**/** root directory; **~** home directory

**Useful Command**

**mkdir** dir1 dir2 “dir 3”

**rmdir** dir//only work when dir is empty

**rm** -r(recursive) -f(force, 不需要确认) lab or -rf

**pwd//**prints the name of the current directory

**mv** file “newfile”//rename file;

**mv** dir1 dir2//move dir1 to dir2

**ls** -l = **ll**//List dir in long content **e.g.:**

-rwxr-xr-x 2 user group 4096 May 12 12:40 filename

**ls** -a// show hidden file(name begin with “.”)

**touch** file//create file or update its timestamp

**cat** file//输出文件内容

**cp** file1 file2// copy

**cp** -r(递归) dir1 dir2//同上,但是包括文件夹内部所有文件

**chmod** [who(u,g,o,a)][operator(+-=)][permission] file

4=r, 2=w, 1=x

permission: user, group, other;read, write, execute

**grep** “abc” file//返回带有abc的行

**grep** -E //启用扩展**正则表达式**, 以下为部分正则:

**.**//任意单字符; ^//行开始; $//行结束;

?//前一个字符0or1次; +//~>0次; \*//~>=0次

()//group; |//or; [1-5]//匹配1到5中的数字字符

pattern {n}(匹配出现n次); {n,m}(n-m次, 可取边界 )

**cut** -d ',' -f2 file//以”,”为分隔, 提取每行第二块文本

**diff** file1 file2// 比对file1file2不同

‘> eee’:插入eee ; ‘< ccc’:更改为ccc

**wc** file//统计行数, words和bytes,用-l仅显示行数**e.g.**

1(lines) 14(words) 514(bytes) file

**sort** file//按字母序排序文件

-n:数字排序;-r:倒序;-kn:对第n fields排序;-t:设置分隔符

**uniq** file//删除相邻的重复行

**spell** file//显示incorrect的词

**su**//superuser mode

**find** path -name “filename” -type(-f:只搜file;-d:只搜dir)

**standard I/O**

0-stdin; 1-stdout; 2-error out

><: redirection(覆盖模式) >><<(追加模式)

|: pipe, 传输数据到指令

**bash scrip:**

变量赋值 a=”114”; 使用变量 $a \后加特殊字符

$[#a]//变量长度; ${a:pos:len}//子字符串;

${a/old/new}替换子字符串

“`cat file` 111”, 单引号内

if [] then elif then else fi;

for ? in ? do done; while [] do done;

**Condition**:

**str**: [“$a”] a不为空=[-n “$a”]=-[-z “$a”]; 比较符:==!=

\>\<//按字典序比较字符串; 在[[ ]]中时无需”\”转义

**file**: -e//exist; -f//is file; -d//为文件夹; -s//size>0;

-r//readable; -w-x同理; **logic**:-a=&&, -o=||, !:Not

条件为**指令**, 则当成功执行时为真

**整数**: [$a operation $b]; -eq==, ne!=, gt>, lt<, ge>=, le...

**数学运算**前加**let**: **let** "b = $a + 9"

**MakeFile:**

**$^**//所有依赖项(去重) **$<**//第一个依赖项 **$@**//当前项

**$+**//所有依赖项(保留重复); **$?** 所有比目标新的依赖项

**$\***//模式匹配的内容（用于模式规则）

a.o: a.cpp a.h

g++ -c $^

main: a.o, b.o...

g++ $^ -o main

clean:

rm -f \*.o main

.PHONY: clean(以上为dddsx所写)

**C++ File I/O**

**fstream**

**文本, 信件

AI 生成的内容可能不正确。**

filename//文件名称; mode//打开文件的模式，

- `std::ios::**in**`：以输入模式打开文件。

- `std::ios::**out**`：以输出模式打开文件。

- `std::ios::**app**`: 追加模式打开文件。

- `std::ios::**ate**`: 打开文件并定位到文件末尾。

- `std::ios::**trunc**`: 打开文件并截断文件, 即清空文件内容

**sstream**

- `**istringstream**`: 用于从字符串中读取数据。

- `**ostringstream**`: 用于将数据写入字符串。

- `**stringstream**`: 是上述两种组合, 可以同时读取和写入

文本, 信件

AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

**iomanip** 库中的函数通常与 << 和 >> 操作符一起使用，以实现对输出流的控制。

图形用户界面, 日历

AI 生成的内容可能不正确。

**Vector**

图形用户界面

AI 生成的内容可能不正确。

**List**

图形用户界面

AI 生成的内容可能不正确。

**Map**

图形用户界面, 文本

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。