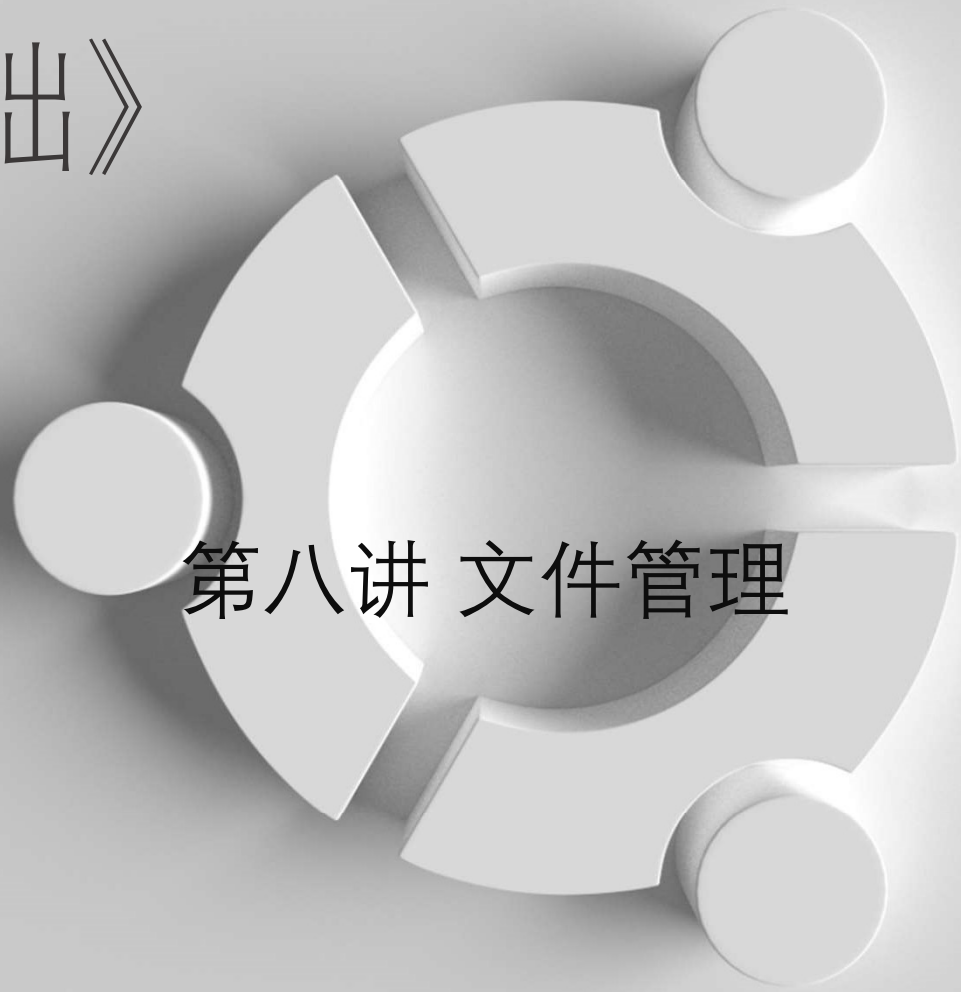
河北师范大学 软件学院
Software College of Hebei Normal University

《Linux基础》



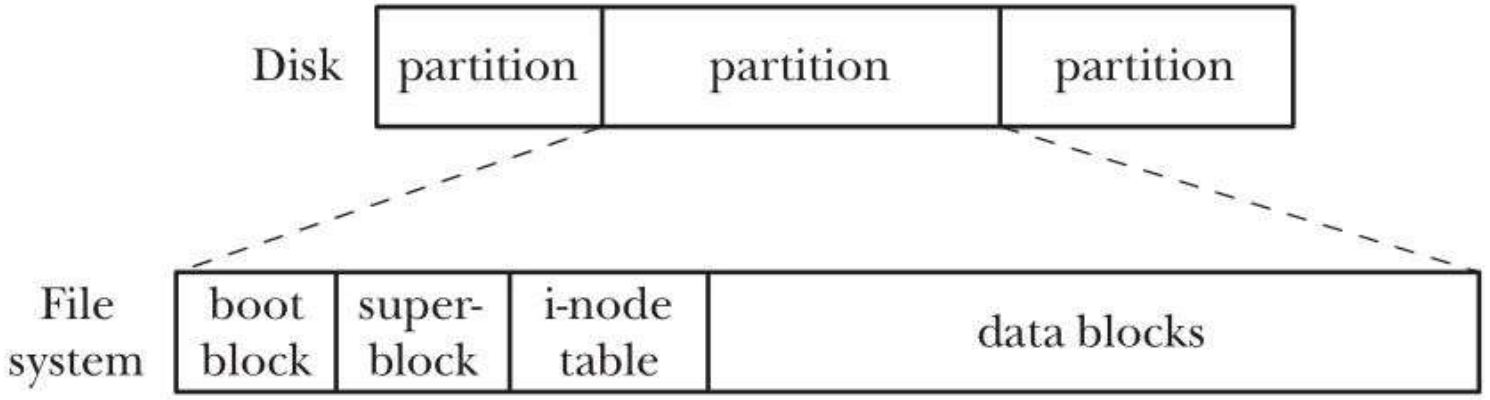
第八讲 文件管理

一切皆文件

- 在Linux上，一切皆是文件。外接设备也会被映射为文件，在/dev目录下。
- 目录也是文件，一种特殊的文件，记录的是其他文件的信息。
- 这是从Unix继承过来的思想，具有统一性的设计理念，对开发以及平常使用都具备统一的操作方式。
- Linux上的文件名称区分大小写，这点和Windows不同，Windows是不区分的。

Linux文件系统简明解释

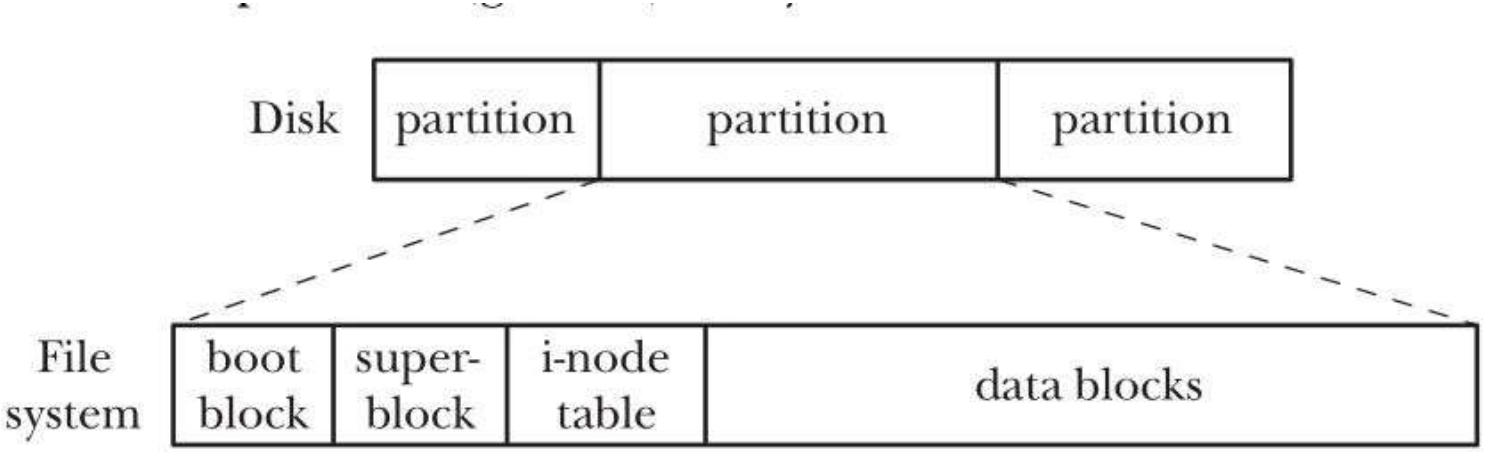
- 磁盘被分割成块进行存储，称为扇区。一般一个扇区512字节。文件系统在此基础上把数据存储分为Boot block，Super block，i-node table，Data blocks几个区域。实际存储数据的是Data blocks。
- Super block存储文件系统类型、i-node table大小等信息。 i-node记录文件在DataBlocks的存储位置。
- Boot block对于可启动分区有用。如果系统安装在此分区，则此区域存储启动信息。



Linux文件系统简明解释

- 在系统层面来说，实际是通过文件路径名称找到文件的i-node然后对文件数据进行操作。目录文件记录了其他文件的文件名与i-node，对于用户使用来说，这些都是透明的，用户并不需要关心这些。

- 了解这些，对后面讲到的软链接与硬链接有帮助。



相关命令

ls	列出文件/目录信息
stat	显示文件详细信息
rmdir	删除空目录
cp ， rm ， mv	依次为复制，删除，移动
ln	创建硬链接或符号链接
chmod	更改文件权限以及更改文件所有者
chown	更改文件所有者

cp,rm,rmdir,mv,stat

cp tmp/a tmp/b	复制tmp/a文件，到tmp目录，命名为b
cp c/fst.c tmp/	复制c/fst.c文件到tmp目录，名称不变
cp –R c/ tmp/	递归复制目录c到tmp
rmdir tmp	删除tmp目录，目录必须为空
rm c/test.c	删除c/test.c文件
rm -rf tmp	强制删除tmp目录，目录不为空也可以删除
mv c/fst bin/	移动c/fst文件到bin目录
mv c/fst1 bin/fst	移动c/fst1文件到bin目录，命名为fst
stat tmp/run	显示文件详细信息

列出文件权限等信息

- shell中输入ls -l

```
670453 drwxrwxr-x  3 wy wy 4096 2月  26 18:01 c
680019 drwxrwxr-x  2 wy wy 4096 3月  30 15:21 caclt
655371 drwxrwxr-x  2 wy wy 4096 4月  20 15:16 def
655372 -rw-rw-r--  1 wy wy  392 4月  22 21:47 iore.c
676246 drwxrwxr-x  3 wy wy 4096 3月   9 18:06 node
```

- 第一项表示文件对应的inode号；第二项d表示目录，l表示链接文件，-是普通文件，r，w，x分别表示可读，可写，可执行。连续三个分别表示文件所属用户具有的权限，文件所属组具有的权限，其他用户具有的权限。-表示没有权限。
- 第三项是文件的硬链接数。第四，五项是文件所属用户和文件所属组。第六项是文件大小，字节为单位。接下来是创建时间，文件名。

文件权限与标志位

- r：可读
- w：可写，可以更改文件/目录的内容，可以删除文件/目录。
- x：可执行，程序要具有可执行权限。目录必须要有可执行权限才可以进入。
- 八进制采用三个位表示，r，w，x占有的位分别为：

  r ：100； w：010； x：001

用户	用户组	其他用户
r w x	r - x	r - x
1 1 1	1 0 1	1 0 1

八进制表示：755

文件默认权限

- 系统创建文件时是有一个默认权限的，通过使用权限掩码进行默认权限的设置。
- 使用umask命令可以查看/设置权限掩码：

  umask　　显示权限掩码

  umask 022 设置权限掩码

- 系统不允许在创建一个文件时就赋予它执行权限，必须在创建后用chmod命令增加这一权限；但是目录则允许设置执行权限。

- 默认权限计算规则：用777按位减去掩码中的相应位，并且文件还要减去可执行位。

更改文件权限

- 使用chmod命令改变文件与目录的权限。
- 使用示例：

chmod 755 bin/pse	rwxr-xr-x
chmod +x bin/pse	添加可执行权限，所属用户与用户组具备可执行权限
chmod -w bin/pse	去掉写权限，用户，用户组，其他用户都会去掉写权限
chmod u=rwx,g=rx,o=r bin/pse	相当于chmod 754 bin/pse

更改文件所属用户与所属组

- 使用chown更改文件所属用户和用户组。
- 示例：

chown oklinux:oklinux hd1	更改hd1文件所属用户为oklinux，所属用户组为oklinux
chown :brave hd1	更改hd1文件所属用户组
chown oklinux: hd1	更改文件所属用户
chown oklinux:oklinux tmp/ -R	递归更改所有文件/目录的用户以及用户组

硬链接（hard link）

- ln [TARGET] [LINK NAME]　默认创建硬连接。
- 示例：ln $PWD/hd1 hd2 会在当前目录创建文件的硬链接hd2。
- 在执行连接之前，存放连接的目录中不能有与链接名同名的文件。如果创建硬连接，则TARGET文件必须存在，并且不能是目录。

- 硬链接并没有建立新文件。相当于文件有一个别名，多个文件名使用一个inode，增加了文件的硬链接计数。rm删除文件会减少硬链接计数，计数为0才会从文件系统中删除。
- inode 号仅在各文件系统下是唯一的，当 Linux 挂载多个文件系统后将出现 inode 号重复的现象。所以创建硬链接不能跨文件系统也不能跨分区。

软链接/符号链接（soft link / symbolic link）

- 符号链接类似于Windows上的快捷方式。
- ln –s [TARGET] [LINK NAME]。用ln -s命令建立符号连接时，TARGET最好用绝对路径。
- 示例：ln -s /bin/date $HOME/bin/t。在主目录下的bin目录创建符号链接t指向/bin/date。

- 创建符号链接就会创建一个文件，此文件记录的是另一个文件的路径。删除源文件或目录，只删除了数据，不会删除软链接。一旦以同样文件名创建了源文件，连接将继续指向该文件。

- 符号链接的大小是其指向文件名称的字节数。
- 符号链接可以跨分区跨文件系统，在实际使用中，符号链接很普遍。

本节课任务

- 在当前用户主目录创建目录：stu
- 使用vim创建文件 stu/a.sh，并写入以下内容：

```
#!/bin/bash
echo ‘Hello, this is my first shell program’
echo ‘Your system info:’
uname -a
```

- 给stu/a.sh文件加入可执行权限，并运行stu/a.sh
- 用户主目录创建bin
- 对stu/a.sh创建符号链接：bin/fi
- 运行 source .profile
- 运行fi