会关闭 ， daemonize 设置位yes 窗口关闭 不会把 redis 关掉

* 1. 启动 redis 命令 ：redis -server /redis.conf
  2. 进入redis 客户端 ./redis-cli
  3. ping 是否启动成功 redis-cli -p 6379
  4. Ping 显示 pong 则运行成功

1. Redis启动后的 注意事项
   1. Redis 是单进程的， 用单进程模型来 处理来自客户端的请求。

接收到请求后 需要进行读写 ，读写的进行 是通过 linux 的epoll函数来处理的 ，（epoll依赖主线程 linux 的内核）

* 1. 什么是 epoll ：用来 提高程序 在大并发 连接中 只有少量的用户使用的情况下 占用系统CPU的利用率
  2. Select 切换数据库 Redis 默认装了 16 个库（数据库用0-15来命名 ）默认是在 第0个数据库 如果选择数据库 select 数据库序号
  3. Dbsize 查询当前库长数据二蛋条数
  4. Keys \* （查询所有的 key ） 查询所有 相当于 select \*

Keys 关键字? 模糊查询

* 1. Flushdb 清除当前库中所有的 数据
  2. Flushdball 清除所有库中所有的 数据

1. Redis 的5大数据类型
   1. String 最基本的数据类型 key/value
      1. string类型是二进制安全的（二进制安全： 任意的数据类型 包括图片、序列胡化对象）
      2. 一个string的字符串 最大可存512 M
   2. HSAH 类java中的 map <object，object>
   3. List 有序的字符串列表 可以在头部 、 尾部插入值
   4. Set 无序
   5. Zset 和set类似 ，区别在 set 前关联了double类型的 分数
      1. 游戏分数的 排名 如1万分 的有多少人 （取区间）
2. Redis 命令参考大全 http://redisdoc.com
3. 对于Key 关键字的操作
   1. Keys \* 查询所有
   2. Key 关键字 ？ 模糊查key
   3. Exists key 判断 是否存在 1：有 0 没有
   4. Move key 数字 将当前的键值 移到指定的库
   5. Explre key value 秒 key设置存活时间 （-1 表示永不过期，-2 表示已过期 移除内存）
   6. Ttl key 查看当前 key 还能存过多久
   7. Del key 删除key 1 成功 0 失败
   8. 注意： 如果 当前库中已经 有存在的key ，如果在set 相同名字的key进来，会把key中 value的值覆盖
   9. Type key 查看key中value值的数据类型
4. redis 字符串 string
   1. Append key 字符 拼接value的字符
   2. Incr key (注意key中value的值 如果是数字 那么 每incr一次 value的值就会 +1)
   3. Incrby key 数值 每次加 数字
   4. decrby key 数值 每次减 数字
   5. Getrange key n1 n2 n1 n2 是数字 ， 按照位置截取 value的值

注意：0 ，-1 是截取全部

* 1. Setrange key n1 值 将值往 n1 后面拼接进去
  2. Setex key value set key的值 ，如果key已经存在将set不进去返回0，如果key不存在可以set进去返回1
  3. 示
  4. senx

1. jedis 客户端连接 redis
   1. 如果是maven工程，需要指定坐标
   2. 创建jedis 客户端对象 Jedis a=new Jedis(ip,端口);
   3. 调用jedis命令操作redis数据库
   4. 关闭jedis
   5. 如果连接不上redis 、可能跟防火墙、redis 本身IP配置是否允许配置有关
      1. 将redis.conf : # bind 127.0.0.1 允许外部机器ip访问
      2. 将redis.conf protected-mode no 不保护redis
      3. 将redis默认的端口号注册到 防火墙中

单机版 redis 默认有16个库 0-15 ，集群版 默认一个库 0

Usr 类似 windows下C盘的 program file文件夹 ， 程序的 默认安装路径

Redis 实现主页面的缓存

1. 为了实现系统的高可用 ， redis需要进行 主备 实现（redis集群搭建）
   1. 安装redis
   2. 启动redis ：检测redis 是否启动netstat -Intp | grep 6379 （必须在redis安装的根目录） usr/local
      1. 前端启动

在安装的根目录下 ./redis-server 启动

后端启动 ：需要将redis.conf 配置 文件拷贝到 redis的安装目录下 ，然后修改配置文件

Daemonize yes ，然后启动 ./redis-server redis.conf

* 1. 集群（6个redis 实例：6个端口号 7001 - 7006） 设置3个主节点 3 个从节点
     1. 将redis 的安装目录 拷贝 6 个 至redis-cluster 下然后 分别删除 dump.rdb
        1. 修改 redis.conf 下的端口号 port 7001
        2. 将cluster-abled yes
     2. 重复6 次操作
  2. 将搭建集群的 ruby脚本拷贝到 redis-cluster目录下 ：redis-rtib.rb 在解压后的src文件夹下
  3. 启动6 个 redis 实例 （一个个单独启动比较麻烦） 创建 启动6 个的脚本
     1. Vim redis-start.sh
     2. 将 redis-start.sh 变成可执行文件 chmod +x redis-start.sh
     3. 运行 ./redis-start.sh
     4. 查询是否启动 ps aux|grep redis

1) ps a 显示现行终端机下的所有程序，包括其他  
2）ps u 　以用户为主的格式来显示程序状况。   
3）ps x 　显示所有程序，不以终端机来区分。

* 1. 创建集群 （ip是当前linux 的ip）

./redis-trib.rb create --replicas 1   
 192.168.1.156:7001  
 192.168.1.156:7002  
 192.168.1.156:7003  
 192.168.1.156:7004  
 192.168.1.156:7005  
 192.168.1.156:7006

* 1. 测试 三个主节点 三个从节点 ， 从任一一个 节点 都可以访问各个 节点里的数据

redis-cli -h 127.0.0.1 -p 7001 至 7002 -c （注意一定要加-c） 都能连通

2、然后向业务中天添加缓存