

Linux系统

- **命令1**

- 查看目录

- ls
 - -a la
 - 不隐藏任何以. 开始的项目
 - -l ll
 - 使用较长格式列出信息
 - -h
 - 与-l一起, 以易于阅读的格式输出文件大小
 - 这些命令可以混合使用 ls -alh

- 切换目录

- cd . 当前目录
 - cd .. 回到上一级目录
 - cd -
 - 返回上一次的目录
 - cd /
 - 回到根目录
 - cd ~ 和 cd
 - 返回用户所在地址
 - cd 文件和文件夹的绝对路径和相对路径

- 绝对路径与相对路径

- 从根目录开始的路径是绝对路径, 其他的是相对路径
 - 绝对路径: /home/xiowang/study_python/demo.py
 - 相对路径: ./study_python/demo.py

- 创建, 删除, 重命名文件和目录

- touch 文件名 创建文件
 - mkdir 目录名 创建文件
 - 创建所依赖的文件夹,递归创建多级目录
 - 使用-p
 - 创建目录时同时设置权限
 - -m
 - 显示详细执行过程
 - -V

- 设置目录安全上下文
 - -z
- rm 文件名或者 目录名 删除指定的文件或文件夹
 - 命令选项
 - -i
 - 交互式提示
 - -r
 - 递归删除文件和目录及其内容
 - -f
 - 强制删除忽略不存在的文，不会提示
 - -d
 - 删除空的目录时会用到
 - -v
 - 显示详细的执行过程
 - rmdir 目录名
 - 没有命令选项时，删除空目录
 - -v
 - 显示详细的执行过程
 - -p
 - 递归删除目录及其内部子文件
 - 复制和移动
 - cp 拷贝文件和目录
 - -i
 - 交互式提示
 - -r
 - 递归拷贝文件和目录
 - -v
 - 显示拷贝后的路径描述
 - -a
 - 保留文件的原有权限
 - mv 移动文件和目录及重命名
 - -i 交互式提示
 - -v 显示移动后的路劲描述
- 终端格式说明
 - command [-options][-parameter]
 - 命令和参数都是可选的，可以有多个命令和多个参数

- 选项分为长选项和短选项
 - 短选项 -a,-i
 - 长选项--help
- 查看命令帮助
 - 方式一
 - 命令名 --help
 - 方式二
 - man 命令名
- 重定向
 - 写入模式 > 文件名
 - 追加模式 >> 文件名
- **命令2**
 - 查看文件内容命令
 - cat 文件名 查看小型文件，可以同时查看多个文件
 - 可以结合重定向使用
 - more 文件名 查看内容较多的文件
 - 可以分页显示
 - 可以使用的命令
 - 下一行 按 Enter
 - 下一页 按空格 或者 F 键
 - 上一页 按 D
 - 退出 按 Q
 - gedit 文件名
 - 既可以查看，还可以修改编辑
 - 管道|
 - 终端数据的临时存储器
 - 不会产生文件，终端数据过多时，经常使用管道结合more命令来使用
 - 很多命令都可以结合管道使用
- 软连接和硬连接
 - 软链接
 - 类似 快捷方式，需要使用的是 **绝对路径** 作为连接条件
 - **同一片存储信息 只用一个指针，只是对指针的重复引用**
 - 方便对路径较深的文件 快速访问
 - 创建软件链接的命令：ln -s /home/xiowang/demo.py demo-s.py
 - 硬链接

- 相当于给 **同一片存储信息** 起了一个 新的名字，即使源文件丢失，数据任然不会丢失
- 创建硬链接时，硬连接数会加1
- 不可以给目录创建硬链接
- **同一片存储信息 有两个指针**
- 创建软件链接的命令: `ln demo.py demo-h.py`
- 文本搜索命令
 - `grep`
 - 命令可以混合使用
 - 显示行号 `-n`
 - 忽略大小写 `-i`
 - 取反 `-v`
 - 还可以结合正则表达式去使用
 - `^a` 以a开头
 - `k$` 以k结尾
 - `k.F` 中间的.可以代替 除换行符以外的任意字符
 - 结合管道去使用
 - `ls -lah | grep '内容'`
 - 结合文件使用
 - `grep 'K.g' demo.py -inv`
 - 搜索的内容可以去掉双引号或者单引号
- 查找文件命令
 - `find`
 - 在指定的目录下查找文件或者文件夹
 - 指定文件名或者文件的名字时: `find . -name '11.txt'`
 - 结合通配符去使用-----模糊匹配
 - `*` 匹配0个或多个字符
 - `?` 匹配任意一个字符
 - 注意:
 - 在使用通配符的时候不能使用单双引号
 - 拓展:
 - `rm,mv,cp` 也可以使用通配符
- 压缩和解压缩命令
 - 压缩
 - `tar`
 - 打包为 `tar.gz` 格式的:

- `tar -zcvf test.tar.gz *.txt`
- 打包为 tar.bz2 格式的:
 - `tar -jcvf test.tar.bs2 *.txt`
- zip
 - zip 指定目录 *.txt
 - 最简单, 但是也最浪费空间
 - 但是速度快, 适用于同事之间传输文件
- 解压
 - tar
 - tar.gz 格式的:
 - `tar -zxvf test.tar.gz -C 指定的目录`
 - tar.bz2格式的:
 - `tar -jxvf test.tar.bz2 -C 指定的目录`
 - 说明:
 - 在不指定目录的情况下 就是直接解压到当前目录下
 - unzip
 - unzip 需要解压的文件.zip -d 指定目录
 - 在不指定目录的情况下 就是直接解压到当前目录下
- 在使用 tar 或者 unzip 时的命令说明:
 - tar的:
 - -c 创建打包文件
 - -v 显示打包或者解包的详细信息
 - -f 指定文件名称 必须放在所有选项的后面
 - -z 指定打包或则解压的类型必须是 (.gz)
 - -j 指定打包或则解压的类型必须是 (.bz2)
 - -x 表示解包
 - -C 指定需要解压到哪个文件目录时使用的命令
 - unzip的: -d 指定需要解压到哪个文件目录时使用的命令
- 文件权限命令
 - 权限:
 - r 可读, 权限值是 4
 - w 可写, 权限值是 2
 - x 可执行, 权限值是 1
 - - 无权限, 权限值是 0
 - 权限对应的角色
 - u user, 表示该权限的所有者

- g group,表示用户组
- o other,表示其他用户
- a all,表示所有用户
- 修改文件权限的操作
 - + 增加权限
 - - 撤销权限
 - = 设置权限
- chmod
 - 字母法:
 - chmod u-r 1.txt
 - chmod a=rw 1.txt
 - chmod u=rwx,g=rx,o=rx demo.py
 - chmod u-w+x,g+w,o+w demo.py
 - 在文件第一行声明文需要执行的解释器的位置，就可以直接执行了
 - `#!/usr/local/bin/python3`
 - 在py文件中#号键的含义:
 - 注释: # 注释内容
 - 指定文件的编码格式: `#_*_coding:utf-8_*`
 - 指定python 解释器的路径: `#!/python解释器路径`
 - 数字法:
 - chmod 664 demo.py
 - chmod 755 demo.py
- 获取管理员权限的命令
 - 切换到root用户，获取管理员权限: `sudo -s`
 - 执行某一次命令时需要使用到管理员权限时使用: `sudo`
 - 获取ip地址: `ifconfig`
 - 查看现在有几个用户在线登录 `who`
 - 查看当前用户 `whoami`
 - 退出 `exit`
 - 修改密码 `passwd`
 - 查看某个命令的位置: `which 命令`
 - 立即关机: `shutdown -h now`
 - 重启: `reboot`
- 用户相关操作
 - 创建（添加）用户: `useradd`
 - 命令的选项

- -m 自动创建用户的主目录，主目录名就是用户名
 - -g 指定用户所属的用户组，默认不指定会自动创建一个同名的用户组
- 创建用户的时候需要使用管理员权限
 - `sudo useradd -m 用户名 : sudo useradd -m laowang`
- 查看用户
 - `id 用户名`
 - `cat /etc/passwd`
- 查看用户组
 - `cat /etc/group`
- 切换用户
 - `su - 用户名`
- 修改用户的信息
 - 修改用户密码
 - `sudo passwd 用户名`
 - 修改用户的权限----在附加组内就有该附加组的权限
 - `sudo usermod -G 权限名1, 权限名2 用户名 : sudo usermod -G sudo,adm laowang`
 - 追加附加组
 - `sudo gpasswd -a 用户名 附加组名`
 - 删除附加组
 - `sudo gpasswd -d 用户名 附加组名`
- 删除用户
 - `userdel`
 - 命令选项 `-r` 在删除用户的同时删除用户主目录
 - `sudo userdel -r 需要删除的用户的用户名 : sudo userdel -r laowang`
-
- 用户组相关操作
 - 在创建用户的时候，需要先创建用户组，在用户组的内部，再创建组内的用户
 - 创建一个用户组
 - `sudo groupadd 用户组名`
 - 这个时候创建用户时，就可以将用户分配到相应的组中：`sudo useradd -m -g 用户组名 用户名`
 - `sudo useradd -m -g test_xiao_zu xioHua`
 - 修改用户的主组
 - `sudo usermod -g 新的组名 用户名 : sudo usermod -g test_new_xiao_zu xioHua`

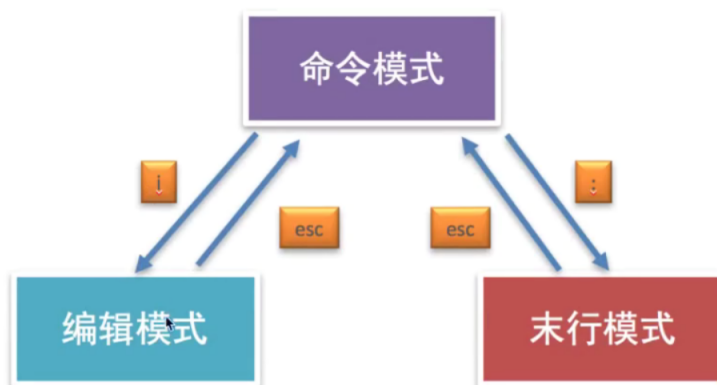
- 删除用户组
 - 注意: 只有当 没有用户使用 该组的时候, 才可以将这个组删除
 - 语法: `sudo groupdel 需要删除的组`
 - 例子: `sudo groupdel a_group`

• 命令3

- 远程登录和远程拷贝
- vim编辑器的介绍和使用

- vim的工作模式
 - 命令模式
 - 编辑模式
 - 末行模式
 - 工作模式效果图

工作模式效果图:



注意点:

编辑模式和末行模式之间不能直接进行切换, 都需要通过命令模式来完成。

- vim末行模式的命令

- `:w` 保存
- `:wq` 保存退出
- `:x` 保存退出
- `:q!` 强制退出

- vim常用的命令1

<code>yy</code>	复制光标所在行
<code>p</code>	粘貼
<code>dd</code>	删除/剪切当前行
<code>V</code>	按行选中
<code>u</code>	撤销
<code>ctr+r</code>	反撤销

- vim常用的命令2

>>	往右缩进
<<	往左缩进
/搜索的内容	搜索指定内容
:%s/要替换的内容/替换后的内容/g	全局替换
:开始行数,结束行数s/要替换的内容/替换后的内容	局部替换
.	重复上一次命令操作

- vim常用的命令3

G	回到最后一行
gg	回到第一行
数字+G	回到指定行
shift+6	回到当前行的行首
shift+4	回到当前行的行末
ctr+f	下一屏
ctr+b	上一屏

- 软件的安装和卸载

- 方式一

- 离线安装 deb文件格式
 - 安装 `sudo dpkg -i xxx.deb`
 - 卸载

- 方式二

- 在线安装
 - `sudo apt-get install 安装包`
 - 改镜像源
 - 可视化修改
 - 手动修改

以上内容整理于 [幕布文档](#)