Linux:

'pwd' 显示当前所在位置

'ls' 列出当前所在位置所有文件 -- list

'cd 文件名'进入文件夹 -- change directory 可以进入多级

'cd ..' 返回到上一级路径

'ls -l' 用列表列出，就是竖着放出来，大小是字节为单位

'ls -a' 显示所有文件，包括隐藏文件

ls -a -l' 可以一起用，反过来也行

ls -al' 和上面一样，可以一起用

-A 输出所有的文件，除了'.' 和 '..'

-l use a long listing format

-h, -human-readable

**创建 和移动**

'touch 文件名' 创建文件，不能创建文件夹

‘mkdir 文件夹名字’ 创建新目录，就是创建文件夹，只能一次一级

'rm 文件名字' 直接删除文件，清空所在内存，不会移废纸篓 默认只能移除文件

\*\*\* 创建隐藏文件: ’ 文件.后缀'就是以'.'开头的文件都是隐藏文件

‘rm -d 文件夹名字' 给rm加个描述就能移除文件夹了, 但是必须是空文件夹

‘rm -r 文件夹名字' 可以直接移除有东西的文件夹 -r是recursive的意思，就是把文件夹里的全移走

'mv 文件/文件夹名字 文件夹名字‘ 相当于剪切, 可以移动文件夹

'mv 文件名 ./新的文件名' 改名，就是把文件移动到当前目录同时改名

’cp 文件 文件夹‘ 拷贝 文件可以多级， 文件夹可以用'.'和'..'

'.'当前目录

'..'父目录

'cp -r 文件夹 文件夹' -r是描述用的，把文件改成文件夹了

'cp 文件名 ./新的文件名' 复制到同目录，同时改名

'cp 文件名 ../新的文件名' 复制到父目录，同时改名

'which 命令' 查看命令所在位置

压缩和解压缩

‘tar 选项 文件’ 常用 -cvf 和 -xvf

'tar -cvf tar包名 待压缩文件或目录‘ ---压缩

‘tar -tvf tar包名’ ---列出tar包中文件列表

’tar -xvf tar包名' ---解压包中文件到当前目录

'tar -xvf tar包名 -C 指定解压目录' ---解压包中文件到当前目录

-c 创建压缩文件

-x 解开压缩文件

-t 列出压缩包中的文件列表

-v 显示命令的执行过程

-f 目标文件名

文件

搜索地址

'find 地址 -name 文件名字‘ 地址到尽可能精确 文件名字要完全一致，过程中 ctrl + C 终止程序

”find 地址 -name ‘字符和\*以及？’“ 用正则符号来模糊搜索，但是要把名字用单引号

查看

'cat 文件名' 显示全部文件

’more 文件名' 分页显示文件，按 空格 下一页，按 b 上一页，或者用方向键

'cat -b 文件名' 显示全部文件，每一行前面标行号,但是空行不算行号

'cat -n 文件名’ 同样，但是空行也算行号

搜索文件

'grep 关键词 文件名’ 和command+f一样，但是不知道哪一行

‘grep -n 关键词 文件名’ 同样，显示哪一行

'grep -v 关键词 文件名' 显示不包含关键词的行，包括空行，可以和-n一起用

'grep -i 关键词 文件名' 忽略大小写抓取

/etc中放配置文件，用其中的services文件举例，其中#是注释

sunzhuofan@MacBook-Pro /etc % grep -v '#' services 不要注释的，但是不在开头的注释也整行忽略了

#是特殊字符要用‘’

sunzhuofan@MacBook-Pro /etc % grep -v ^'#' services 就是不要用#开头的

sunzhuofan@MacBook-Pro /etc % grep -v $'#' services 就是不要用#结尾的

编辑

'echo 字符' 是把字符打印到终端

'echo 字符 > 文件名字' 用字符覆盖文件

'echo 字符 >> 文件名字' 用字符追加文件结尾这两个操作当前目录没有这个文件会创建新的

大于号主要用于追加或者覆盖文件，命令结果都可以

比如说sunzhuofan@MacBook-Pro sample % ls . >> test.txt就是把当前目录的所有文件名追加到文件里

管道

就是将一个操作的输出结果用一个管道接到下一个操作的输入,用 | 链接

sunzhuofan@MacBook-Pro sample % ls -l | grep test3.txt

python3 --version 得到python的版本

文件统计

'wc 文件名' 统计文件的行数, 单词数, 字符数, 空格和换行都算一个字符

'wc -c 文件名' 统计bytes数量, 字节

'wc -m 文件名'统计字符数量

'wc -l 文件名'统计行数

'wc -w 文件名'统计单词数量

例如，当从我的电脑上复制名为myfile.txt的文件到lab 机器上home目录里的291文件夹：

scp myfile.txt basabaa@ohaton.cs.ualberta.ca:291

回来：

scp basabaa@ohaton.cs.ualberta.ca:291/myfile.txt . 要在本地机器执行

运行C

gcc -Wall -std=c99 main.c -o main

ssh zhuofan3@ug10.cs.ualberta.ca 201

ssh zhuofan3@ug05.cs.ualberta.ca 291 ---1740983 % 35 = 13cd

ctrl + D 退出远程连接（优雅～）

1、从本地将文件传输到服务器

scp【本地文件的路径】【服务器用户名】@【服务器地址】：【服务器上存放文件的路径】

2、从本地将文件夹传输到服务器

scp -r【本地文件的路径】【服务器用户名】@【服务器地址】：【服务器上存放文件的路径】

3、将服务器上的文件传输到本地

scp 【服务器用户名】@【服务器地址】：【服务器上存放文件的路径】【本地文件的路径】

4、将服务器上的文件夹传输到本地

scp -r 【服务器用户名】@【服务器地址】：【服务器上存放文件的路径】【本地文件的路径】

vim

删除命令：

普通模式：

x删除单一字符

0 将光标定位到行首的位置

$ 将光瞄定位到行末位置

gg 将光标定位到文件开头

G 将光标定位到文件末尾

dd 删除光标所在正行

u 撤销

ctrl+r 撤销掉撤销

git改英文：

直接输入alias查看现有别名

git='LANG=en\_GB git'

/usr/local/bin/git

gitee 克隆：username是用户名下面的@。。。

编辑～/.ssh/config 文件能省去编写ssh 后面地址的时间

格式：

Host 【任意名字】//这个任意名字代替了下面的地址

Hostname 【主机名字】

User 【登陆的用户名】

"

Host lab

Hostname ug13.ca.ualberta.ca

User zhuofan3

"

这个host让 zhuofan3@ug13.cs.ualberta.ca 变成了 lab

公钥public key在本地：./Users/sunzhuofan/.ssh

生成公钥：在.ssh目录： ssh-keygen -t rsa -t是加密，rsa是加密格式

ssh-keygen -t ed25519 -f ~/.ssh/ualberta\_cs\_labs -N

这个是CMPUT 201 给出的生成公钥的格式，同样是在本地，ed25519似乎是加密格式

-f -N不知道干什么，剩下的是改文件名，不改应该就是id\_加密格式

,pub结尾的是公钥，没有的是私钥，私钥不可以分享。

lab 桌面地址： /cshome/zhuofan3

通配符

'\* 字符' 星号指任意个字符，但是星号后面必须以指定字符结尾，说白了就是找出所有以指定字符结尾

'?' 指代一个字符： ‘？？.txt'就是指代所有文件名是两个字符接着.txt

make\_makefile

make工程管理器：

makefile Makefile --make先执行小写makefile

小项目gcc编译的快大项目的话要是发现需要改一个文件就可能会需要整个工程重新编译会很慢，所以需要工程管理器

a.out -> main.o tool1.o tool2.o

main.o -> main.c

tool1.o -> tool1.c

tool2.o -> tool2.c

vim makefile

mytool:main.o tool1.o tool2.o

gcc main.o tool1.o tool2.o -o mytool

main.o:main.c

gcc main.c -c -Wall -o main.o

tool1.o:tool1.c

gcc tool1.c -c -Wall -o tool1.o

tool2.o:tool2.c

gcc tool2.c -c -Wall -o tool2.o

clean:

rm -f \*.o mytool core

///////////////////////////////////////////////

demo:helloworld.c

gcc -Wall -std=c99 helloworld.c -o helloworld

clean:

-rm -f \*.o helloworld core

然后就生成了一个可执行文件 mytool

vim helloworld.sh

#!/bin/bash

echo "Hello world"

mkdir newfolder

:wq

chmod +x helloworld.sh

chmod 设置模式 文件名

模式：操作 权限

操作：

：增加权限

-：减少权限

=：设定权限

权限：

w：写

r：读

x：excutable可执行，没权限进不去