

grep

- 指定一个格式，将文件中符合格式的行打印出来

按文件内容进行搜索

某一行中有部分满足正则表达式。


\$grep 选项 格式 文件名.
正则表达式

grep命令选项

- F 忽略控制字符, 按字面值匹配
- c 只显示次数.
- i 匹配内容的时候, 忽略大小写.
- n. 显示行号

grep命令的文本格式

正则表达式: 字符串的生成规则



| | | |
|---|---|---|
| } | a | ✓ |
| | b | ✓ |
| | c | ✗ |

gray ✓
grey ✓
gry x

- 字符串内容和规则一致
- 字符串根据规则进行匹配

• 匹配任意一个字符. 相当于通配符的问号

$[a-z]$ 匹配 $a-z$ 任意一个字符.

正则表达式

- 1 { \cdot $[]$ 匹配字符
 $a, b, c, \dots, 0, 1, \dots; A, \dots Z$ 匹配字符
(字符串) } 基本单位.
- 2 { $?$ 基本单位? \Rightarrow 基本单位出现0或1次
 $*$ 基本单位* \Rightarrow 基本单位出现0~多次
 \wedge shift + 6 $^$ 基本单位 \Rightarrow 基本单位出现在行首
 $\$$ shift + 4 基本单位\$ \Rightarrow 基本单位出现在行尾

"bo?k"

b k
b o k

grep和find的同时使用

先用find找文件.

再用grep找内容

```
$ find . -name "*.ch" | xargs grep -nE "main"
```

其他命令

\$ alias 别名

alias cp='rm -rf'

\$ history 历史命令记录

\$ man 帮助, 1~9 手册
-k -f

一次输入多个命令 分号

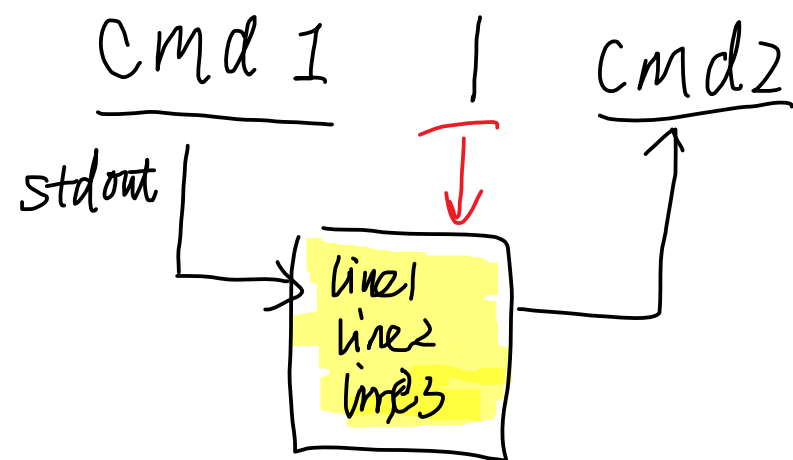
cmd1 ; cmd2

先执行cmd1, 再执行cmd2.

管道

man sh

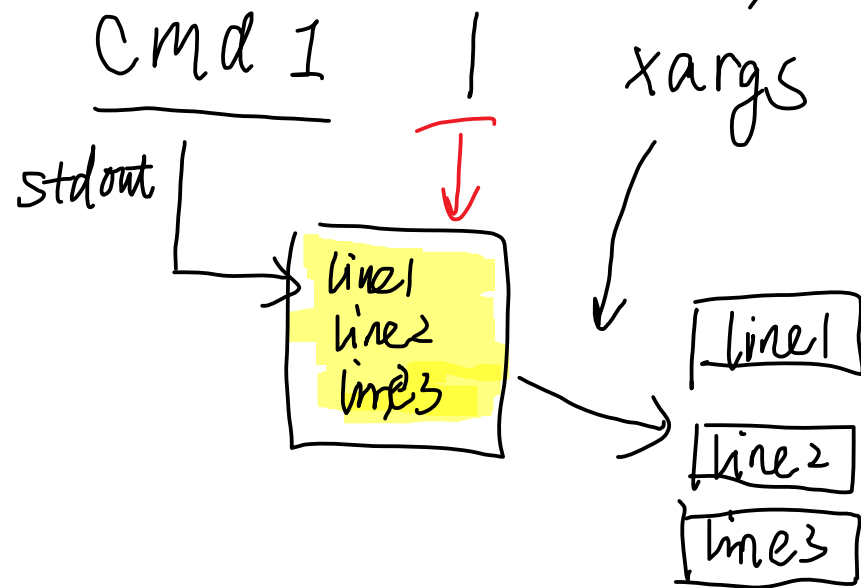
/pipeline.



(不需要文件号).
history | tail

xargs

| xargs



1. 将管道内容按行分割

2. 循环将每一行作为 cmd2 的最后一个参数。

```
$ find . -name "*.c" | xargs grep -nE "main"
```

Cmd2

file1.c
file2.c
file3.c

while((

{
 cmd2 line1~3

}

grep -nE "main" file1.c

grep -nE "main" file2.c

grep -nE "main" file3.c

命令替换

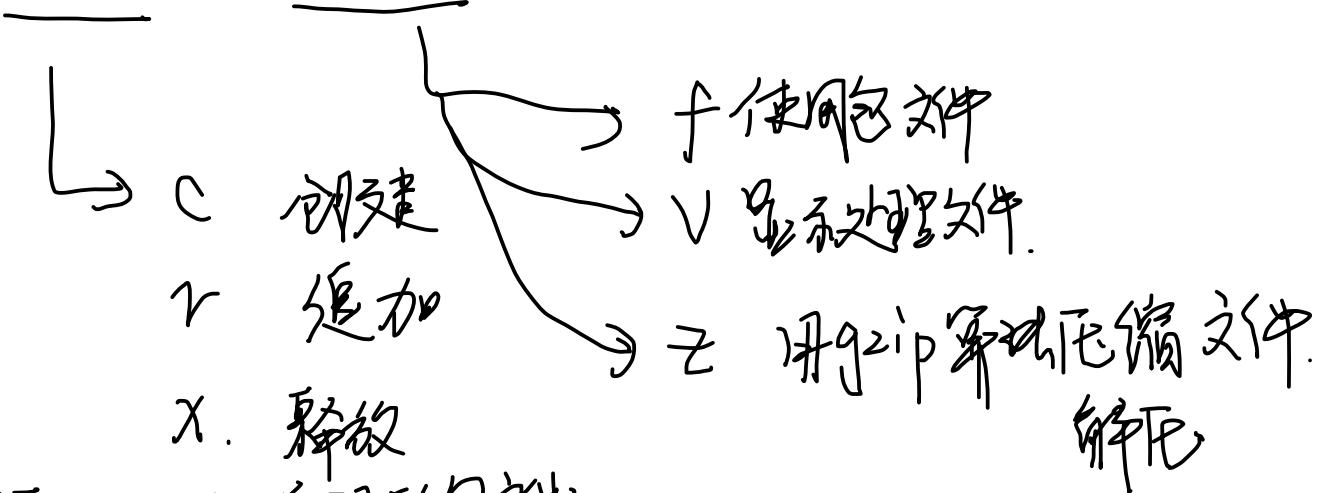
\$ cmd1 cmd2
↑
ESC 下面 1 左键.

\$ cmd1 \$(cmd2)

先运行 cmd2. 将结果作为 cmd1 的参数

打包和压缩

\$tar [主选项+辅助选项] 目标文件 源文件/目录



```
graph TD; A["[主选项+辅助选项]"] --> B["c 创建"]; A --> C["r 追加"]; A --> D["x 移除"]; A --> E["f 使用包文件"]; A --> F["v 显示处理文件"]; A --> G["z 用gzip算法压缩文件"]; A --> H["解压缩"];
```

主选项分支：
c 创建
r 追加
x 移除

辅助选项分支：
f 使用包文件
v 显示处理文件
z 用gzip算法压缩文件
解压缩

打包会先分配一个较大空间的包文件

bzip2

压缩文件后缀 -bz2

-d 解压

-v 显示文件.

scp命令

上传/下载.

scp -r 路径 路径
- 源 目标

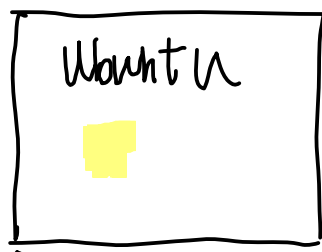
远程主机的路径:

用户名@IP地址:
 路径名.

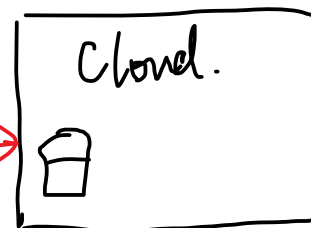
39.96.42.90
test test

密钥管理

ssh-keygen



scp



cd ~/.ssh 没有目录就创建一个
ssh-keygen 按多次回车 得到一对密钥
1 使用scp将公钥上传到远程服务器
登录远程服务器，然后把公钥进行追加
cat id_rsa.pub >> .ssh/authorized_keys
2 将公钥内容复制到gitee的个人设置里面
要重新克隆 选择ssh协议

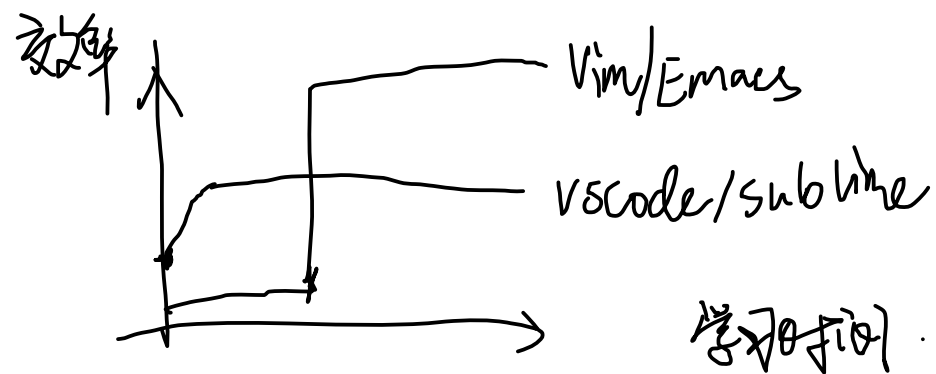
vim文本编辑器

完全脱离鼠标

国内使用C++ Tencent Baidu

Vim

Emacs Nano



vim

\$vim 文件名.

a 后面

A 行尾

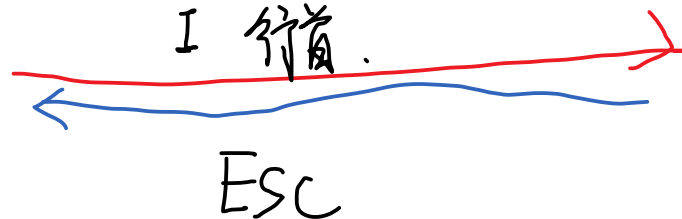
i 光标前插文本

I 行首.

append.

普通模式

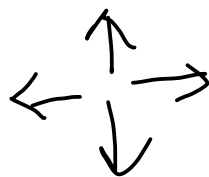
输入命令



编辑模式

光标移动

h k
j l



删除和修改

x 删除字符.

dd 删除一整行

[n]dd 删除n行

~~D~~/d\$ 删除尾部

d^ 删除头部

dw 删除单词 (word)

d[n]w 删除n个单词.

删除等价于剪切

p 粘贴

u. 撤销

ctrl+r 撤销撤销

yy 复制一行

yank

[n]yy.

光标快速移动

^ 行首

\$ 行尾

ctrl+d down

ctrl+u up

ctrl+f forward

ctrl+b backward

gg 回到文件头

G 回到文件尾

H 页首

L 最后一行的行首

w 光标往后移动一个单词

b 向前移动

[n]+向前n行

[n]-向前n行

: [n] [n]G 去第n行

查找与替换

`/[str]` 查找

n 下一个匹配结果

N 上一个匹配结果

支持正则表达式的.

`:s /[str] /[dst]`

只对单行生效, 替换一次

`:s /[str] /[dst] /g`

全部替换

i 忽略大小写.

`:x,y s/[str] /[dst] /ig`

$x \sim y$

`:% s/[str] /[dst] /ig`

可视化

V 横选

ctrl+v 选择若干行 → I → "//" → ESC

其他操作

: help

: %! xxd 文本 → 二进制

: %! xxd -r 回来

: w (written) 不要按 ctrl+s ctrl+g 解除

: q

: q!

{ : wq
: x

vim外命令

```
vim +200 The_Holy_Bible.txt
```

```
vim +/God The_Holy_Bible.txt
```

多窗口

: new + 文件名
,
: vnew
,
: split
,
: vsplit
,

Ctrt + W, W

多标签 : tabnew .

gt 下↑

gT 上↑

代码对齐

=

① visual 途中再代码对齐

② $gg = C_1$

.vimrc文件

- .vimrc文件放在用户家目录