

文件

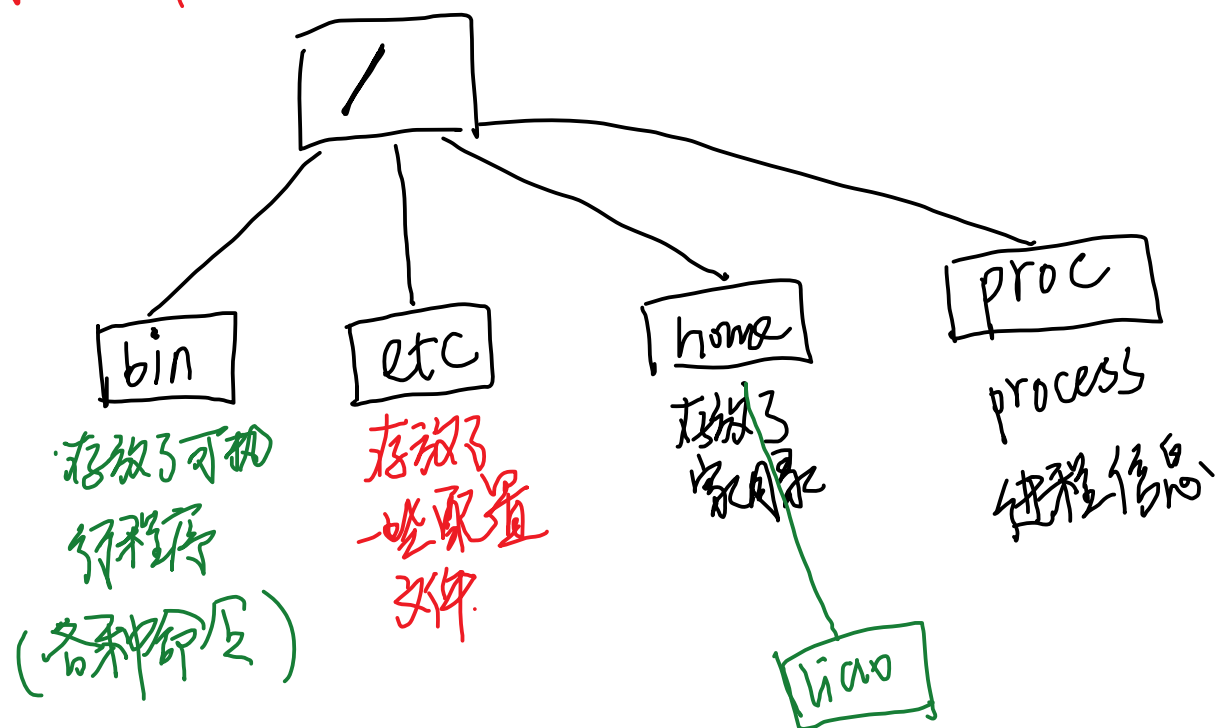
- 文件是存储在磁盘上数据集合

文件系统（虚拟文件系统 VFS）

树形结构

根结点

目录是非叶子结点



文件路径

绝对路径

从根出发

找到文件

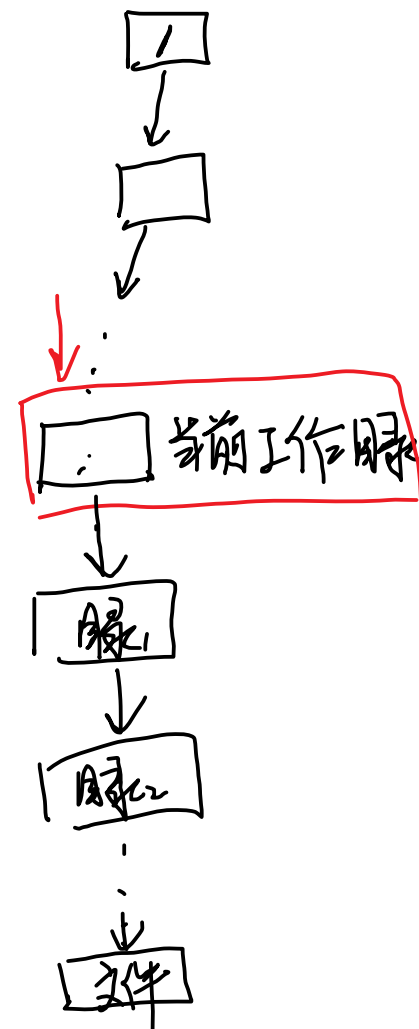
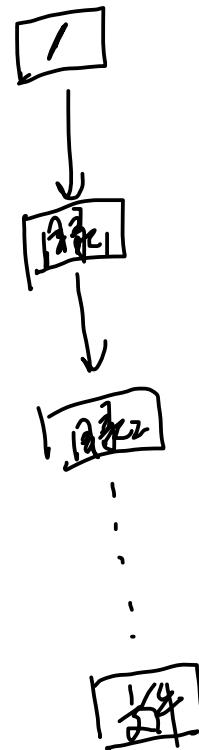
目录层次

/ 目录₁ / 目录₂ / ... / 文件

相对路径

目录₁ / 目录₂ / ... / 文件

从当前工作目录



当前工作目录

shell命令的当前位置

\$pwd (prints working directory)

切换当前工作目录

\$ cd.

(change directory)



当前用户的家目录

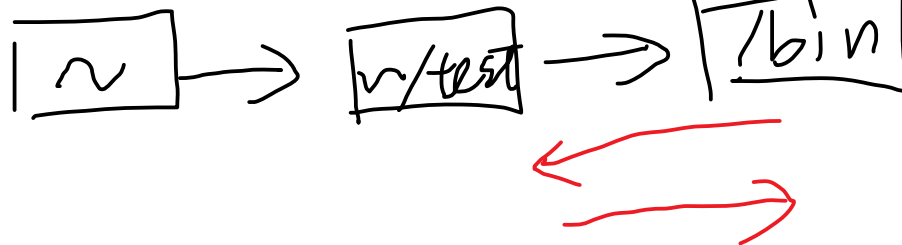
/home / ~~用户名~~
家目录.

\$ cd . 保持位置不变

\$ cd .. 回到父目录

\$ cd & \$ cd ~ 回到家目录

\$ cd - 回到上一次的目录 { 每次使用cd
的时候会保存上一次的
路径



创建目录

\$ mkdir 创建.

\$ rmdir 删除 只能删除空目录

显示目录下面的文件

\$ls 显示当前工作目录的文件

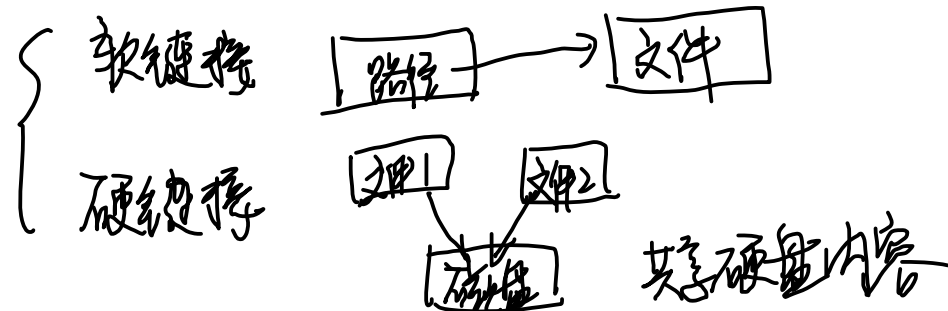
命令 --help

\$ls 路径名

\$ls -a all 显示隐藏文件 以.开头的文件.

\$ls -l 显示详细信息

文件的详细信息



```
drwxrwxr-x 3 liao liao 4096 Apr 7 10:16 dir1
-r--r--r-- 1 liao liao 4 Apr 6 23:39 file1
--w--w--w- 1 liao liao 4 Apr 6 23:40 file2
```

文件类型	d 目录 - 普通文件	c 字符设备文件	b 块设备文件	l 软链接
用户权限	3位二进制 → 8进制 r 读 w 写 x 执行	IO 以字节	IO 以块为单位	快捷方式

3个
8进制

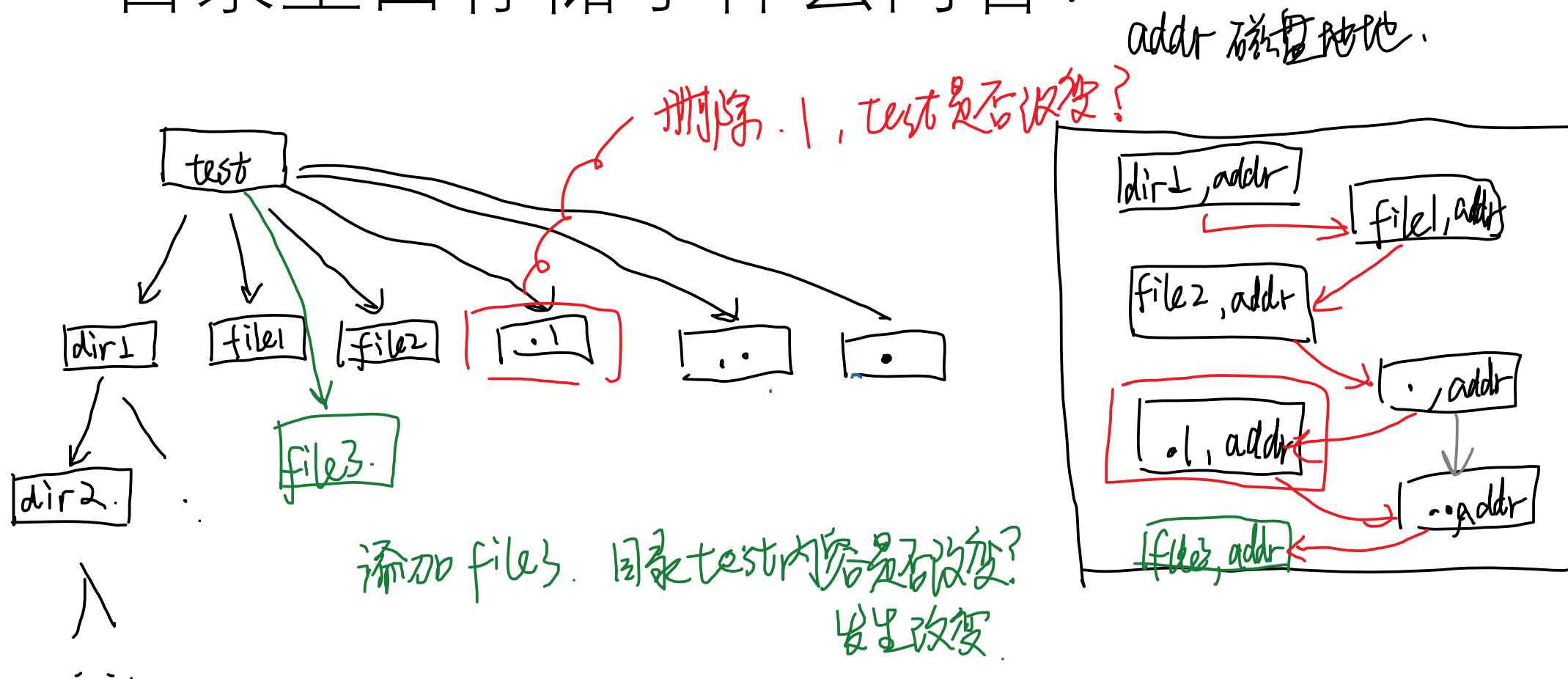
- 组权限
- 其他用户权限
- 硬

- 用户名.
- 组名.
- 文件大小 单位是字节.

- 最后修改时间
- 文件名

-lh
自动选择文件
大小的单位

目录里面存储了什么内容?



结论: 添加和删除文件, 需要改变目录内容, 需要目录的写权限

拷贝文件或者目录

\$cp 源文件 目标路径 拷贝一份源文件到目标路径下 覆盖同名文件.

\$cp 源文件 目标文件. 拷贝一份源文件, 取代目标文件. 若目标文件不存在, 则创建 覆盖源文件.

Ubuntu 不会提示覆盖行为.

-i 出现提示 $\begin{cases} Y & \text{Yes} \\ n & \text{No} \end{cases}$

⇒ 写成脚本很不方便的

-f 强制 不出现提示

-r 拷贝目录和它的所有子文件. recursive 递归.

剪切文件或者目录

\$ mv 使用方法和 cp 是一样的.

重命名. \$mv 源文件 目标文件 把文件名修改, 从名字1 → 名字2.
 名字1 名字2

-i -f

删除文件

\$rm -f

-r

不要在root用户的情况下 使用 rm -r。

树状目录显示

\$sudo apt install tree.

\$tree

\$tree 路径

\$treec -h 显示文件大小.

文件的权限

用户 $\begin{cases} r & \text{读} \\ w & \text{写} \\ x & \text{执行} \end{cases}$

组 $\begin{cases} r \\ w \\ x \end{cases}$

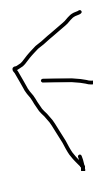
其他用户 $\begin{cases} r \\ w \\ x \end{cases}$

权限有三种：读 写 执行

用户的身份有三种：用户 组 其他用户

$rwxrwxr-x$

liao liao



liao

liao

rwx

Zhang

liao

rwx

Li

Li

~~$r-x$~~

目录的权限

rwx rwx r-x hiao hiao dir.

dir/

file1 dir1

\$ mkdir dir1 ✓ ✓ X

\$ rm file1 ✓ ✓ X

dir2 --x -w -r-- hiao hiao file2.

\$ cp file2 file3 X X ✓



dir 链表结构

那组
hiao hiao
Zhang hiao
Li Li

CP 的原理.

先读取文件内容,
再进行拷贝

修改文件的权限

\$chmod

\$man

→ manual 帮助手册

1. 文字设定法.

多次, 用逗号分隔

\$chmod [who] [+|-|=] [mode] 文件名.

who {
u 用户
g 组
o 其他
a 所有

可简写为.

+ 增加

- 去除

= 指定

mode {
r
w
x

删除读?

① -r

② a-r

③ ugo-r

-rw-rw-r-- 1 liao liao 6 Apr 7 11:35 file3

修改文件的权限

~~421~~ ~~421~~ ~~421~~ 1 liao liao 6 Apr 7 11:35 file3

权限 权限 权限

→ 3位二进制数 → 664

rwX rwxr-X
7 7 5

2. 数字设定法

\$chmod 八进制数 文件名

掩码. 二进制数.

权限掩码

002

⇒ {

rwX rwxrwx

11111111

777

000 000 010

002

111 111 101

775

110 110 110

666

000 000 010

002

110 110 100

664

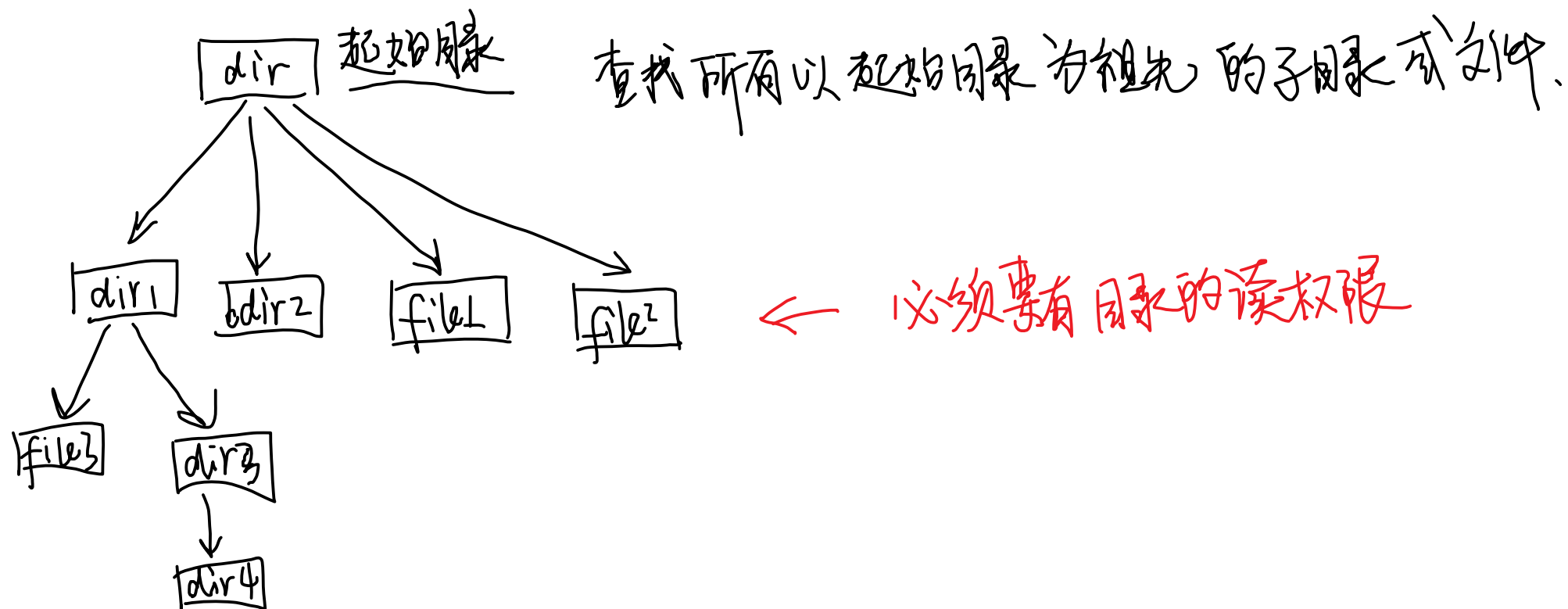
查看与改权限掩码

\$umask

修改只生效于单次登录

文件的查找

`$find` 起始目录 查找条件 操作.



按名字查找

- name

```
[liao@ubuntu ~/test]$ tree
```

```
.
├── dir2
│   ├── 1
│   ├── dir2
│   ├── file1
│   ├── file3
│   └── testdir2
├── dir3
├── dir4
├── file1
├── file2
├── file3
└── file4
```

5 directories, 7 files

```
[liao@ubuntu ~/test]$ find . -name file3
./dir2/file3
./file3
```

通配符

通配符.

* 0~多个任意字符.

"dir*"

"file*"

? 1个任意字符.

"file?"

└─ 不匹配空
└─ 不匹配10.

```
[liao@ubuntu ~/test]$ find . -name "file*"
```

```
./file1  
./file  
./dir2/file1  
./dir2/file3  
./file2  
./file3  
./file4  
./file10
```

```
[liao@ubuntu ~/test]$ find . -name "file?"
```

```
./file1  
./dir2/file1  
./dir2/file3  
./file2  
./file3  
./file4
```

```
[liao@ubuntu ~/test]$ find . -name "file[A-Z]"
```

```
./fileA
```

```
[liao@ubuntu ~/test]$ find . -name "file[A-Za-z]"
```

```
./files  
./filea  
./fileA
```

[范围] 1个范围内字符.

"file[0-9]"

索引

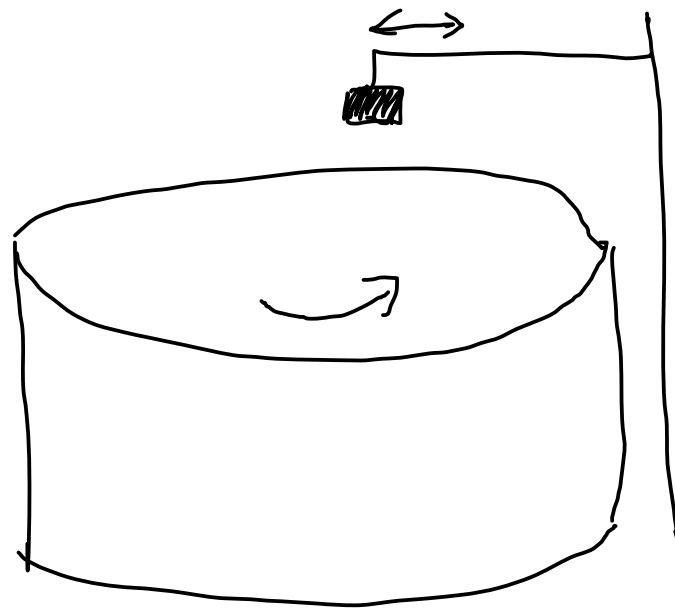
为什么 windows 搜索慢?

直接访问磁盘.

为什么 win10 & linux 快?

索引. (把文件名等信息和树形结构, 单独存放在文件)

搜索 只看索引文件.



磁盘读写是机械运动
速度很慢.

按其他信息查找

gid group

uid user

empty 查找空文件.

perm 按权限查找

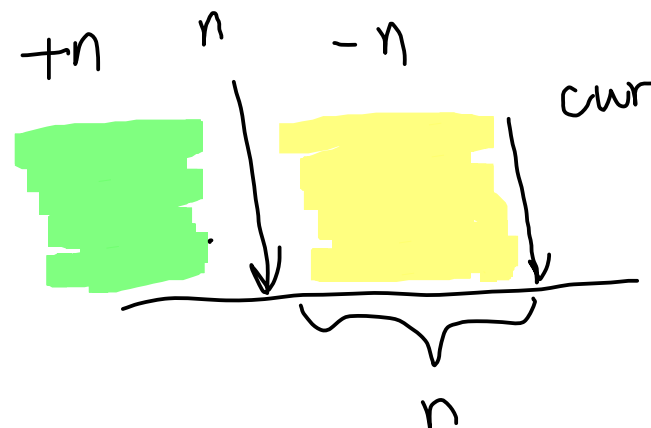
size 默认单位为块.

type 按类型查找

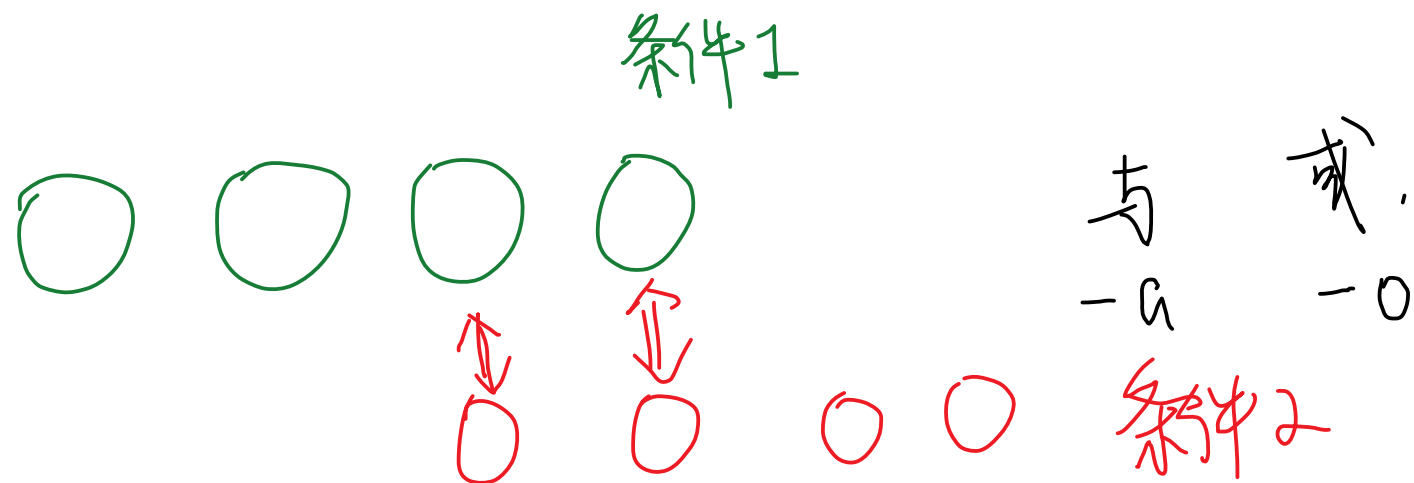
b	block (buffered) special
c	character (unbuffered) special
d	directory
p	named pipe (FIFO)
f	regular file
l	symbolic link; this is never true if ken. If you want to search for symbo
s	socket

按照时间查找

[alc1m] (min | time)
↓ ↓ ↓
时间 状态 内容
分钟 天



查找条件的运算



非操作 !