目录

[1：文件系统类型 2](#_Toc384138440)

[1.1：data block (资料区块) 3](#_Toc384138441)

[1.2：inode(ls –li xxx) 4](#_Toc384138442)

[1.3：Superblock (超级区块) 6](#_Toc384138443)

[2：man指令 7](#_Toc384138444)

[3：显示日历 7](#_Toc384138445)

[4：计算器 7](#_Toc384138446)

[5：查看有谁在线 7](#_Toc384138447)

[6：man数字所代表的的内容 7](#_Toc384138448)

[7：网络的连机状态 8](#_Toc384138449)

[8：后台执行程序 8](#_Toc384138450)

[9：将数据同步写入硬盘中的指令 8](#_Toc384138451)

[10：关机，重启启动指令(系统管理员指令) 9](#_Toc384138452)

[11：run level 10](#_Toc384138453)

[12：开机过程中文件系统错误的问题 10](#_Toc384138454)

[12.1：原因 10](#_Toc384138455)

[12.2：解决方法 11](#_Toc384138456)

[12.3：预防发生文件系统错误 12](#_Toc384138457)

[13：忘记root密码 13](#_Toc384138458)

[14：权限权组 13](#_Toc384138459)

[15：变换身份 18](#_Toc384138460)

[16:linux目录划分 18](#_Toc384138461)

[16.1：请说明/bin与/usr/bin目录所放置的执行文件有何不同之处？ 18](#_Toc384138462)

[16.2：请说明/bin与/sbin目录所放置的执行文件有何不同之处？ 18](#_Toc384138463)

[16.3：哪几个目录不能够与根目录(/)放置到不同的partition中？并请说明该目录所放置的数据为何？ 18](#_Toc384138464)

[16.4：试说明为何根目录要小一点比较好？另外在分割时，为什举/home, /usr, /var, /tmp最好与根目录放到不同的分割槽？ 试说明可能的原因为何(由目录放置数据的内容谈起)？ 18](#_Toc384138465)

[16.5：早期的 Unix 系统文件名最多允许 14 个字符，而新的 Unix 与 Linux 系统中，文件名最多可以容许几个字符？ 19](#_Toc384138466)

[16.6：我需要将一个档案的权限改为 -rwxr-xr-- 请问该如何下达指令？ 19](#_Toc384138467)

[16.7：若我需要更改一个档案的拥有者与群组，该用什么指令？ 19](#_Toc384138468)

[16.8：Linux 传统的文件系统为何？此外，常用的 Journaling 文件格式有哪些？ 19](#_Toc384138469)

[16.9：请问底下的目录与主要放置什么数据： /etc/, /etc/init.d, /boot, /usr/bin, /bin, /usr/sbin, /sbin, /dev, /var/log 19](#_Toc384138470)

[16.10：若一个档案的档名开头为『 . 』，例如 .bashrc 这个档案，代表什么？另外，如何显示出这个文件名与他的相关属性？ 20](#_Toc384138471)

[17：复制文件 20](#_Toc384138472)

[18：移动文件，改变文件的名字 21](#_Toc384138473)

[19：显示文件内容 21](#_Toc384138474)

[19.1: less 21](#_Toc384138475)

[19.2: od 22](#_Toc384138476)

[20：档案时间 22](#_Toc384138477)

[21：创建档案和目录的默认权限 23](#_Toc384138478)

[22：查看文件类型 24](#_Toc384138479)

[23：寻找命令所在的目录 24](#_Toc384138480)

[24：寻找档案 24](#_Toc384138481)

[24.1：whereis 24](#_Toc384138482)

[24.2：locate 25](#_Toc384138483)

[25：linux常用装置 25](#_Toc384138484)

[26：磁盘 26](#_Toc384138485)

[26.1：基本知识 26](#_Toc384138486)

[26.2：新增一颗硬盘时所要做的动作 26](#_Toc384138487)

[26.3：磁盘分区 27](#_Toc384138488)

[27：查看当前挂在的装置 29](#_Toc384138489)

[28：查看每个区段与 superblock 的信息 30](#_Toc384138490)

[29：档案的读取步骤 30](#_Toc384138491)

[30：磁盘与目录的容量 32](#_Toc384138492)

[30.1：df(列出文件系统的整体磁盘使用量) 32](#_Toc384138493)

[30.2：du(评估文件系统的磁盘使用量(常用在推估目录所占容量)) 33](#_Toc384138494)

[30.3：举例（如何查看一个文件大小和修改时间） 34](#_Toc384138495)

# 1：文件系统类型

由于ext3文件系统多了日志的记录， 对于系统的复原比较快速，因此建议你务必要选择新的ext3不要用ext2

较新的操作系统的档案数据除了档案实际内容外， 通常含有非常多的属性，例如 Linux 操作系统的档案权限(rwx)与文件属性(拥有者、群组、时间参数等)。 文件系统通常会将这两部份的数据分别存放在不同的区块，权限与属性放置到 inode 中，至于实际数据则放置到 data block 区块中。 另外，还有一个超级区块 (superblock) 会记录整个文件系统的整体信息，包括 inode 与 block 的总量、使用量、剩余量等。 每个 inode 与 block 都有编号，至于这三个数据的意义可以简略说明如下：

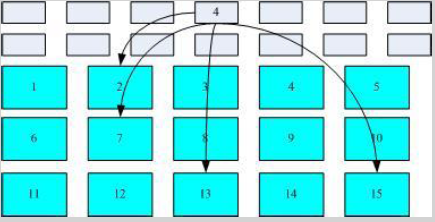
 superblock：记录此 filesystem 的整体信息，包括inode/block的总量、使用量、剩余量， 以及文件系统的格式与相关信息等；

 inode：记彔档案的属性，一个档案占用一个inode，同时记录此档案的数据所在的block 号码；

 block：实际记录档案的内容，若档案太大时，会占用多个 block 。

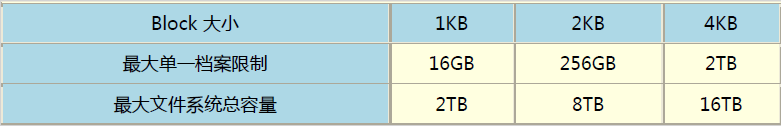
由于每个 inode 与 block 都有编号，而每个档案都会占用一个 inode ，inode 内则有档案数据放置的block 号码。 因此，我们可以知道的是，如果能够找到档案的inode 的话，当然自然就会知道这个档案所放置数据的 block 号码， 当然也就能够读出该档案的实际数据了。这是个比较有效率的作法，因为如此一来我们的磁盘就能够在短时间内读取出全部的数据， 读写的效能比较好啰。

我们将 inode 与 block 区块用图解来说明一下，如下图所示，文件系统先格式化出 inode 与 block 的区块，假设某一个档案的属性与权限数据是放置到 inode 4 号(下图较小方格内)，而这个 inode 记录了档案数据的实际放置点为 2, 7, 13, 15 这四个 block 号码，此时我们的操作系统就能够据此来排列磁盘的阅读顺序，可以一口气将四个 block 内容读出来！ 那么数据的读取就如同下图中的箭头所指定的模样了。



## 1.1：data block (资料区块)

data block 是用来放置档案内容数据地方，在 Ext2 文件系统中所支持的 block 大小有 1K, 2K 及 4K 三种而已。在格式化时 block 的大小就固定了，且每个 block 都有编号，以方便 inode 的记录啦。 不过要注意的是，由于block 大小的差异，会导致该文件系统能够支持癿最大磁盘容量不最大单一档案容量幵丌相同。 因为 block 大小而产生癿 Ext2 文件系统限制如下：(注2)



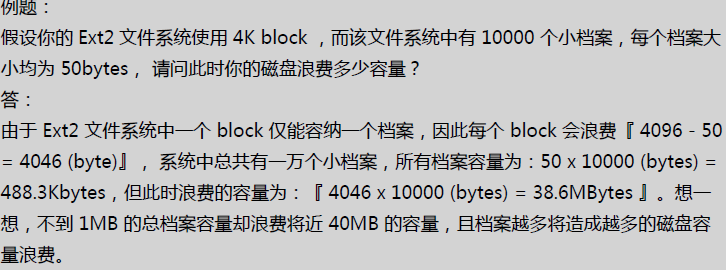
Ext2 文件系统的 block 基本限制如下：

 a:原则上，block 的大小与数量在格式化完就不能够再改变了(除非重新格式化)；

 b:每个 block 内最多只能够放置一个档案的数据；

 c:承上，如果档案大于 block 的大小，则一个档案会占用多个 block 数量；

 d:承上，若档案小于 block ，则该 block 的剩余容量就不能够再被使用了(磁盘空间会浪费)。



## 1.2：inode(ls –li xxx)

inode 记录的档案数据至少有底下这些：(注4)

 a:该档案的存取模式(read/write/excute)；

 b:该档案的拥有者不群组(owner/group)；

 c:该档案的容量；

 d:该档案建立或状态改变的时间(ctime)；

 e:最近一次的读取时间(atime)；

 f:最近修改的时间(mtime)；

 g:定义档案特性的旗标(flag)，如 SetUID...；

 h:该档案真正内容的指向 (pointer)；

inode 的数量与大小也是在格式化时就已经固定了，除此之外 inode 还有些什么特色呢？

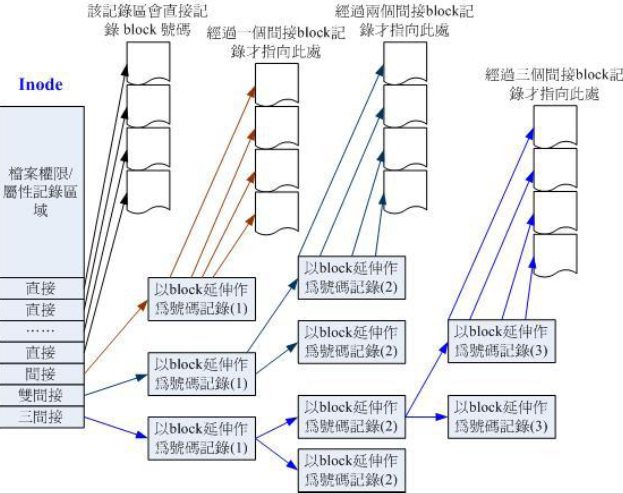
 a:每个 inode 大小均固定为 128 bytes；

 b:每个档案都仅会占用一个 inode 而已；

 c:承上，因此文件系统能够建立的档案数量与 inode 的数量有关；

 d:系统读取档案时需要先找到 inode，并分析 inode 所记录的权限与用户是否符合，若符合才能够开始实际读取 block 的内容。

我们约略来分析一下 inode / block 与档案大小的关系好了。inode 要记录的数据非常多，但偏偏又只有 128bytes 而已， 而 inode 记录一个 block 号码要花掉 4byte ，假设我一个档案有 400MB 且每个 block 为 4K 时， 那么至少也要十万笔 block 号码的记录呢！inode 哪有这举多可记录的信息？为此我们的系统很聪明的将 inode 记录 block 号码的区域定义为12个直接，一个间接, 一个双间接不一个三间接记录区。这是啥？我们将 inode 的结构画一下好了。



## 1.3：Superblock (超级区块)

Superblock 是记录整个 filesystem 相关信息的地方， 没有 Superblock ，就没有这个 filesystem 了。他记录的信息主要有：

 a:block 与 inode 的总量；

 b:未使用与已使用的 inode / block 数量；

 c:block 与 inode 的大小 (block 为 1, 2, 4K，inode 为 128 bytes)；

 d:filesystem 的挂载时间、最近一次写入数据的时间、最近一次检验磁盘 (fsck) 的时间等文件系统的相关信息；

 e:一个 valid bit 数值，若此文件系统已被挂载，则 valid bit 为 0 ，若未被挂载，则 valid bit 为 1 。

Superblock 是非常重要的，因为我们这个文件系统的基本信息都写在这里，因此，如果 superblock 死掉了， 你的文件系统可能就需要花费很多时间去挽救啦！一般来说， superblock 的大小为 1024bytes。相关的 superblock 讯息我们等一下会以 dumpe2fs 指令来呼叫出来观察喔！

此外，每个 block group 都可能含有 superblock 喔！但是我们也说一个文件系统应该仅有一个 superblock 而已，那是怎么回事啊？ 事实上除了第一个 block group 内会含有 superblock 之外，后续的 block group 不一定含有 superblock ， 而若含有 superblock 则该 superblock 主要是做为第一个 block group 内 superblock 的备份咯，这样可以进行 superblock 的救援呢！

# 2：man指令

通过/xxxx,来搜索关键字

man指令描述采用正则的方式来描述[]表示字符集只能筛选一个

# 3：显示日历

cal year | cal month year

# 4：计算器

bc scale=number (number是小数点的位数)

quit离开bc的软件环境

# 5：查看有谁在线

who

# 6：man数字所代表的的内容

|  |  |  |
| --- | --- | --- |
| 1 | 用户在shell环境中可以操作的指令或可执行文件 |  |
| 2 | 系统核心可呼叫的函数与工具等 |  |
| 3 | 一些常用的函数(function)与函式库(library)，大部分为C的函式库(libc) |  |
| 4 | 装置档案的说明，通常在/dev下的档案 |  |
| 5 | 配置文件或者是某些档案的格式 |  |
| 6 | 游戏(games) |  |
| 7 | 惯例与协议等，例如Linux文件系统、网绚协议、ASCII code等等的说明 |  |
| 8 | 系统管理员可用的管理指令 |  |
| 9 | 跟kernel有关的文件 |  |

# 7：网络的连机状态

netstat -a

# 8：后台执行程序

ps -aux

# 9：将数据同步写入硬盘中的指令

sync

所有的数据都得要被读入内存后才能够被CPU所处理，但是数据又常常需要由内存写回硬盘当中(例如储存的动作)。 由于硬盘的速度太慢(相对于内存来说)，如果常常让数据在内存与硬盘中来回写入/读出，系统的效能就不会太好。 因此在Linux系统中，为了加快数据的读取速度，所以在默认的情况中， 某些已经加载内存中的数据将不会直接被写回硬盘，而是先暂存在内存当中，如此一来， 如果一个数据被你重复的改写，那么由于他尚未被写入硬盘中，因此可以直接由内存当中读取出来， 在速度上一定是快上相当多的！ 不过，如此一来也造成些讲的困扰，那就是万一你的系统因为某些特殊情况造成不正常关机 (例如停电或者是不小心踢到power)时，由于数据尚未被写入硬盘当中，哇！所以就会造成数据的更新不正常啦！ 那要怎么办呢？这个时候就需要sync这个指令来进行数据的写入动作啦！ 直接在文字接口下输入sync，那么在内存中尚未被更新的数据，就会被写入硬盘中！

# 10：关机，重启启动指令(系统管理员指令)

shutdown ，reboot, halt, poweroff

shutdown -h now 立刻关机，其中 now 相当于时间为 0 的状态

shutdown -h 20:25 系统在今天的 20:25 分会关机，若在21:25才下达此指令，则隔天才关机

shutdown -h +10 系统再过十分钟后自动关机

shutdown -r now 系统立刻重新启动

shutdown -r +30 'The system will reboot' 再过三十分钟系统会重新启动，并显示后面的讯息给所有在在线的使用者

shutdown -k now 'This system will reboot' 仅发出警告信件的参数！系统并不会关机啦！吓唬人！

此外，需要注意的是，时间参数请务必加入指令中，否则shutdown会自动跳到 run-level 1 (就是单人维护的登入情况)

# 11：run level

之前谈到的是系统运作的模式，分为纯文本(run level 3)及图形接口模式(run level 5)。除了这两种模式外，有没有其他模式呢？其实Linux共有七种执行等级

run level 0：关机

run level 3：纯文本模式

run level 5：含有图形接口模式

run level 6：重新启动

如果你想要关机的话， 除了上述的shutdown -h now以及poweroff之外，你也可以使用 init 0 来关机

# 12：开机过程中文件系统错误的问题

## 12.1：原因

在开机的过程中最容易遇到的问题就是硬盘可能有坏轨或文件系统发生错误(数据损毁)的情况， 这种情况虽然不容易发生在稳定的Linux系统下，不过由于不当的开关机行为， 还是可能会造成的，常见的发生原因可能有：

1.1：最可能发生的原因是因为断电或不正常关机所导致的文件系统发生错误， 鸟哥的主机就曾经发生过多次因为跳电，家里的主机又没有安装不断电系统， 结果就导致硬盘内的文件系统错误！文件系统错误并非硬件错误，而是软件数据的问题喔！

1.2：硬盘使用率过高或主机所在环境不良也是一个可能的原因， 例如你开放了一个FTP服务，里面有些数据很有用， 所以一堆人抢着下载，如果你又不是使用较稳定的SCSI接口硬盘，仅使用一般PC使用的硬盘， 虽然机率真的不高，但还是有可能造成硬盘坏轨的。此外，如果主机所在环境没有散热的设备， 或者是相对湿度比较高的环境，也会容易造成硬盘的损坏喔！

## 12.2：解决方法

因为出错扇区所挂载的目录不同，处理的流程困难度就有差异了。 举例来说，如果你的根目录『/』并没有损毁，那就很容易解决，如果根目录已经损毁了，那就比较麻烦

1：如果根目录没有损毁

假如你发生错误的partition是在/dev/sda7这一块，那举在开机的时候，屏幕应该会告诉你：press root password or ctrl+D : 这时候请输入root的密码登入系统，然后进行如下动作：

在光标处输入root密码登入系统，进行单人单机的维护工作；

输入『 fsck /dev/sda7 』(fsck 为文件系统检查的指令，/dev/sda7为错误的partition，请依你的情况下达参数)， 这时屏幕会显示开始修理硬盘的讯息，如果有发现任何的错误时，屏幕会显示： clear [Y/N]？ 的询问讯息，就直接输入 Y 吧！

修理完成之后，以 reboot 重新启动

2：如果根目录损毁了

一般初学者喜欢将自己的硬盘只划分为一个大partition，亦即只有根目录， 那文件系统错误一定是根目录的问题啰！这时你可以将硬盘拔掉，接到另一台Linux系统的计算机上， 并且不要挂载(mount)该硬盘，然后以root的身份执行『 fsck /dev/sdb1 』(/dev/sdb1 指的是你的硬盘装置文件名，你要依你的实际状况来设定)，这样就 OK 啰！

另外，也可以使用近年来很热门的Live CD，也就是利用光盘开机就能够进入Linux操作系统的特性， 你可以前往：『http://knoppix.tnc.edu.tw/』 这个网站来下载，幵且刻录成为CD，这个时候先用Live CD光盘开机，然后使用fsck去修复原本的根目录， 例如： fsck /dev/sda1 ，就能够救回来了！

3：如果硬盘整个坏掉：

如果硬盘实在坏的离谱时，那就先将旧硬盘内的数据，能救出来的救出来，然后换一颗硬盘来重新安装Linux吧！ 不要不愿意换硬盘啊！啥时后硬盘会坏掉谁也说不准的！

## 12.3：预防发生文件系统错误

妥善保养硬盘： 例如：主机通电之后不要搬动，避免移动或震动硬盘；尽量降低硬盘的温度，可以加装风扇来冷即硬盘； 戒者可以换装 SCSI 硬盘。

 划分不同的partition： 为什么磁盘分区这么重要！因为Linux每个目录被读写的频率不同，妥善的分割将会让我们的Linux更安全！ 通常我们会建议划分下列的磁盘区块：

o /

o /boot

o /usr

o /home

o /var

这样划分有些好处，例如/var是系统默认的一些数据暂存或者是cache数据的储存目录， 像 e-mail 就含在这里面。如果还有使用proxy时，因为常常存取，所以有可能会造成磁盘损坏， 而当这部份的磁盘损坏时，由于其他的地方是没问题的，因此资料得以保存，而且在处理时也比较容易！

# 13：忘记root密码

先将系统重新启动，在读秒的时候按下任意键就会出现如同第四章图3.2的选单画面，仔细看选单底下的说明， 按下『e』就能够进入grub的编辑模式了

|  |  |
| --- | --- |
| kernel /vmlinuz-2.6.18-128.el5 ro root=LABEL=/ rhgb quiet |  |

此时，请将光标移动到kernel那一行，再按一次『 e 』进入kernel该行的编辑画面中， 然后在出现的画面当中，最后方输入 single ：

|  |  |
| --- | --- |
| kernel /vmlinuz-2.6.18-128.el5 ro root=LABEL=/ rhgb quiet single |  |

再按下『 Enter 』确定之后，按下 b 就可以开机进入单人维护模式了！ 在这个模式底下，你会在tty1的地方不需要输入密码即可取得终端机的控制权(而且是使用root的身份喔！)。 之后就能够修改root的密码了！请使用 passwd 来修改root的密码喔！

# 14：权限权组

r:对于目录来说可以查看它里面文件的权限 通过 ls –al xxx 来列出它里面的文件名，

并且这些文件的权限都是显示问号，但是不具备进入该目录的权限，对于文件它可以查看它的内容

x:对于目录来说可以cd 进入该目录，对于文件说如果是可执行文件一般可以执行它

例题： 假设有个账号名称为dmtsai，他的家目录在/home/dmtsai/，dmtsai对此目录具有[rwx]的权限。 若在此目录下有个名为the\_root.data的档案，该档案的权限如下： -rwx------ 1 root root 4365 Sep 19 23:20 the\_root.data 请问dmtsai对此档案的权限为何？可否删除此档案？

答： 如上所示，由二dmtsai对此档案来说是『others』的身份，因此这个档案他无法读、无法编辑也无法执行， 也就是说，他无法变动这个档案的内容就是了。 但是由二这个档案在他的家目录下， 他在此目录下具有rwx的完整权限，因此对二the\_root.data这个『档名』来说，他是能够『删除』的！ 结论就是，dmtsai这个用户能够删除the\_root.data这个档案！

例题：让一个使用者 vbird 能够进行『cp /dir1/file1 /dir2』的指令时，请说明 dir1, file1, dir2 的最小所需权限为何？

答： 执行 cp 时， vbird 要『能够读取来源文件，并且写入目标文件！』所以应参考上述第二点与第四点的说明！ 因此各档案/目录的最小权限应该是：

 dir1 ：至少需要有 x 权限；

 file1：至少需要有 r 权限；

dir2 ：至少需要有 w, x 权限。

|  |  |
| --- | --- |
| 例题： 有一个档案全名为 /home/student/www/index.html ，各相关档案/目录的权限如下： drwxr-xr-x 23 root root 4096 Sep 22 12:09 /  drwxr-xr-x 6 root root 4096 Sep 29 02:21 /home  drwx------ 6 student student 4096 Sep 29 02:23 /home/student  drwxr-xr-x 6 student student 4096 Sep 29 02:24 /home/student/www  -rwxr--r-- 6 student student 369 Sep 29 02:27 /home/student/www/index.html  请问 vbird 这个账号(不属与student群组)能否读取 index.html 这个档案呢？  答： 虽然 www 与 index.html 是可以让 vbird 读取的权限，但是因为目录结构是由根目录一层一层读取的， 因此 vbird 可进入 /home 但是却不可进入 /home/student/ ，既然连进入 /home/student 都不许了， 当然就读不到 index.html 了！所以答案是『vbird不会读取到 index.html 的内容』喔 ！ 那要如何修改权限呢？其实只要将 /home/student 的权限修改为最小 711 ，或者直接给予 755 |  |

情景模拟：假设系统中有两个账号，分别是 alex与arod ，这两个人除了自己群组之外还共同支持一个名为 project 的群组。假设这两个用户需要共同拥有 /srv/ahome/ 目录的开发权，且该目录不许其他人进入查阅。 请问该目录的权限设定应为何？请先以传统权限说明，再以 SGID 的功能解析。

答：

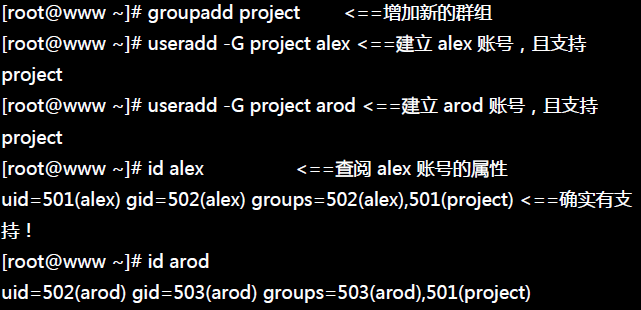
准备：

 目标：了解到为何项目开发时，目录最好需要设定 SGID 的权限！

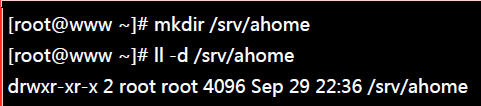
 前提：多个账号支持同一群组，且共同拥有目录的使用权！

 需求：需要使用 root 的身份来进行 chmod, chgrp 等帮用户设定好他们的开发环境才行！ 这也是管理员癿重要任务之一！

首先我们得要先制作出这两个账号的相关数据，账号/群组的管理在后续我们会介绍， 您这里先照着底下的指令来制作即可：



1. 首先建立所需要开发的项目目录：



2. 从上面的输出结果可发现 alex与arod 都不能在该目录内建立档案，因此需要进行权限与属性的修改。 由于其他人均不可进入此目录，因此该目录的群组应为project，权限应为770才合理。



3. 实际分别以两个使用者来测试看看，情况会是如何？先用 alex 建立档案，然后用 arod 去处理看看。



由上面的结果我们可以知道，若单纯使用传统的rwx 而已，则对刚刚 alex 建立的 abcd 这个档案来说， arod 可以删除他，但是却不能编辑他！这不是我们要的样子啊！赶紧来重新规划一下。

4. 加入 SGID 的权限在里面，并进行测试看看：



所以最终的结果显示，此目录的权限最好是『2770』，所属档案拥有者属于root即可，至于群组必须要为两人共同支持的project 这个群组才行！

# 15：变换身份

su – xxx（用户名） exit退出

# 16:linux目录划分

## 16.1：请说明/bin与/usr/bin目录所放置的执行文件有何不同之处？

/bin主要放置在开机时，以及进入单人维护模式后还能够被使用的指令，至于/usr/bin则是大部分软件提供的指令放置处。

## 16.2：请说明/bin与/sbin目录所放置的执行文件有何不同之处？

/bin放置的是一般用户惯用的指令，至于/sbin则是系统管理员才会使用到的指令。不过/bin与/sbin都与开机、单人维护模式有关。 更多的执行档会被放置到/usr/bin及/usr/sbin底下。

## 16.3：哪几个目录不能够与根目录(/)放置到不同的partition中？并请说明该目录所放置的数据为何？

/etc(配置文件), /bin(一般身份可用执行文件), /dev(装置档案), /lib(执行档的函式库或核心模块等), /sbin(系统管理员可用指令)

## 16.4：试说明为何根目录要小一点比较好？另外在分割时，为什举/home, /usr, /var, /tmp最好与根目录放到不同的分割槽？ 试说明可能的原因为何(由目录放置数据的内容谈起)？

根据FHS的说明，越小的/可以放置的较为集中且读取频率较不频繁，可避免较多的错误。 至与/home(用户家目录), /usr(软件资源), /var(变动幅度较大的数据), /tmp(系统暂存，数据莫名)中， 因为数据量较大或者是读取频率较高，或者是不明的使用情况较多，因此建议不要与根目彔放在一起， 也会有助于系统安全。

## 16.5：早期的 Unix 系统文件名最多允许 14 个字符，而新的 Unix 与 Linux 系统中，文件名最多可以容许几个字符？

由二使用Ext2/Ext3文件系统，单一档名可达 255 字符，完整文件名 (包含路径) 可达 4096 个字符

## 16.6：我需要将一个档案的权限改为 -rwxr-xr-- 请问该如何下达指令？

chmod 754 filename 或 chmod u=rwx,g=rx,o=r filename

## 16.7：若我需要更改一个档案的拥有者与群组，该用什么指令？

chown, chgrp

## 16.8：Linux 传统的文件系统为何？此外，常用的 Journaling 文件格式有哪些？

传统文件格式为：ext2, Journaling 有 ext3 及 Reiserfs 等

## 16.9：请问底下的目录与主要放置什么数据： /etc/, /etc/init.d, /boot, /usr/bin, /bin, /usr/sbin, /sbin, /dev, /var/log

o /etc/：几乎系统的所有配置文件案均在此，尤其 passwd,shadow

o /etc/init.d：系统开机的时候加载服务的 scripts 的摆放地点

o /boot：开机配置文件，也是预设摆放核心 vmlinuz 的地方

o /usr/bin, /bin：一般执行档摆放的地方

o /usr/sbin, /sbin：系统管理员常用指令集

o /dev：摆放所有系统装置档案的目录

o /var/log：摆放系统注册表档案的地方

## 16.10：若一个档案的档名开头为『 . 』，例如 .bashrc 这个档案，代表什么？另外，如何显示出这个文件名与他的相关属性？

有『 . 』为开头的为隐藏档，需要使用 ls -a 这个 -a 的选项才能显示出隐藏档案的内容，而使用 ls -al 才能显示出属性。

# 17：复制文件

可以在任何目录中做其他目录(相对目录和绝对目录)之间文件的复制(相比windows的优势)，当然前提是这个档案要有read权限

在预设的条件中， cp 的来源档与目的档的权限是不同的，目的档的拥有者通常会是指令操作者本身。举例来说， 上面的范例二中，由于我是 root 的身份，因此复制过来的档案拥有者的群组就改变成为 root 所有了

由于具有这个特性，因此当我们在进行备份的时候，某些需要特别注意的特殊权限档案， 例如密码文件 (/etc/shadow) 以及一些配置文件，就不能直接以 cp 来复制，而必须要加上 -a 或者是 -p 等等可以完整复制档案权限的选项才行！另外，如果你想要复制档案给其他的使用者， 也必须要注意到档案的权限(包含读、写、执行以及档案拥有者等等)， 否则，其他人还是无法针对你给予的档案进行修订的动作

cp –a sourcefile destdir 🡪 复制文件的属性跟源文件一模一样

cp –i sourcefile destdir 🡪 若目标文件(destination)已经存在时，在覆盖时会先询问动作的进行(常用)

cp –r sourcedir destdir 🡪 若目标文件(destination)已经存在时，在覆盖时会先询问动作的进行(常用)

cp -s ~/eclipse/eclipse ~/eclipse/eclipse\_slink 建立ecplise的快捷方式(符号链接，区别实体链接 -l)

cp –u sourcefile destdir 目标档案与来源档案有差异时，才会被复制，常用语备份工作

# 18：移动文件，改变文件的名字

mv source destdir 移动普通文件或者目录

mv sourcefilename destfilename 改名字

# 19：显示文件内容

## 19.1: less

空格键 ：向下翻动一页；

[pagedown]：向下翻动一页；

 [pageup] ：向上翻动一页；

y:向上滚动一行

enter:向下滚动一行

 /字符串 ：向下搜寻『字符串』的功能；

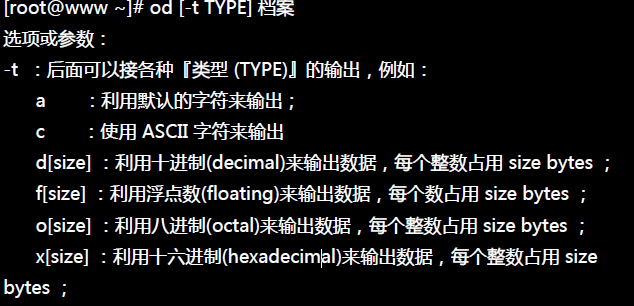
 ?字符串 ：向上搜寻『字符串』的功能；

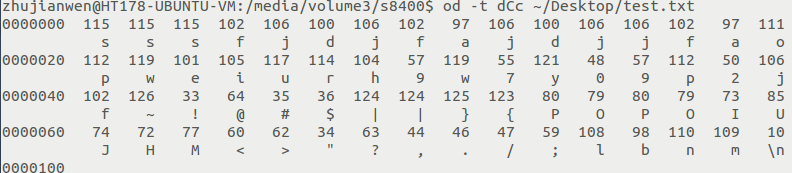
 n ：重复前一个搜寻 (与 / 或 ? 有关！)

 N ：反向的重复前一个搜寻 (与 / 或 ? 有关！)

 q ：离开 less 这个程序；

## 19.2: od





左边第一栏用8进制位来表示字节数 2 \* 8（的一次方）=16个字节

一个字符占一个字节，一个汉字占3个字节

# 20：档案时间

modification time (mtime) ： 当该档案的『内容数据』变更时，就会更新这个时间！内容数据指的是档案的内容，而不是档案的属性或权限喔！

 status time (ctime)： 当该档案的『状态 (status)』改变时，就会更新这个时间，举例来说，像是权限与属性被更改了，都会更新这个时间啊。

 access time (atime)： 当『该档案的内容被取用』时，就会更新这个读取时间 (access)。举例来说，我们使用 cat 去读取 /etc/man.config ， 就会更新该档案的 atime 了。

zhujianwen@HT178-UBUNTU-VM:/media/volume3/s8400$ ls -l ~/Desktop/test.txt （默认显示的是mtime）

-rw-rw-r-- 1 zhujianwen zhujianwen 64 Mar 12 11:10 /home/zhujianwen/Desktop/test.txt

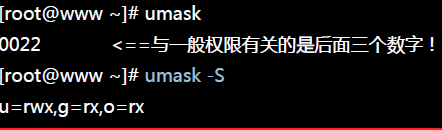
zhujianwen@HT178-UBUNTU-VM:/media/volume3/s8400$ ls -l --time=atime ~/Desktop/test.txt

-rw-rw-r-- 1 zhujianwen zhujianwen 64 Mar 12 11:10 /home/zhujianwen/Desktop/test.txt

zhujianwen@HT178-UBUNTU-VM:/media/volume3/s8400$ ls -l --time=ctime ~/Desktop/test.txt

-rw-rw-r-- 1 zhujianwen zhujianwen 64 Mar 12 11:10 /home/zhujianwen/Desktop/test.txt

# 21：创建档案和目录的默认权限



例题： 假设你的 umask 为 003 ，请问该 umask 情况下，建立的档案与目录权限为？

答： umask 为 003 ，所以拿掉的权限为 --------wx，

因此： 档案： (-rw-rw-rw-) - (--------wx) = -rw-rw-r—

目录： (drwxrwxrwx) - (--------wx) = drwxrwxr--

解释：

 a:若使用者建立为『档案』则预设『没有可执行( x )权限』，亦即只有 rw 这两个项目，也就是最大为 666 分，预设权限如下： -rw-rw-rw-

 b:若用户建立为『目录』，则由于x 与是否可以进入此目录有关，因此默认为所有权限均开放，亦即为 777 分，预设权限如下： drwxrwxrwx

要设定umask 直接在后面跟一个数字

# 22：查看文件类型



# 23：寻找命令所在的目录

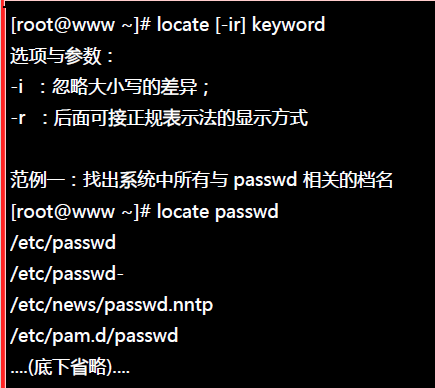
which ls

# 24：寻找档案

## 24.1：whereis



## 24.2：locate



# 25：linux常用装置



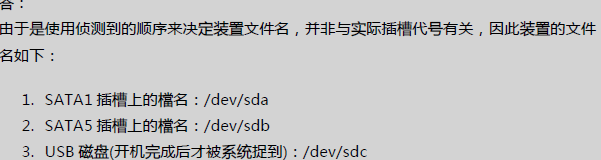
例题： 假设你的主机仅有一颗IDE接口的磁盘，而这一颗磁盘接在IDE2癿Master上面，请问他在Linux操作系统里面的装置文件名为何？

答： 比较上表的装置文件名对照，IDE2癿Master之装置文件名为/dev/hdc

例题：

如果你的PC上面有两个SATA磁盘以及一个USB磁盘，而主板上面有六个SATA的插槽。这两个SATA磁盘分别安插在主板上的SATA1, SATA5插槽上， 请问这三个磁盘在

Linux中的装置文件名为何？



# 26：磁盘

## 26.1：基本知识

扇区(Sector)为最小的物理储存单位，每个扇区为 512 bytes；

 将扇区组成一个圆，那就是磁柱(Cylinder)，磁柱是分割槽(partition)的最小单位；

 第一个扇区最重要，里面有：(1)主要开机区(Master boot record, MBR)及分割表(partition table)， 其中 MBR 占有 446 bytes，而 partition table 则占有 64 bytes。

各种接口的磁盘在Linux中的文件名分别为：

 /dev/sd[a-p][1-15]：为SCSI, SATA, USB, Flash随身碟等接口的磁盘文件名；

 /dev/hd[a-d][1-63]：为 IDE 接口的磁盘文件名；

## 26.2：新增一颗硬盘时所要做的动作

1. 对磁盘进行分割，以建立可用的 partition ；

2. 对该 partition 进行格式化( format )，以建立系统可用的 filesystem；

3. 若想要仔绅一点，则可对刚刚建立好的 filesystem 进行检验；

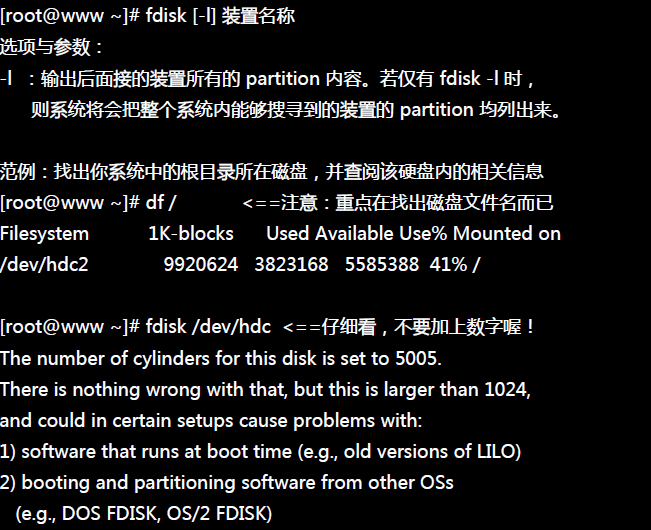
4. 在 Linux 系统上，需要建立挂载点 ( 亦即是目录 )，并将他挂载上来；

5. 磁盘分区槽 (partition) 需要定多大

6. 是否需要加入 journal 的功能？

7. inode 与 block 的数量应该如何规划等等的问题。

## 26.3：磁盘分区







使用『 p 』可以列出目前这颗磁盘的分割表信息，这个信息的上半部在显示整体磁盘的状态。 以鸟哥这颗磁盘为例，这个磁盘共有 41.1GB 左右的容量，共有 5005 个磁柱，每个磁柱透过 255 个磁头在管理读写， 每个磁头管理 63 个扇区，而每个扇区的大小均为 512bytes ，因此每个磁柱为『 255\*63\*512 = 16065\*512 = 8225280bytes 』。 下半部的分割表信息主要在列出每个分割槽的个别信息项目。每个项目的意义为：

 Device：装置文件名，依据不同的磁盘接口/分割槽位置而变。

 Boot：是否为开机引寻块块？通常Windows 系统的 C 需要这块！

 Start, End：这个分割槽在哪个磁柱号码之间，可以决定此分割槽的大小；

 Blocks：就是以 1K 为单位的容量。如上所示，/dev/hdc1 大小为104391K = 102MB

 ID, System：代表这个分割槽内的文件系统应该是啥！不过这个项目只是一个提示而已， 不见得真的代表此分割槽内的文件系统喔！

从上表我们可以发现几件事情：

 整部磁盘还可以进行额外的分割，因为最大磁柱为 5005 ，但叧使用到 2052 号而已；

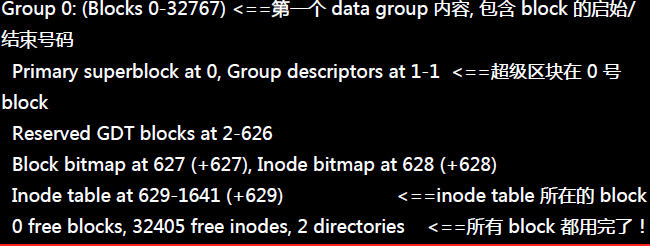
 /dev/hdc5 是由 /dev/hdc4 分割出来的，因为 /dev/hdc4 为 Extended，且 /dev/hdc5 磁柱号码在 /dev/hdc4 之内；

# 27：查看当前挂在的装置



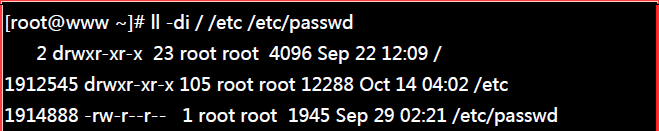
# 28：查看每个区段与 superblock 的信息





# 29：档案的读取步骤

举例来说，如果我想要读取 /etc/passwd 这个档案时，系统是如何读取的呢？



在鸟哥的系统上面与 /etc/passwd 有关的目录与档案数据如上表所示，该档案的读取流程为(假设读取者身份为 vbird 这个一般身份使用者)：

1. / 的 inode：

透过挂载点的信息找到 /dev/hdc2 的 inode 号码为 2 的根目录 inode，且 inode 规范的权限让我们可以读取该 block 的内容(有 r 与 x) ；

2. / 的 block：

经过上个步骤取得 block 的号码，并找到该内容有 etc/ 目录的 inode 号码 (1912545)；

3. etc/ 的 inode：

读取 1912545 号 inode 得知 vbird 具有 r 与 x 的权限，因此可以读取 etc/ 的 block 内容；

4. etc/ 的 block：

经过上个步骤取得 block 号码，并找到该内容有 passwd 档案的 inode 号码 (1914888)；

5. passwd 的 inode：

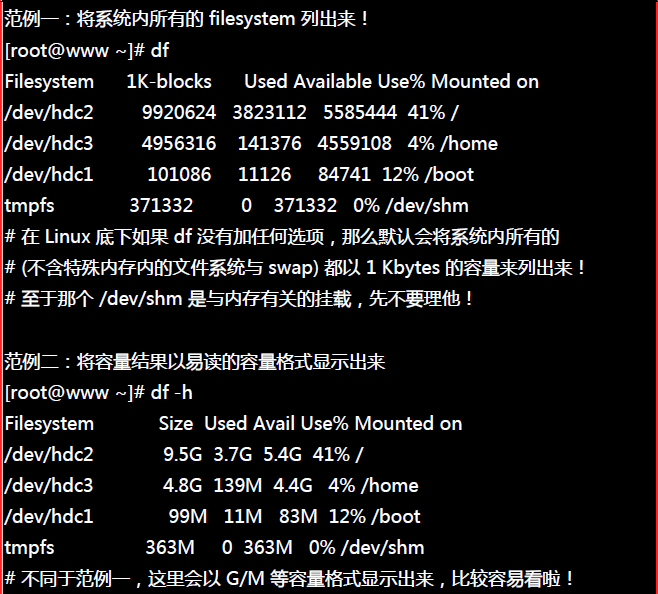
读取 1914888 号 inode 得知 vbird 具有 r 的权限，因此可以读取 passwd 的 block 内容；

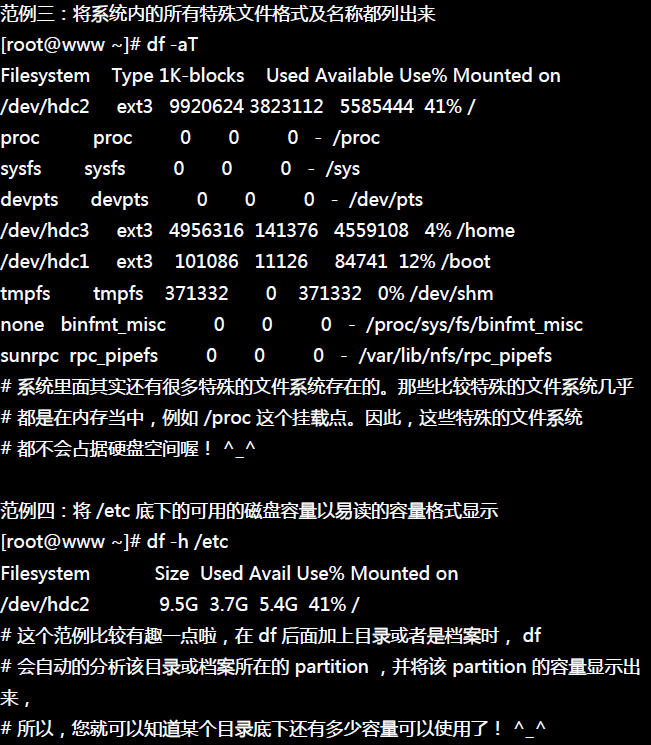
6. passwd 的 block：

最后将该 block 内容的数据读出来。

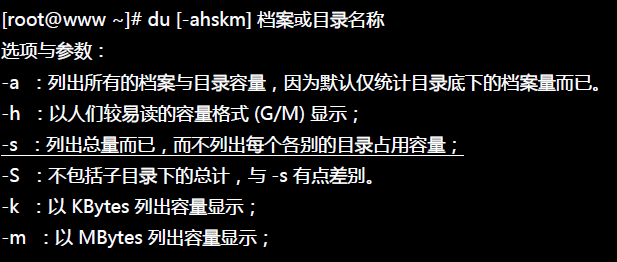
# 30：磁盘与目录的容量

## 30.1：df(列出文件系统的整体磁盘使用量)





## 30.2：du(评估文件系统的磁盘使用量(常用在推估目录所占容量))



## 30.3：举例（如何查看一个文件大小和修改时间）

