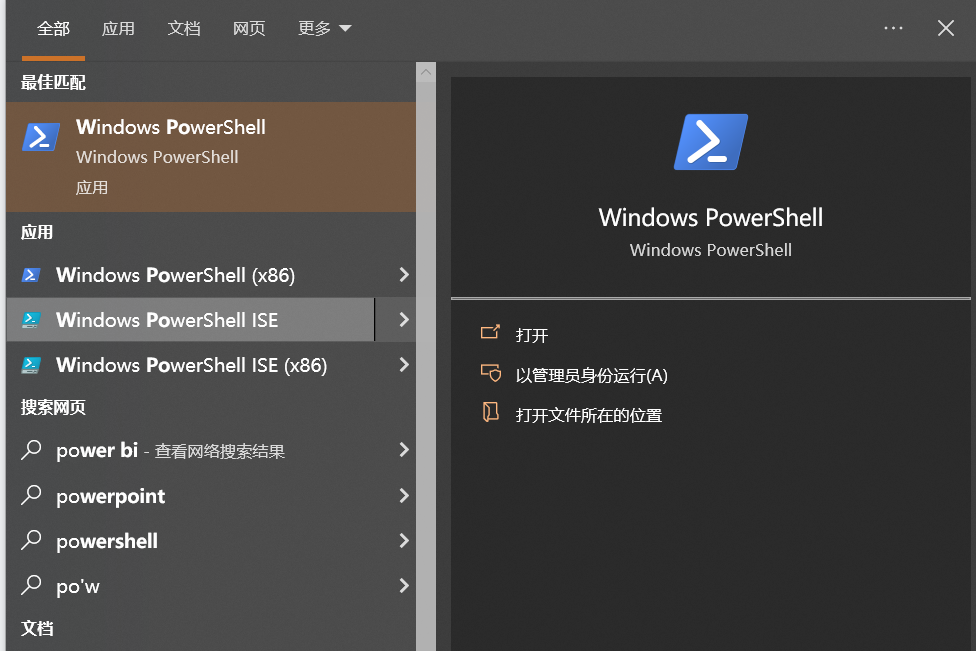
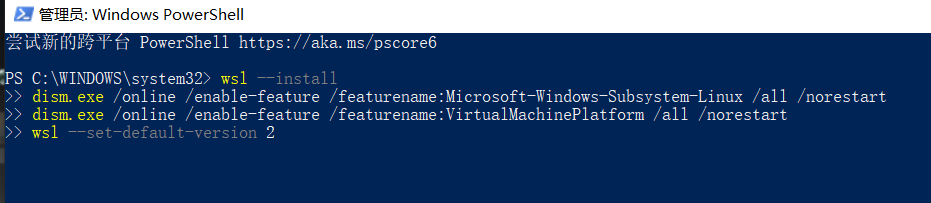
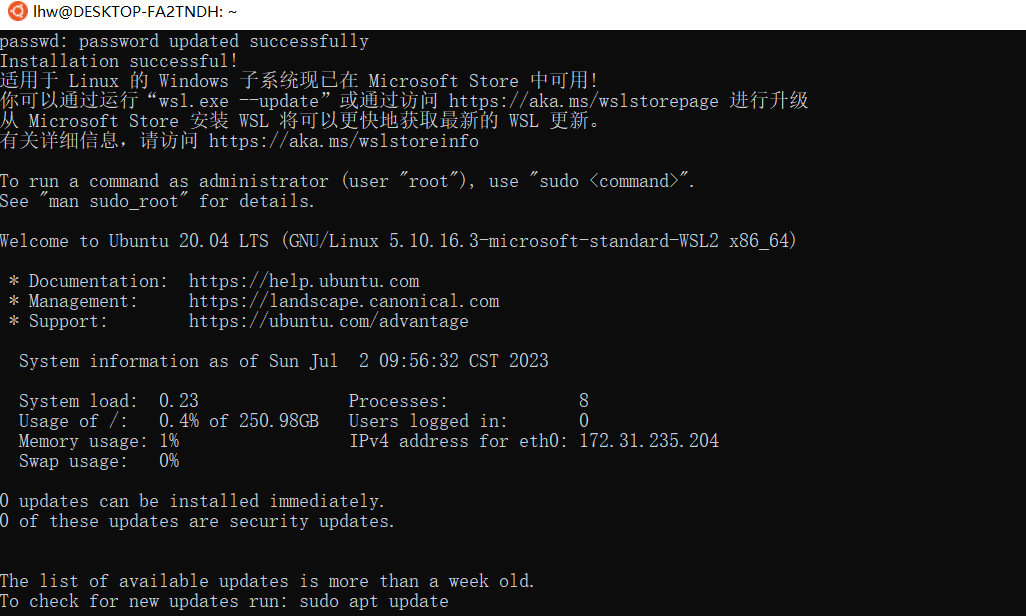
打开PowerShell



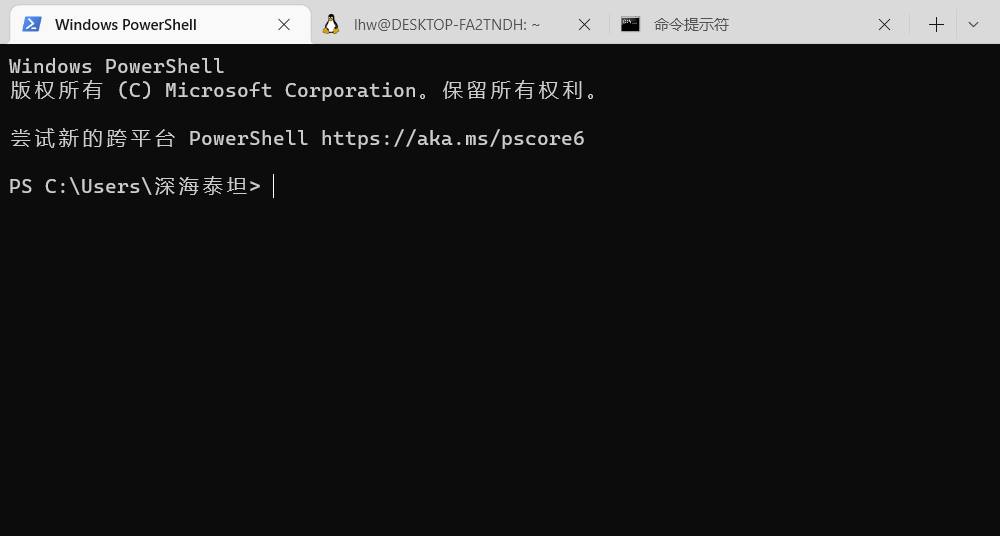
输入安装wsl命令



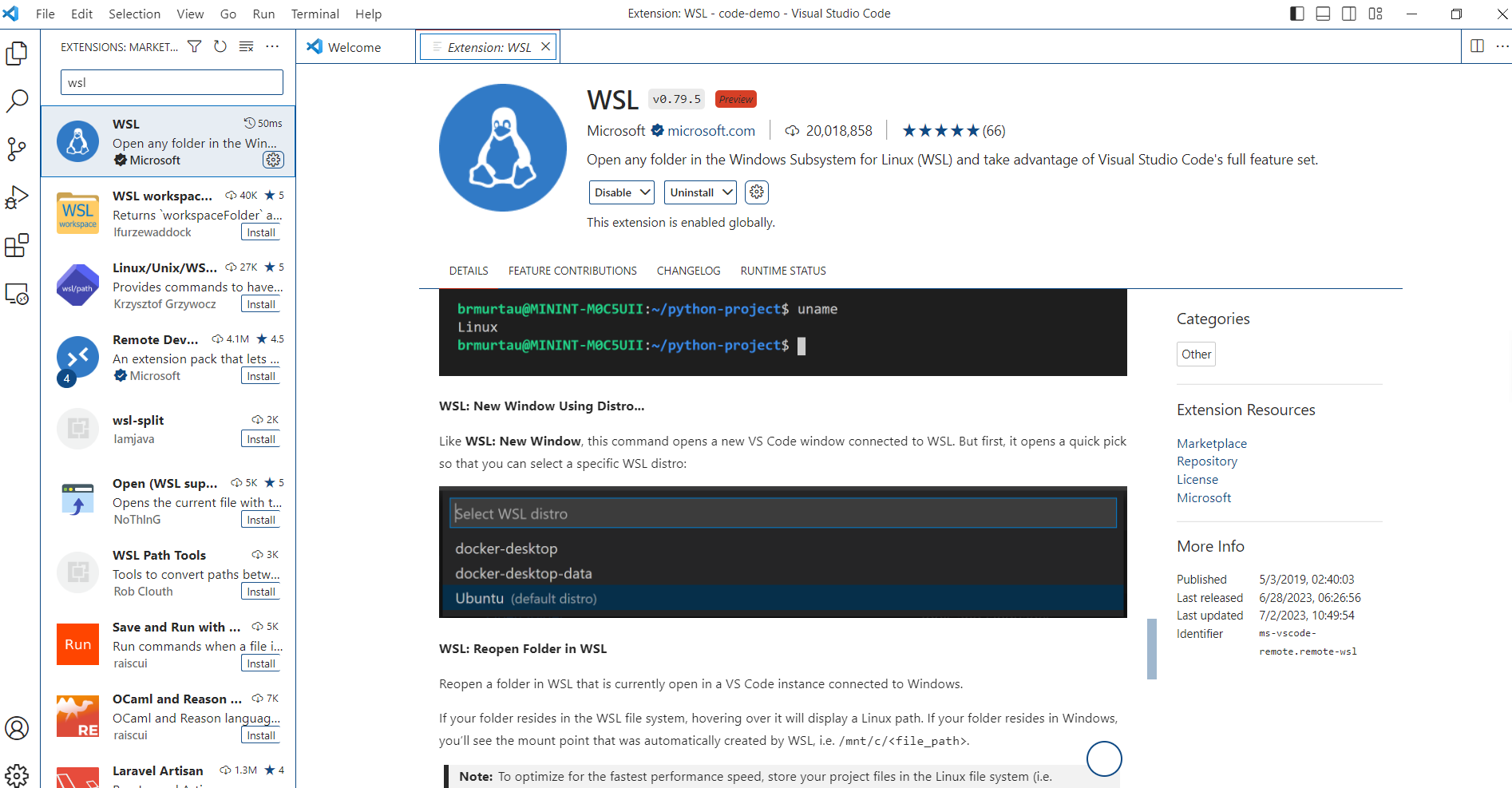
安装Ubuntu



下载安装Windows Terminal

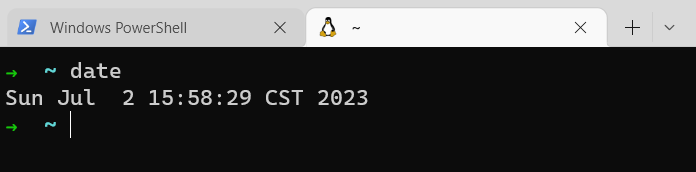


下载vscoade并且在里面安装wsl

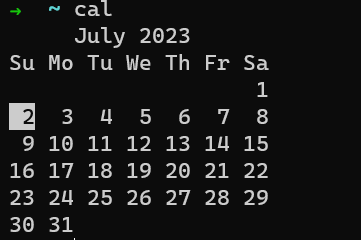


Linux命令

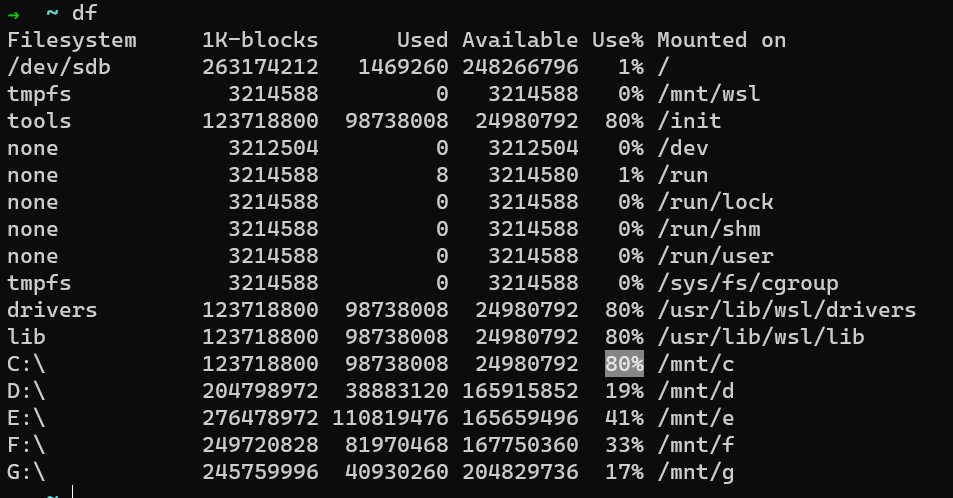
date 显示系统当前时间和日期



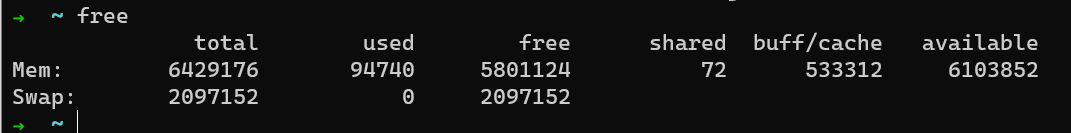
cal 显示当前月份的日历



df 查看磁盘剩余空间的数量



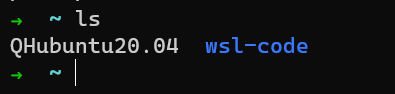
free 显示空闲内存的数量



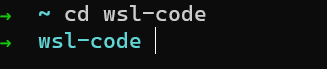
pwd 打印出当前工作目录名



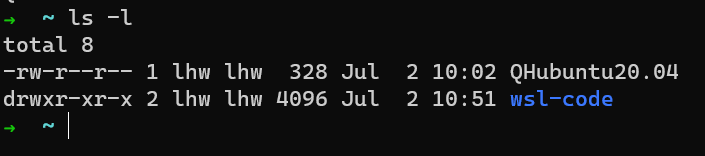
ls 列出目录内容



cd “文件名” 跳转到指定文件中



Ls -l 列出目录内容并且更加详细



-rw-r--r-- 对于文件的访问权限。第一个字符指明文件类型。在不同类型之间， 开头的“－”说明是一个普通文件，“d”表明是一个目录。其后三个字符是文件所有者的 访问权限，再其后的三个字符是文件所属组中成员的访问权限，最后三个字符是其他所 有人的访问权限。

1 文件的硬链接数目。参考随后讨论的关于链接的内容。

lhw 文件属主的用户名。

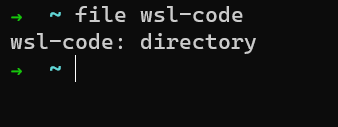
lhw 文件所属用户组的名字。

328 以字节数表示的文件大小。

Jul 2 10:02 上次修改文件的时间和日期。

QHubuntu20.04 文件名

file filename 打印出文件内容的简单描述



\* 匹配任意多个字符（包括零个或一个）

? 匹配任意一个字符（不包括零个）

[characters] 匹配任意一个属于字符集中的字符

[!characters] 匹配任意一个不是字符集中的字符

[[:class:]] 匹配任意一个属于指定字符类中的字符

[:alnum:] 匹配任意一个字母或数字

[:alpha:] 匹配任意一个字母

[:digit:] 匹配任意一个数字

[:lower:] 匹配任意一个小写字母

[:upper] 匹配任意一个大写字母

\* 所有文件

g\* 文件名以“g”开头的文件

b\*.txt 以"b"开头，中间有零个或任意多个字符，并以".txt"结尾的文件

Data??? 以“Data”开头，其后紧接着3个字符的文件

[abc]\* 文件名以"a","b",或"c"开头的文件

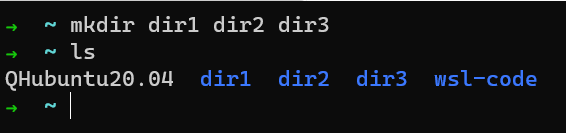
BACKUP.[0-9][0-9][0-9] 以"BACKUP."开头，并紧接着3个数字的文件

[[:upper:]]\* 以大写字母开头的文件

[![:digit:]]\* 不以数字开头的文件

\*[[:lower:]123] 文件名以小写字母结尾，或以 “1”，“2”，或 “3” 结尾的文件

mkdir dir1 dir2 dir3 创建三个目录dir1 dir2 dir3



cp item... directory

-a, --archive 复制文件和目录，以及它们的属性，包括所有权和权限。 通常，复本具有用户所操作文件的默认属性。

-i, --interactive 在重写已存在文件之前，提示用户确认。如果这个选项不指定， cp 命令会默认重写文件。

-r, --recursive 递归地复制目录及目录中的内容。当复制目录时， 需要这个选项（或者-a 选项）。

-u, --update 当把文件从一个目录复制到另一个目录时，仅复制 目标目录中不存在的文件，或者是文件内容新于目标目录中已经存在的文件。

-v, --verbose 显示翔实的命令操作信息

cp file1 file2 复制文件 file1 内容到文件 file2。如果 file2 已经存在， file2 的内容会被 file1 的内容重写。如果 file2 不存在，则会创建 file2。

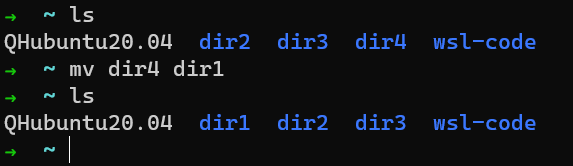
cp -i file1 file2 这条命令和上面的命令一样，除了如果文件 file2 存在的话，在文件 file2 被重写之前， 会提示用户确认信息。

cp file1 file2 dir1 复制文件 file1 和文件 file2 到目录 dir1。目录 dir1 必须存在。

cp dir1/\* dir2 使用一个通配符，在目录 dir1 中的所有文件都被复制到目录 dir2 中。 dir2 必须已经存在。

cp -r dir1 dir2 复制目录 dir1 中的内容到目录 dir2。如果目录 dir2 不存在， 创建目录 dir2，操作完成后，目录 dir2 中的内容和 dir1 中的一样。 如果目录 dir2 存在，则目录 dir1 (和目录中的内容)将会被复制到 dir2 中。

mv dir4 dir1 将dir4改名为dir1



-i --interactive 在重写一个已经存在的文件之前，提示用户确认信息。 如果不指定这个选项，mv 命令会默认重写文件内容。

-u --update 当把文件从一个目录移动另一个目录时，只是移动不存在的文件， 或者文件内容新于目标目录相对应文件的内容。

-v --verbose 当操作 mv 命令时，显示翔实的操作信息。

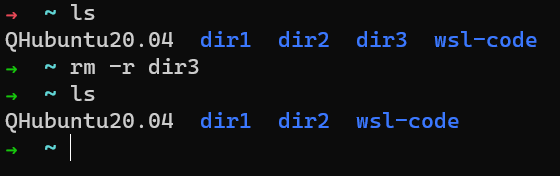
mv file1 file2 移动 file1 到 file2。如果 file2 存在，它的内容会被 file1 的内容重写。 如果 file2 不存在，则创建 file2。 每种情况下，file1 不再存在。

mv -i file1 file2 除了如果 file2 存在的话，在 file2 被重写之前，用户会得到 提示信息外，这个和上面的选项一样。

mv file1 file2 dir1 移动 file1 和 file2 到目录 dir1 中。dir1 必须已经存在。

mv dir1 dir2 如果目录 dir2 不存在，创建目录 dir2，并且移动目录 dir1 的内容到 目录 dir2 中，同时删除目录 dir1。如果目录 dir2 存在，移动目录 dir1（及它的内容）到目录 dir2。

rm dir3 删除文件



-i, --interactive 在删除已存在的文件前，提示用户确认信息。 如果不指定这个选项，rm 会默默地删除文件

-r, --recursive 递归地删除文件，这意味着，如果要删除一个目录，而此目录 又包含子目录，那么子目录也会被删除。要删除一个目录，必须指定这个选项。

-f, --force 忽视不存在的文件，不显示提示信息。这选项颠覆了“--interactive”选项。

-v, --verbose 在执行 rm 命令时，显示翔实的操作信息。

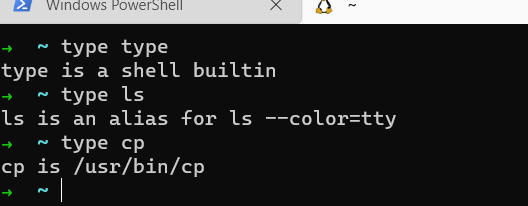
rm file1 默默地删除文件

rm -i file1 除了在删除文件之前，提示用户确认信息之外，和上面的命令作用一样。

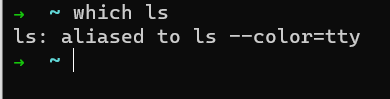
rm -r file1 dir1 删除文件 file1, 目录 dir1，及 dir1 中的内容。

rm -rf file1 dir1 同上，除了如果文件 file1，或目录 dir1 不存在的话，rm 仍会继续执行。

type command 显示命令的类别，给出一个特定的命令名（做为参数）。



which 显示一个可执行程序的位置



help 得到 shell 内部命令的帮助文档

cat 连接文件

sort 排序文本行

uniq 报道或省略重复行

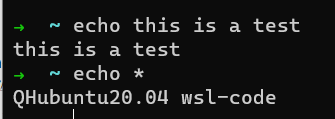
grep 打印匹配行

wc 打印文件中换行符，字，和字节个数

head 输出文件第一部分

tail 输出文件最后一部分

echo 显示一行文本



转义字符 有时候我们只想引用单个字符。我们可以在字符之前加上一个反斜杠，在这个上下文中叫做转义字符。 经常在双引号中使用转义字符，来有选择地阻止展开。

clear 清空屏幕

history 显示历史列表内容

Ctrl-a 移动光标到行首。

Ctrl-e 移动光标到行尾。

Ctrl-f 光标前移一个字符；和右箭头作用一样。

Ctrl-b 光标后移一个字符；和左箭头作用一样。

Alt-f 光标前移一个字。

Alt-b 光标后移一个字。

Ctrl-l 清空屏幕，移动光标到左上角。clear 命令完成同样的工作。

Ctrl-d 删除光标位置的字符。

Ctrl-t 光标位置的字符和光标前面的字符互换位置。

Alt-t 光标位置的字和其前面的字互换位置。

Alt-l 把从光标位置到字尾的字符转换成小写字母。

Alt-u 把从光标位置到字尾的字符转换成大写字母。

Ctrl-k 剪切从光标位置到行尾的文本。

Ctrl-u 剪切从光标位置到行首的文本。

Alt-d 剪切从光标位置到词尾的文本。

Alt-Backspace 剪切从光标位置到词头的文本。如果光标在一个单词的开头，剪切前一个单词。

Ctrl-y 把剪切环中的文本粘贴到光标位置。

自动补全shell 能帮助你的另一种方式是通过一种叫做自动补全的机制。当你敲入一个命令时，按下tab 键，自动补全就会发生。

Ctrl-p 移动到上一个历史条目。类似于上箭头按键。

Ctrl-n 移动到下一个历史条目。类似于下箭头按键。

Alt-< 移动到历史列表开头。

Alt-> 移动到历史列表结尾，即当前命令行。

Ctrl-r 反向递增搜索。从当前命令行开始，向上递增搜索。

Alt-p 反向搜索，不是递增顺序。输入要查找的字符串，然后按下Enter，执行搜索。

Alt-n 向前搜索，非递增顺序。

Ctrl-o 执行历史列表中的当前项，并移到下一个。如果你想要执行历史列表中一系列的命令，这很方便。

!! 重复最后一次执行的命令。可能按下上箭头按键和enter键更容易些。

!number 重复历史列表中第number行的命令。

!string 重复最近历史列表中，以这个字符串开头的命令。

!?string 重复最近历史列表中，包含这个字符串的命令。

id 显示用户身份号

chmod 更改文件模式

umask 设置默认的文件权限

su 以另一个用户的身份来运行shell

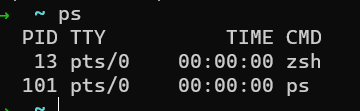
sudo 以另一个用户的身份来执行命令

chown 更改文件所有者

chgrp 更改文件组所有权

passwd 更改用户密码

ps 查看进程



R 运行。这意味着，进程正在运行或准备运行。

S 正在睡眠。进程没有运行，而是，正在等待一个事件，比如说，一个按键或者网络数据包。

D 不可中断睡眠。进程正在等待I/O，比方说，一个磁盘驱动器的I/O。

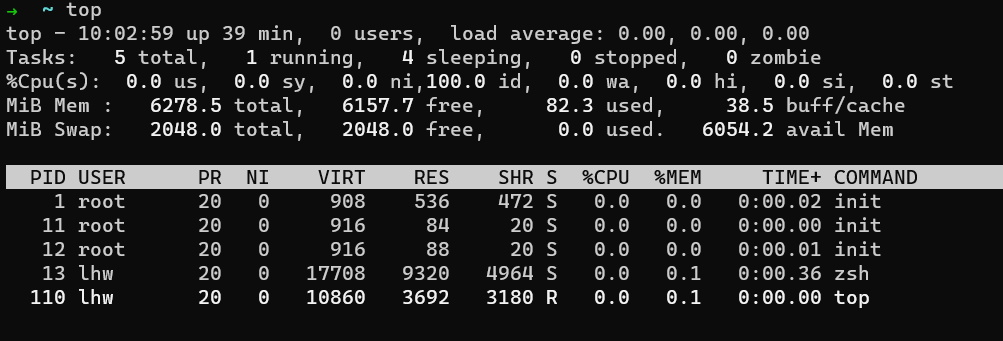
T 已停止。已经指示进程停止运行。稍后介绍更多。

Z 一个死进程或“僵尸”进程。这是一个已经终止的子进程，但是它的父进程还没有清空它。 （父进程没有把子进程从进程表中删除）

< 一个高优先级进程。这可能会授予一个进程更多重要的资源，给它更多的 CPU 时间。 进程的这种属性叫做niceness。具有高优先级的进程据说是不好的（less nice），因为它占用了比较多的CPU时间，这样就给其它进程留下很少时间。

N 低优先级进程。一个低优先级进程（一个“好”进程）只有当其高优先级进程执行之后，才会得到处理器时间。

top 动态查看进程



jobs 列出活跃的任务

bg 把一个任务放到后台执行

fg 把一个任务放到前台执行

kill 给一个进程发送信号

killall 杀死指定名字的进程

shutdown 关机或重启系统

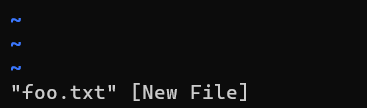
printenv | less检查环境变量

set 设置shell 选项

export 导出环境变量，让随后执行的程序知道。

alias 创建命令别名

vi foo.txt



l or右箭头 向右移动一个字符

h or 左箭头 向左移动一个字符

j or下箭头 向下移动一行

k or 上箭头 向上移动一行

0 (零按键) 移动到当前行的行首。

^ 移动到当前行的第一个非空字符。

$ 移动到当前行的末尾。

w 移动到下一个单词或标点符号的开头。

W 移动到下一个单词的开头，忽略标点符号。

b 移动到上一个单词或标点符号的开头。

B 移动到上一个单词的开头，忽略标点符号。

Ctrl-f or Page Down向下翻一页

Ctrl-b or Page Up 向上翻一页

numberG 移动到第number行。例如，1G移动到文件的第一行。

G 移动到文件末尾。

Shell 提示符中用到的转义字符

\a 以 ASCII 格式编码的铃声 . 当遇到这个转义序列时，计算机会发出嗡嗡的响声。

\d 以日，月，天格式来表示当前日期。例如，“Mon May 26.”

\h 本地机的主机名，但不带末尾的域名。

\H 完整的主机名。

\j 运行在当前 shell 会话中的工作数。

\l 当前终端设备名。

\n 一个换行符。

\r 一个回车符。

\s shell 程序名。

\t 以24小时制，hours:minutes:seconds 的格式表示当前时间.

\T 以12小时制表示当前时间。

@ 以12小时制，AM/PM 格式来表示当前时间。

\A 以24小时制，hours:minutes 格式表示当前时间。

\u 当前用户名。

\v shell 程序的版本号。

\V Version and release numbers of the shell.

\w 当前工作目录名。

\W 当前工作目录名的最后部分。

! 当前命令的历史号。

# 当前 shell 会话中的命令数。

$ 这会显示一个"$"字符，除非你拥有超级用户权限。在那种情况下， 它会显示一个"#"字符。

[ 标志着一系列一个或多个非打印字符的开始。这被用来嵌入非打印 的控制字符，这些字符以某种方式来操作终端仿真器，比方说移动光标或者是更改文本颜色。

] 标志着非打印字符序列结束。

mount挂载一个文件系统

umount 卸载一个文件系统

fsck 检查和修复一个文件系统

fdisk分区表控制器

mkfs 创建文件系统

fdformat 格式化一张软盘

dd 把面向块的数据直接写入设备

genisoimage (mkisofs)创建一个ISO 9660的映像文件

wodim (cdrecord) 把数据写入光存储媒介

md5sum计算MD5检验码

ping 发送ICMP ECHO\_REQUEST 软件包到网络主机

traceroute 打印到一台网络主机的路由数据包

netstat 打印网络连接，路由表，接口统计数据，伪装连接，和多路广播成员

ftp 因特网文件传输程序

wget 非交互式网络下载器

ssh OpenSSH SSH 客户端（远程登录程序）

locate 查找文件的简单方法

find 查找文件的复杂方式

find 文件类型

b块设备文件

c字符设备文件

d目录

f普通文件

l符号链接

find 大小单位

b 512 个字节块。如果没有指定单位，则这是默认值。

c 字节

w两个字节的字

k千字节(1024个字节单位)

M兆字节(1048576个字节单位)

G千兆字节(1073741824个字节单位)

find 测试条件

-cmin n 匹配的文件和目录的内容或属性最后修改时间正好在 n 分钟之前。 指定少于 n 分钟之前，使用 -n，指定多于 n 分钟之前，使用 +n。

-cnewer file 匹配的文件和目录的内容或属性最后修改时间早于那些文件。

-ctime n 匹配的文件和目录的内容和属性最后修改时间在 n\*24小时之前。

-empty 匹配空文件和目录。

-group name 匹配的文件和目录属于一个组。组可以用组名或组 ID 来表示。

-iname pattern 就像-name 测试条件，但是不区分大小写。

-inum n 匹配的文件的 inode 号是 n。这对于找到某个特殊 inode 的所有硬链接很有帮助。

-mmin n 匹配的文件或目录的内容被修改于 n 分钟之前。

-mtime n 匹配的文件或目录的内容被修改于 n\*24小时之前。

-name pattern 用指定的通配符模式匹配的文件和目录。

-newer file 匹配的文件和目录的内容早于指定的文件。当编写 shell 脚本，做文件备份时，非常有帮助。 每次你制作一个备份，更新文件（比如说日志），然后使用 find 命令来决定自从上次更新，哪一个文件已经更改了。

-nouser 匹配的文件和目录不属于一个有效用户。这可以用来查找 属于删除帐户的文件或监测攻击行为。

-nogroup 匹配的文件和目录不属于一个有效的组。

-perm mode 匹配的文件和目录的权限已经设置为指定的 mode。mode 可以用 八进制或符号表示法。

-samefile name 相似于-inum 测试条件。匹配和文件 name 享有同样 inode 号的文件。

-size n 匹配的文件大小为 n。

-type c 匹配的文件类型是 c。

-user name 匹配的文件或目录属于某个用户。这个用户可以通过用户名或用户 ID 来表示。

find 命令的逻辑操作符

-and 如果操作符两边的测试条件都是真，则匹配。可以简写为 -a。 注意若没有使用操作符，则默认使用 -and。

-or 若操作符两边的任一个测试条件为真，则匹配。可以简写为 -o。

-not 若操作符后面的测试条件是真，则匹配。可以简写为一个感叹号（!）。

() 把测试条件和操作符组合起来形成更大的表达式。这用来控制逻辑计算的优先级。 默认情况下，find 命令按照从左到右的顺序计算。经常有必要重写默认的求值顺序，以得到期望的结果。 即使没有必要，有时候包括组合起来的字符，对提高命令的可读性是很有帮助的。注意 因为圆括号字符对于 shell 来说有特殊含义，所以在命令行中使用它们的时候，它们必须 用引号引起来，才能作为实参传递给 find 命令。通常反斜杠字符被用来转义圆括号字符。

gzip 压缩或者展开文件

-c 把输出写入到标准输出，并且保留原始文件。也有可能用--stdout 和--to-stdout 选项来指定。

-d 解压缩。正如gunzip 命令一样。也可以用--decompress 或者--uncompress 选项来指定.

-f 强制压缩，即使原始文件的压缩文件已经存在了，也要执行。也可以用--force 选项来指定。

-h 显示用法信息。也可用--help 选项来指定。

-l 列出每个被压缩文件的压缩数据。也可用--list 选项。

-r 若命令的一个或多个参数是目录，则递归地压缩目录中的文件。也可用--recursive 选项来指定。

-t 测试压缩文件的完整性。也可用--test 选项来指定。

-v 显示压缩过程中的信息。也可用--verbose 选项来指定。

-number 设置压缩指数。number 是一个在1（最快，最小压缩）到9（最慢，最大压缩）之间的整数。 数值1和9也可以各自用--fast 和--best 选项来表示。默认值是整数6。

bzip2 块排序文件压缩器

-c 把输出写入到标准输出，并且保留原始文件。也有可能用--stdout 和--to-stdout 选项来指定。

-d 解压缩。正如gunzip 命令一样。也可以用--decompress 或者--uncompress 选项来指定.

-f 强制压缩，即使原始文件的压缩文件已经存在了，也要执行。也可以用--force 选项来指定。

-h 显示用法信息。也可用--help 选项来指定。

-l 列出每个被压缩文件的压缩数据。也可用--list 选项。

-r 若命令的一个或多个参数是目录，则递归地压缩目录中的文件。也可用--recursive 选项来指定。

-t 测试压缩文件的完整性。也可用--test 选项来指定。

-v 显示压缩过程中的信息。也可用--verbose 选项来指定。

-number 设置压缩指数。number 是一个在1（最快，最小压缩）到9（最慢，最大压缩）之间的整数。 数值1和9也可以各自用--fast 和--best 选项来表示。默认值是整数6。

tar 磁带打包工具

c 为文件和／或目录列表创建归档文件。

x 抽取归档文件。

r 追加具体的路径到归档文件的末尾。

t 列出归档文件的内容。

zip 打包和压缩文件

rsync 同步远端文件和目录

grep 选项

-i 忽略大小写。不会区分大小写字符。也可用--ignore-case 来指定。

-v 不匹配。通常，grep 程序会打印包含匹配项的文本行。这个选项导致 grep 程序 只会不包含匹配项的文本行。也可用--invert-match 来指定。

-c 打印匹配的数量（或者是不匹配的数目，若指定了-v 选项），而不是文本行本身。 也可用--count 选项来指定。

-l 打印包含匹配项的文件名，而不是文本行本身，也可用--files-with-matches 选项来指定。

-L 相似于-l 选项，但是只是打印不包含匹配项的文件名。也可用--files-without-match 来指定。

-n 在每个匹配行之前打印出其位于文件中的相应行号。也可用--line-number 选项来指定。

-h 应用于多文件搜索，不输出文件名。也可用--no-filename 选项来指定。

POSIX 字符集

[:alnum:] 字母数字字符。在 ASCII 中，等价于：[A-Za-z0-9]

[:word:] 与[:alnum:]相同, 但增加了下划线字符。

[:alpha:] 字母字符。在 ASCII 中，等价于：[A-Za-z]

[:blank:] 包含空格和 tab 字符。

[:cntrl:] ASCII 的控制码。包含了0到31，和127的 ASCII 字符。

[:digit:] 数字0到9

[:graph:] 可视字符。在 ASCII 中，它包含33到126的字符。

[:lower:] 小写字母。

[:punct:] 标点符号字符。在 ASCII 中，等价于：

[:print:] 可打印的字符。在[:graph:]中的所有字符，再加上空格字符。

[:space:] 空白字符，包括空格，tab，回车，换行，vertical tab, 和 form feed.在 ASCII 中， 等价于：[ \t\r\n\v\f]

[:upper:] 大写字母。

[:xdigit:] 用来表示十六进制数字的字符。在 ASCII 中，等价于：[0-9A-Fa-f]

cat 连接文件并且打印到标准输出

sort 给文本行排序

uniq 报告或者省略重复行

cut 从每行中删除文本区域

paste 合并文件文本行

join 基于某个共享字段来联合两个文件的文本行

comm 逐行比较两个有序的文件

diff 逐行比较文件

patch 给原始文件打补丁

tr 翻译或删除字符

sed 用于筛选和转换文本的流编辑器

aspell 交互式拼写检查器

nl 添加行号

常用 nl 选项

-b style 把 body 按被要求方式数行，可以是以下方式：a = 数所有行 t = 数非空行。这是默认设置。n = 无 pregexp = 只数那些匹配了正则表达式的行

-f style 将 footer 按被要求设置数。默认是无

-h style 将 header 按被要求设置数。默认是

-i number 将页面增加量设置为数字。默认是一。

-n format 设置数数的格式，格式可以是：ln = 左偏，没有前导零。rn = 右偏，没有前导零。rz = 右偏，有前导零。

-p 不要在没一个逻辑页面的开始重设页面数。

-s string 在没一个行的末尾加字符作分割符号。默认是单个的 tab。

-v number 将每一个逻辑页面的第一行设置成数字。默认是一。

-w width 将行数的宽度设置，默认是六。

fold 限制文件列宽

fmt 一个简单的文本格式转换器

pr 让文本为打印做好准备

printf 格式化数据并打印出来

groff 一个文件格式系统

pr 转换需要打印的文本文件

+first[:last] 输出从 first 到 last（默认为最后）范围内的页面。

-columns 根据 columns 指定的列数排版页面内容。

-a 默认多列输出为垂直，用 -a (across)可使其水平输出。

-d 双空格输出。

-D format 用 format 指定的格式修改页眉中显示的日期，日期命令中 format 字符串的描述详见参考手册。

-f 改用换页替换默认的回车来分割页面。

-h header 在页眉中部用 header 参数替换打印文件的名字。

-l length 设置页长为 length，默认为66行（每英寸6行的美国信纸）。

-n 输出行号。

-o offset 创建一个宽 offset 字符的左页边。

-w width 设置页宽为 width，默认为72字符。

lpr 打印文件

lp 打印文件（System V）

-d printer 设定目标（打印机）为 printer。若d 选项未指定，则使用系统默认打印机。

-n number 设定的打印份数为 number。

-o landscape 设置输出为横向。

-o fitplot 缩放文件以适应页面。打印图像时较为有用，如 JPEG 文件。

-o scaling=number 缩放文件至 number。100表示填满页面，小于100表示缩小，大于100则会打印在多页上。

-o cpi=number 设定输出为 number 字符每英寸。默认为10。

-o lpi=number 设定输出为 number 行每英寸，默认为6。

-o page-bottom=points

a2ps 为 PostScript 打印机格式化文件

--center-title text 设置中心页标题为 text。

--columns number 将所有页面排列成 number 列。默认为2。

--footer text 设置页脚为 text。

--guess 报告参数中文件的类型。由于 a2ps 会转换并格式化所有类型的数据，所以当给定文件类型后，这个选项可以很好的用来判断 a2ps 应该做什么。

--left-footer text 设置左页脚为 text。

--left-title text 设置页面左标题为 text。

--line-numbers=interval 每隔 interval 行输出行号。

--list=defauls 显示默认设置。

--list=topic 显示 topic 设置，topic 表示下列之一：代理程序（用来转换数据的外部程序），编码，特征，变量，媒介（页面大小等），ppd（PostScript 打印机描述信息），打印机，起始程序（为常规输出添加前缀的代码部分），样式表，或用户选项。

--pages range 打印 range 范围内的页面。

--right-footer text 设置右页脚为 text。

--right-title text 设置页面右标题为 text。

--rows number 将所有页面排列成 number 排。默认为1。

-B 没有页眉。

-b text 设置页眉为 text。

-f size 使用字体大小为 size 号。

-l number 设置每行字符数为 number。此项和 -L 选项（见下方）可以给文件用其他程序来更准确的分页，如 pr。

-L number 设置每页行数为 number。

-M name 使用打印媒介的名称——例如，A4。

-n number 每页输出 number 份。

-o file 输出到文件 file。如果指定为 - ，则输出到标准输出。

-P printer 使用打印机 printer。如果未指定，则使用系统默认打印机。

-R 纵向打印。

-r 横向打印。

-T number 设置制表位为每 number 字符。

-u text 用 text 作为页面底图（水印）。

lpstat 显示打印机状态信息

-a [printer...] 显示 printer 打印机的队列。这里显示的状态是打印机队列承受任务的能力，而不是实体打印机的状态。若未指定打印机，则显示所有打印队列。

-d 显示系统默认打印机的名称。

-p [printer...] 显示 printer 指定的打印机的状态。若未指定打印机，则显示所有打印机状态。

-r 显示打印系统的状态。

-s 显示汇总状态。

-t 显示完整状态报告。

lpq 显示打印机队列状态

lprm 取消打印任务

cancel 取消打印任务（System V）

测试文件表达式

file1 -ef file2 file1 和 file2 拥有相同的索引号（通过硬链接两个文件名指向相同的文件）。

file1 -nt file2 file1新于 file2。

file1 -ot file2 file1早于 file2。

-b file file 存在并且是一个块（设备）文件。

-c file file 存在并且是一个字符（设备）文件。

-d file file 存在并且是一个目录。

-e file file 存在。

-f file file 存在并且是一个普通文件。

-g file file 存在并且设置了组 ID。

-G file file 存在并且由有效组 ID 拥有。

-k file file 存在并且设置了它的“sticky bit”。

-L file file 存在并且是一个符号链接。

-O file file 存在并且由有效用户 ID 拥有。

-p file file 存在并且是一个命名管道。

-r file file 存在并且可读（有效用户有可读权限）。

-s file file 存在且其长度大于零。

-S file file 存在且是一个网络 socket。

-t fd fd 是一个定向到终端／从终端定向的文件描述符 。 这可以被用来决定是否重定向了标准输入／输出错误。

-u file file 存在并且设置了 setuid 位。

-w file file 存在并且可写（有效用户拥有可写权限）。

-x file file 存在并且可执行（有效用户有执行／搜索权限）。

#!/bin/bash

# sys\_info\_page: program to output a system information page

PROGNAME=$(basename $0)

TITLE="System Information Report For $HOSTNAME"

CURRENT\_TIME=$(date +"%x %r %Z")

TIMESTAMP="Generated $CURRENT\_TIME, by $USER"

report\_uptime () {

cat <<- \_EOF\_

<H2>System Uptime</H2>

<PRE>$(uptime)</PRE>

\_EOF\_

return

}

report\_disk\_space () {

cat <<- \_EOF\_

<H2>Disk Space Utilization</H2>

<PRE>$(df -h)</PRE>

\_EOF\_

return

}

report\_home\_space () {

if [[ $(id -u) -eq 0 ]]; then

cat <<- \_EOF\_

<H2>Home Space Utilization (All Users)</H2>

<PRE>$(du -sh /home/\*)</PRE>

\_EOF\_

else

cat <<- \_EOF\_

<H2>Home Space Utilization ($USER)</H2>

<PRE>$(du -sh $HOME)</PRE>

\_EOF\_

fi

return

}

usage () {

echo "$PROGNAME: usage: $PROGNAME [-f file | -i]"

return

}

write\_html\_page () {

cat <<- \_EOF\_

<HTML>

<HEAD>

<TITLE>$TITLE</TITLE>

</HEAD>

<BODY>

<H1>$TITLE</H1>

<P>$TIMESTAMP</P>

$(report\_uptime)

$(report\_disk\_space)

$(report\_home\_space)

</BODY>

</HTML>

\_EOF\_

return

}

# process command line options

interactive=

filename=

while [[ -n $1 ]]; do

case $1 in

-f | --file) shift

filename=$1

;;

-i | --interactive) interactive=1

;;

-h | --help) usage

exit

;;

\*) usage >&2

exit 1

;;

esac

shift

done

# interactive mode

if [[ -n $interactive ]]; then

while true; do

read -p "Enter name of output file: " filename

if [[ -e $filename ]]; then

read -p "'$filename' exists. Overwrite? [y/n/q] > "

case $REPLY in

Y|y) break

;;

Q|q) echo "Program terminated."

exit

;;

\*) continue

;;

esac

fi

done

fi

# output html page

if [[ -n $filename ]]; then

if touch $filename && [[ -f $filename ]]; then

write\_html\_page > $filename

else

echo "$PROGNAME: Cannot write file '$filename'" >&2

exit 1

fi

else

write\_html\_page

fi

联数组必须用带有 -A 选项的 declare 命令创建。在这个脚本中我们创建了如下五个数组：

files 包含了目录中文件的名字，按文件名索引

file\_group 包含了每个文件的组所有者，按文件名索引

file\_owner 包含了每个文件的所有者，按文件名索引

groups 包含了属于索引的组的文件数目

owners 包含了属于索引的所有者的文件数目

行7-10：查看是否一个有效的目录名作为位置参数传递给程序。如果不是，就会显示一条使用信息，并且脚本退出，退出状态为1。

行12-20：循环遍历目录中的所有文件。使用 stat 命令，行13和行14抽取文件所有者和组所有者， 并把值赋给它们各自的数组（行16，17），使用文件名作为数组索引。同样地，文件名自身也赋值给 files 数组。

行18-19：属于文件所有者和组所有者的文件总数各自加1。

行22-27：输出文件列表。为做到这一点，使用了 “${array[@]}” 参数展开，展开成整个的数组元素列表， 并且每个元素被当做是一个单独的词。从而允许文件名包含空格的情况。也要注意到整个循环是包裹在花括号中， 从而形成了一个组命令。这样就允许整个循环输出会被管道输送给 sort 命令的输入。这是必要的，因为 展开的数组元素是无序的。

行29-40：这两个循环与文件列表循环相似，除了它们使用 “${!array[@]}” 展开，展开成数组索引的列表而不是数组元素的。