

Linux 掌握的如何？

对 linux 系统目录结构有一定的了解, 在根目录下的 bin 目录中一般存放这系统的命令, dev 中存放这系统外部的一些设备, 网卡, 硬盘等。在 home 目录中会存放普通用户的目录。在 etc 目录中会存放一些软件的配置文件等等。

文件管理

掌握基本的文件操作。使用 touch, vim 创建一个文件。查看文件内容可以使用 cat, more, less, head, tail。文件内容较少可以使用 cat 输出全部内容。内容过多可以使用 less, more 分屏查看内容。head 可以看文件前几行。tail 也能动态的查看文件新增的内容用于日志追踪。

使用 grep 关键字搜索文本内容。例如在一个目录中筛选.log 结尾的文件使用 ls 目录加上管道符 grep 通配符 点 log。但是查找文件的话 find 命令更好用, 更具文件名查找-name, 类型-type, 拥有者-user。

修改文本内容可以使用 sed, 它能替换指定的内容, 也能在匹配行前或后添加内容, 删除匹配的行。还能使用正则表达式进行匹配。还可以使用 vim 修改文件内容, 在命令模式中使用 dd 删除光标所在行, yy 和 p 可以复制一行和粘贴, #G 可以跳转到多少行。进入插入模式就可以任意修改内容。ESC 回到命令模式: wq 就保存退出。

对文本内容截取可以使用 cut。可以, 以一个字符或多个字符进行分割, 获取需要的某列。awk 也能完成这样的功能的命令, 但它能做一些运算。比如统计目录大小, 先 cd 进入到目录中, 使用 ll 加上管道符, 再使用 awk, 单引号花括号中写上表达式 sum+=\$5;print sum。

用户管理

掌握用户和组管理, 会基本的用户创建, 删除: useradd, userdel。组的创建, 删除 groupadd, groupdel。设置用户和组的密码用 passwd 和 gpasswd。还有管理组中的成员, 用 gpasswd -a, 删除用 -d 参数。-M 重新设置组员。

权限管理

能够对文件的权限进行管理, 使用 chmod 分别为文件的拥有者, 组成员和其他用户设置可读可写或可执行的权限。例如为 shell 脚本添加可执行权限: chmod u+x 空格文件名。还有就是对整个目录及里面所有的文件添加权限可以使用 -R 参数。还有其它修改权限的命令 chown 更改文件的拥有者, chgrp 更改文件所属组。

进程管理

使用 ps 加 -ef 参数查看当前系统的进程。使用 -aux 参数查看所有用户执行的进程。还可以使用 top 命令实时的查看进程。更具查看到的进程的 PID, 可以使用 kill 命令结束进程。

服务管理

使用 systemctl 查看服务的状态, 开启和关闭服务。设置服务开机自启或关闭自启等。例如要查看 httpd 服务的状态: systemctl status httpd, 开启或关闭: systemctl start 或 stop httpd。设置开机自动或关闭自启动: systemctl enable 或 disable httpd。

网络管理

使用 `ifconfig` 或者 `ip addr` 查看本机的 ip 地址。使用 `ping` 命令查看是否与其他主机网络联通。使用 `netstat` 命令查看本机网络端口的使用情况：`netstat` 加 `-tunlp` 参数 管道符 `grep` “: 80” 查看使用 80 端口的进程和其他信息。也可以使用简单的命令 `lsof -l : 80` 查看。还有防火墙的开启关闭等。还有使用 `firewall-cmd` 命令加 `--add-port=80/tcp`，对 80 端口放行，这是临时的，需要永久的加上 `--permanent` 参数。然后重载防火墙 `--reload`。

磁盘管理

使用 `du -ah` 查看当前路径下每个文件的大小。`df -h` 查看整个系统磁盘使用的情况。还有当外部的硬盘连接时，在根目录下的 `dev` 中找到设备 `sda4` 或其他，使用 `mount` 命令将他挂在到根目录下的 `data` 文件夹，硬盘的数据就可以从这个目录访问。卸载硬盘使用 `umount` 即可。对磁盘进行分区可以使用 `fdisk`。

软件安装

还有软件的安装，软件安装有多种方式：使用 `yum` 安装，或者下载二进制的安装包，使用 `rpm` 加参数 `-ivh` 加文件名安装。还有就是下载软件的源码，自己进行编译，再安装软件。例如使用 `wget` 或 `curl` 下载源码压缩包，解压后可以修改安装路径，使用 `make` 命令进行编译，编译完成后使用 `make install` 命令安装软件。

Shell 脚本编写

定义一个用户变量：直接变量名等于变量值；使用 `set` 加管道符再使用 `grep` 搜索变量。`unset` 命令加上变量名可以清除变量；要创建系统变量，可以使用 `export` 命令加上变量名=变量值；查看系统变量使用的是 `env` 命令；清除系统变量使用的 `unset`；

在 shell 脚本中，可以直接使用一些特殊的变量：`$*`：接收运行脚本时传入的所有参数；`$#`：传输脚本参数的个数；`$?`：存储上一条命令执行是否成功的值；`$0`：当前脚本的进程 id；还有位置变量；脚本最大支持 9 个参数，分别用 `$1-$9` 接收。

在脚本中掌握流程控制语句的使用：`if` 条件判断，`case` 条件选择，循环 `for`，`while`，`until`。其中最重要的就是 `if` 条件判断。使用 `test` 命令做出判断；比如判断一个目录或文件是否存在，可以使用 `test -e 路径`；或者用中括号的形式表达：`[-e 路径]`；判断是否是文件或者目录，分别用 `-f` 和 `-d` 参数。还有数字判断：比如判断 `age` 变量是否大于 18：`[$age -gt 18]`；`lt` 判断是否小于，`-eq` 和 `-ne` 判断是否等于或者不等于。多条件的可以使用 `-a` 或 `-o` 连接，表示逻辑与和逻辑或；但一般使用 `&&` 和 `||` 更直观；

在脚本中也会进行重定向的操作；命令的结果可能需要保存下来；就可以在命令的后面加上重定向符 `>` 加上文件路径；也可以双 `>>` 追加内容到文件中；还有错误重定向符号，只负责把错误信息保存下来；比如某个命令后面加上 `> info.log 2> error.log`，分别把正常输出和错误输出保存到 `info.log` 和 `error.log` 中；

定时任务

掌握定时任务使用 `crontab` 命令，临时任务 `at`。查看定时任务 `crontab -l`，`-e` 参数进行任务编辑。任务分为六个部分，前五个都是时间，最后一个可以是一个 shell 脚本路径，也可以是一条命令。例如每天晚上 1 点执行一个脚本：0 1 * * * 再加上脚本的绝对路径。还有临时任务的使用，比如在今天 11: 40 要执行一个脚本，`at -f 脚本绝对路径 11: 40`。明天 11: 40 可以再加个 `tomorrow`。使用 `-l` 参数可以查看到已添加的临时任务，`-r` 参

数加上任务 id 可以删除定时任务。