第4章 文件安全与文件共享

信息安全

信息安全：指是信息网络的硬件、软件及其系统中的数据受到保护，不受偶然的或者恶意的原因而遭到破坏、更改、泄露，系统连续可靠正常地运行，信息服务不中断。

- 不安全因素：
  - 网络设备：Cisco
  - CPU：intel
  - 操作系统：Windows、Mac OS 、iOS、Android
  - 数据库系统：Oracle、SQL-Server、DB2
  - 浏览器：IE、chrome、firefox
  - 邮件系统：outlook
  - 智能手机APP
  - 其他：QQ、网银、微信、木马、……

本章内容

- 描述Linux提供的三种保护和安全机制
- 描述Linux文件用户的类型
- 解释Linux中关于文件访问权限/特权的概念
- 讨论用户如何为文件确定访问特权，描述用户如何设置、更改文件的权限
- 讨论LINUX实现文件共享所采用的方法和命令
- 详细介绍LINUX硬链接和软（符号）链接并探讨它们的优缺点
- 命令：umask 、 chmod 、ln –f、ln –s和symlinks

概述

- Linux提供的三层次文件保护机制:
  1. 使用 登录名（login name）和登录密码（password）
  2. 文件加密
  3. 文件访问特权（File access privileges）
- 本章着重介绍第三种方法。

基于密码的保护

- 每个Linux系统的用户需要有一个由系统管理员来分配username 和 password
- 得到用户密码的三种方法:
  - （1）用户（密码的拥有者）告知其他用户自己的密码;
  - （2）（破解者）猜测用户的密码;
  - （3）用暴力破解的手段来侦测用户密码。

基于文件加密的保护

- 用户使用工具软件将文件的原始内容转换为另一种完全不同的形式
- 转换后的文件称为被加密文件，这个转换过程则称为文件加密
- 使用同样的软件可以把被加密的文件转换为原始的文件，这个过程称为文件解密。

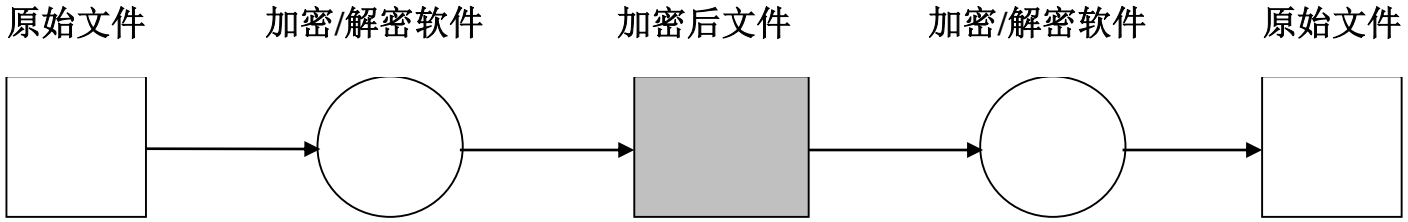


图8.1 文件加密和解密的过程

基于访问权限的文件保护

- 建立防止未授权的用户访问其他用户的文件机制
- 文件的所有者设置特定的访问权限来限制哪些用户可以对该文件进行何种操作
- 策略:
  - 用户分类
  - 访问权限分类
  - 文件操作分类

用户分类

- 每个用户属于某一个组（group）
- 系统中的所有用户组的信息以及该组的用户都记录在/etc/group文件中。
- 文件：所有者（ owner users ）、组（ group users）、其他人（ Other users ）
- Linux有一个特殊用户，称为超级用户或根用户（Superuser or root user）可以访问所有文件。
  - 用户名：root
  - 用户ID：0

基于文件操作/访问权限的分类

- Linux系统中，文件有三种访问权限：
  - 读read (r) :允许读某个文件
  - 写write(w) :允许写、修改和删除某个文件
  - 执行execute (x) :允许执行(run)某个文件
- 对于目录：
  - Read (r) :允许用户列出目录的内容，即可以使用ls命令来列出这个目录下的所有内容
  - Write (w)： 允许用户在目录下建立新文件，删除子目录和文件
  - Execute (x) :允许用户搜索这个目录，如果你没有对目录的执行特权，那么就不能使用ls –l命令来列出目录下的内容或者是使用cd命令来把该目录变成当前目录。

基于文件操作/访问权限的分类(cont)

- 3种用户和3种访问权限：

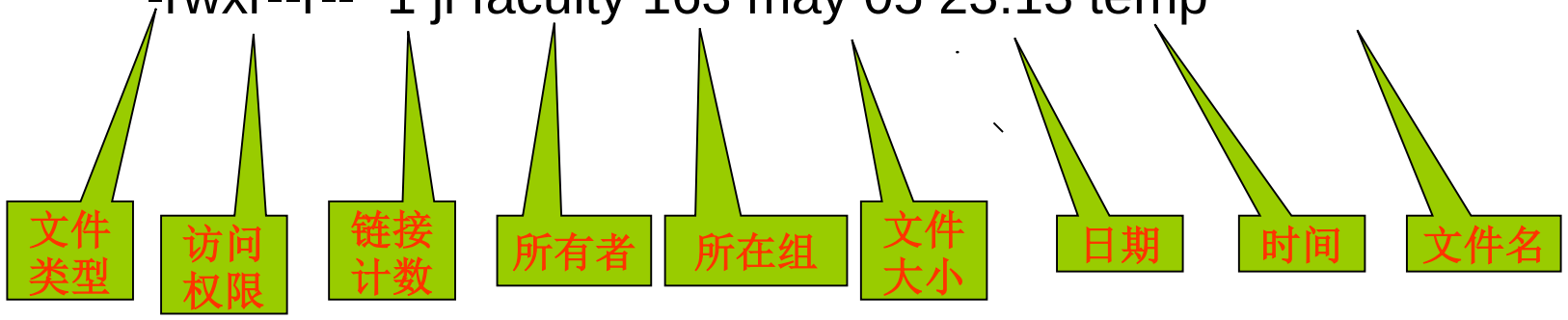
User Type	Permission Type Read (r)	Write (w)	Execute (x)
User (u)	X	X	X
Group (g)	X	X	X
Others (o)	X	X	X

- 用1 bit表示每一种权限，文件用户有8种可能的操作权限
- 3 bits表示某一用户的权限；3种用户，用9 bits来表示。

表			访问特权值列表	
R	W	x	十进制值	含义
0	0	0	0	没有任何访问特权
0	0	1	1	只允许执行
0	1	0	2	只允许写
0	1	1	3	允许写和执行
1	0	0	4	只允许读
1	0	1	5	允许读和执行
1	1	0	6	允许读、写
1	1	1	7	允许读、写和执行

读取和更改文件的访问特权

- 读取文件的访问特权
  - 用 ls -l 或ls -ld命令显示文件的访问权限：



语法：	**ls –l [文件列表]**	
	ls –ld [目录列表]	
目的：	第一个命令的语法：	在屏幕上显示文件和目录的列表；如果参数“文件列表”中包含目录则列出该目录下的文件。
	第二个命令的语法：	显示参数“目录列表”中指定目录下的所有目录列表。
输出：	输出指定目录下的文件或目录的列表。	

读取和更改文件的访问特权(cont)

```
$ ls -l
drwxr-x---    2    sarwar    faculty    512    Apr 23 09:37    courses
-rwxrwxrwx    1    sarwar    faculty     12    May 01 13:22    labs
-rwxr--r--    1    sarwar    faculty    163    May 05 23:13    temp
$
```

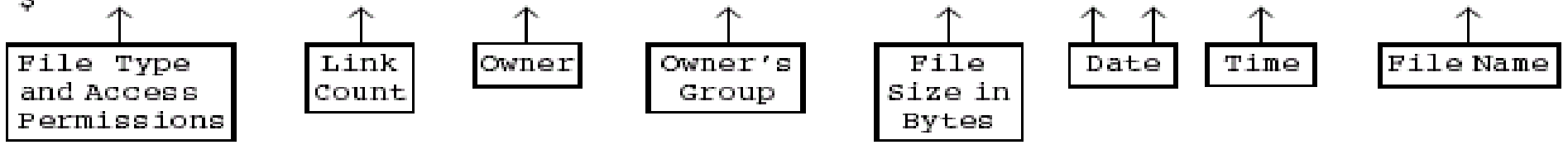


表	访问目录“courses”、“labs”和“temp”的权限设定		
文件名	用户	访问权限	
		组	其他用户
courses	读、写和检索	读和检索	没有特权
labs	读、写和执行	读、写和执行	读、写和执行
temp	读、写和执行	读	读

改变文件的访问特权(cont)

- 使用chmod命令改变文件的访问权限，格式：

语法：	**chmod [options] octal-mode file-list**	
	chmod [options] symbolic-mode file-list	
目的：	改变或设置参数file-list中的文件的访问特权	
常用参数：		
	-R	递归修改或设置文件、目录及其子目录的访问特权。
	-f	强制改变文件访问特权；如果是文件的拥有者，则得不到任何错误信息

  - octal-mode：8进制模式
  - symbolic-mode：符号模式

改变文件的访问特权(cont)

- “symbolic-mode”，格式为<who><operator><privilege>，其中who、operator和privilege的可能取值如表所示。

表	Values for Symbolic Mode Components	
Who	**Operator**	**Privilege**
u User	+ 增加特权	r “读”位（bit）
g Group	- 删除特权	w “写”位
o Other	= 设置特权	x “执行/搜索”位
a All		u 用户当前的访问特权
ugo All		g 组当前的访问特权
		o 其他用户的当前访问特权
		l 锁定特权位
		s 设定用户或组的ID位
		t 粘滞位

改变文件的访问特权(cont)

- chmod 命令符号模式示例:

表	chmod命令示例
命令	目的
chmod u=rwx courses	设定courses目录的拥有者有读、写和搜索的访问特权，同时保证组和其他用户的原有访问特权不变。
chmod ugo-rw sample	任何用户对目录sample没有读和写的访问特权。
chmod a-rw sample	同上
chmod a+x sample	任何用户都拥有对目录sample的执行访问特权。
chmod g=u sample	组内用户对目录sample的访问特权，等于文件拥有者的访问特权。
chmod go= sample	取消组和其他用户对sample所拥有的任何访问特权。

改变文件的访问特权(cont)

- chmod 命令8进制模式示例:

Command	Purpose
chmod 700 *	文件拥有者对当前目录下所有文件和子目录拥有读、写和执行的特权，其他用户没有任何特权
chmod 740 courses	对courses的拥有者设定读、写和执行的特权，对组设定读访问特权，其他用户没有任何特权
chmod 751 ~/courses	对~/courses的拥有者设定读、写和执行的特权，对组设定读和搜索访问特权，其他用户只有搜索特权
chmod 700 ~	对主（登录）目录的拥有者设定读、写和执行的特权，对组和其他用户没有任何特权

改变文件的访问特权(cont)

```
$ cd
$ ls -l
drwxr-x---   2   sarwar   faculty    512   Apr 23 09:37   courses
-rwxrwxrwx   1   sarwar   faculty     12   May 01 13:22   labs
-rwxr--r--   1   sarwar   faculty    163   May 05 23:13   temp
$ chmod 700 courses
$ ls -ld courses
drwx------   2   sarwar   faculty    512   Apr 23 09:37   courses
$ chmod g+rx courses
$ ls -ld courses
drwxr-x---   2   sarwar   faculty    512   Apr 23 09:37   courses
$
$ chmod o+r courses
$ ls -ld courses
drwxr-xr--   2   sarwar   faculty    512   Apr 23 09:37   courses
$ chmod a-w *
$ ls -l
dr-xr-x---   2   sarwar   faculty    512   Apr 23 09:37   courses
-r-xr-xr-x   1   sarwar   faculty     12   May 01 13:22   labs
-r-xr--r---  1   sarwar   faculty    163   May 05 23:13   temp
$ chmod 700 [l-t]*
$ ls -l
dr-xr-x---   2   sarwar   faculty    512   Apr 23 09:37   courses
-rwx------   1   sarwar   faculty     12   May 01 13:22   labs
-rwx------   1   sarwar   faculty    163   May 05 23:13   temp
$
```

改变文件的访问特权(cont)

- chmod 的-R命令递归更改指定目录下所有子目录和目录内的文件的访问特权。

```
$  chmod  -R  711  courses
$  chmod  -R  700  ~/personal/letters
$
```

- chmod命令后设置访问的参数只有1位或2位8进制数，按从右到左匹配。

```
$ chmod 7 courses
$ chmod 70 personal
$ ls -l
d------rwx    2 sarwar   faculty   512 Nov 10 09:43 courses
d---rwx---    2 sarwar   faculty   512 Nov 10 09:43 personal
drw-------    2 sarwar   faculty   512 Nov 10 09:43 sample
$
```

改变文件的访问特权(cont)

- 对目录设置权限：读特权对目录而言意味着可以读出目录的内容，写特权意味着可以在目录下创建或删除一个文件，执行特权意味着可以检索这个目录。

```
$ chmod 600 sample
$ chmod 500 courses
$ chmod 300 personal
$ ls -l
dr-x------   2   sarwar   faculty   512 Nov 10 09:43 courses
d-wx------   2   sarwar   faculty   512 Nov 10 09:43 personal
drw-------   2   sarwar   faculty   512 Nov 10 09:43 sample
$ mkdir courses/ee345
mkdir: Failed to make directory "courses/ee345"; Permission denied
$ cp foo courses
cp: cannot create courses/foo: Permission denied
$ cd sample
sample: Permission denied
$ ls -l personal
personal unreadable
$
```

改变文件的访问特权(cont)

```
$ ls -ld dir1
d-w-------   2   msarwar faculty   512 Oct 22 12:13 dir1
$ cp prog1.cpp dir1
cp: cannot create dir2/prog1.cpp: Permission denied
$ rm dir2/f1
dir2/f1: Permission denied
$ chmod u+x dir2
$ ls -ld dir2
d-wx------   2   msarwar faculty   512 Oct 22 12:13 dir2
$ rm dir2/f1
$
```

默认的文件访问特权

- 当创建一个文件或目录的时候，LINUX系统根据umask命令的参数来设定新创建文件或目录的访问特权，
- umask [mask] :新创建的文件或目录的访问特权都将设置为1，除了在参数mask中设置为1的对应位。
- 命令umask的参数是位的掩码（bit mask），用八进制表示。掩码位为1表示新创建的文件相应的访问特权应该被关闭。
- umask命令不带任何参数，这显示当前设置的俺码。
- 新建文件的访问特权用如下的公式计算：
  - 文件访问权限=默认的访问权限— mask
- 默认的访问权限:
  - 执行文件为777
  - 文本(text)文件为666

默认的文件访问特权（续）

- 如果使用命令umask 013并运行成功：
  - 对于一个新创建的可执行文件的访问特权就是764（777-013）
  - 每个新创建的可执行文件或目录就拥有缺省的访问特权rwxrw-r--
- 如果掩码被设置成777：
  - 所有新创建的文件和目录就没有任何的访问特权，因为所有的位都已经被umask设置了，
  - 新创建的可执行文件和目录的访问特权被设置为000（777-777）
- 访问特权的掩码常常设置为022
  - 即所有新创建的可执行文件和目录的访问特权为755（777-022）
  - 对于新创建的文本文件，则应该是644（666-022）

默认的文件访问特权（续）

- 例:
  - umask 022
  - touch foo
  - umask 077
  - touch bar
  - mkdir foobar
  - ls –l foo bar foobar

    -rw-r--r-- 1 ji faculty 0 Nov 5 16:16 foo

    -rw------- 1 ji faculty 0 Nov 5 16:16 bar

    drwx------ 2 ji faculty 512 Nov 5 16:16 foobar

特殊访问位

- 允许普通用户运行某个可执行文件，其权限是文件拥有者的权限，通过设置有效用户标识(Effective user id)实现
- 三个特殊而重要的位：
  - set-user-ID(SUID)
  - set-group-ID(SGID)
  - sticky

特殊访问位（续）

setting the SUID bit:

- 文件/etc/passwd，只有超级用户有权访问该文件。用户执行passwd命令，试图去更改/etc/passwd文件来写入新的登录密码，但此时命令却没有对/etc/passwd文件的访问特权。怎样能让普通用户通过执行passwd命令来修改他们的登录密码，同时又不能随意修改文件/etc/passwd破坏其它用户信息的完整性。
- 每个linux文件都有一个附加的保护位（SUID），如果对一个可执行文件设置了这个标志位，那么该可执行文件就以这个文件的拥有者的权限运行。如命令：passwd、lp、mail、mv、ps
- SUID设置方法：
  - chmod 4xxx *file-list* (in octal mode, xxx is the permissions of ugo)
  - chmod u+s *file-list* (symbolic mode)

特殊访问位（续）

- 当设置SUID位为1时，如果用户对该文件有执行权限，那么执行位被设置位‘s’，否则执行位变为‘S’
- Examples:

```
$ ls -l cp.new
-rwx--x---    1    ji    faculty  12  May    08  20:00    cp.new
$ chmod 4710 cp.new
$ ls -l cp.new
-rws--x---    1    ji    faculty  12  May    08  20:00    cp.new
$ chmod u-s cp.new
$ chmod u-x cp.new
$ ls -l cp.new
-rw---x---    1    ji    faculty  12  May    08  20:00    cp.new
$ chmod u+s cp.new
$ ls -l cp.new
-rwS--x---    1    ji    faculty  12  May    08  20:00    cp.new
$
```

特殊访问位（续）

Set-Group-ID (SGID)

- chmod 2xxx file-list
- chmod g+s file-list
- 当SGID位被置位1的时候，如果组对应的可执行位是x，那么则改为小写的‘s’，否则被改为大写的‘S’。
- Examples:

```
$ ls -l cp.new
-rwxr-x--x  1    ji    faculty    12    May 08    20:00    cp.new
$ chmod 2751 cp.new
$ ls -l cp.new
-rwxr-s--x  1    ji    faculty    12    May 08    20:00    cp.new
$ chmod g-s cp.new
$ chmod g-x cp.new
$ ls -l cp.new
-rwxr----x  1    ji    faculty    12    May 08    20:00    cp.new
$ chmod g+s cp.new
-rwxr-S--x  1    ji    faculty    12    May 08    20:00    cp.new
```

特殊访问位（续）

sticky Bit

- 如果某用户对某个目录有写权限，该用户可能会删除这个目录下的文件
- sticky位被设置，可以保证只有文件拥有者可以删除或重命名某个目录下的文件，即使其他用户有写权限也不能。
- 共享目录中使用
- 设置sticky位的方法：
  - chmod 1xxx file-list
  - chmod +t file-list

特殊访问位（续）

- 如果sticky位为1，并且其他用户对目录有可执行的权限，那么该权限位变为小写的‘t’，如果没有可执行的权限，那么该权限位就变为大写的‘T’。
- Examples:

```
$ chmod 1775 testSticky
$ ls -l
total 1
drwxrw-r-t      2 ji faculty          512 Oct  28 12:24 testSticky
$ chmod 760 testSticky
$ chmod +t testSticky
$ ls -l
total 1
drwxrw---T      2 ji faculty          512 Oct  28 12:24 testSticky
```

文件访问权限和类型

- 文件的访问特权和类型信息都存储在一个16位的空间中，低9位用来存储文件访问特权，接下来的3位用来存储特别的权限位，高4位用来保存文件类型
- u16 i-mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				SUID	SGID	Sticky	r	w	x	r	w	x	r	w	x
文件类型位				特别访问位			拥有者特权位			组特权位			其他用户特权位		

图　Linux ext2/ext3文件系统中文件类型位和访问特权位的位置

chattr命令

- 通过chattr设置文件的属性。

  格式：chattr [-RV] [-+=AacDdijsSu] [-v version] files...

  主要参数或选项：

  - -R：递归处理，将指定目录下的所有文件及子目录一并处理
  - + ：在原有参数设定基础上，追加参数。
  - - ：在原有参数设定基础上，移除参数。
  - = ：更新为指定参数设定。
  - A：文件或目录的 atime (access time)不可被修改。
  - a：只能向文件中添加数据，而不能删除。
  - i：设定文件不能被删除、改名、设定链接关系，同时不能写入或新增内容。i参数对于文件 系统的安全设置有很大帮助。
  - s：保密性地删除文件或目录，即硬盘空间被全部收回。
  - u：与s相反，当设定为u时，数据内容其实还存在磁盘中，可以用于undeletion.

chattr命令

- 例：

```
# touch chattr_test
# chattr +i chattr_test
# rm chattr_test
rm: remove write-protected regular empty file `chattr_test`? y
rm: cannot remove `chattr_test`: Operation not permitted
```

  - 此时连root本身都不能直接进行删除操作，必须先去除i设置后再删除。

lsattr命令

- 查看文件属性

  格式：lsattr [-RVadlv] [files...]

  参数或选项说明：

  - -R：递归列示目录及文件属性。
  - -a：显示所有文件属性，包括隐藏文件(.)、当时目录(./)及上层目录(../)。
  - -d：仅列示目录属性。

文件共享

文件共享的方法

- 通过副本共享：
  - 复制需要共享的文件并分发到组内每个成员
- 通过同一用户名登录共享：
  - 共用一个账号登录系统。
- 为共享文件设立适当的访问权限：
  - 把所有要共享的文件放到一个成员帐号下，设置组成员具有读写和执行的权限
- 为团队成员建立一个用户组
  - 建立一个新用户组只包括项目组的所有成员，每个用户用自己的账号登录，为自己的文件设立适当的访问权限，使得它们可以被组中的其他成员访问。
- 通过文件链接共享， Linux两类链接：
  - 硬链接（hard link）
  - 软（符号）链接（Soft/symbolic link）

通过文件链接共享

- ln命令用来建立硬链接和符号链接。
- 语法：

  ln [options] existing-file new-file

  ln [options] existing-file-list directory

  - 常用选项：

    -f 强迫建立链接

    -n 如果“new-file”已存在，不创建链接。

    -s 建立一个符号链接而不是硬链接

    -d 建立目录的硬链接

- 例：

  $ ln Chapter3 Chapter3.hard

硬链接

- 硬链接是一个指向文件索引节点的指针。
- ln 并不会影响文件的内容，它只是建立另一个文件名称而已
- 例:

```
$ ls -il test1
2513974 -rw-r--r--  1 root root 556 Jul 12 21:06 test1
$ ln test1 test2
$ ls -il test*
2513974 -rw-r--r--  2 root root 556 Jul 12 21:06 test1
2513974 -rw-r--r--  2 root root 556 Jul 12 21:06 test2⇦建立新链接

$ ln test1 test2
$ cat test1
        Welcome to Linux World
$ cat test2
        Welcome to Linux World
```

目录项、索引结点、文件内容关系

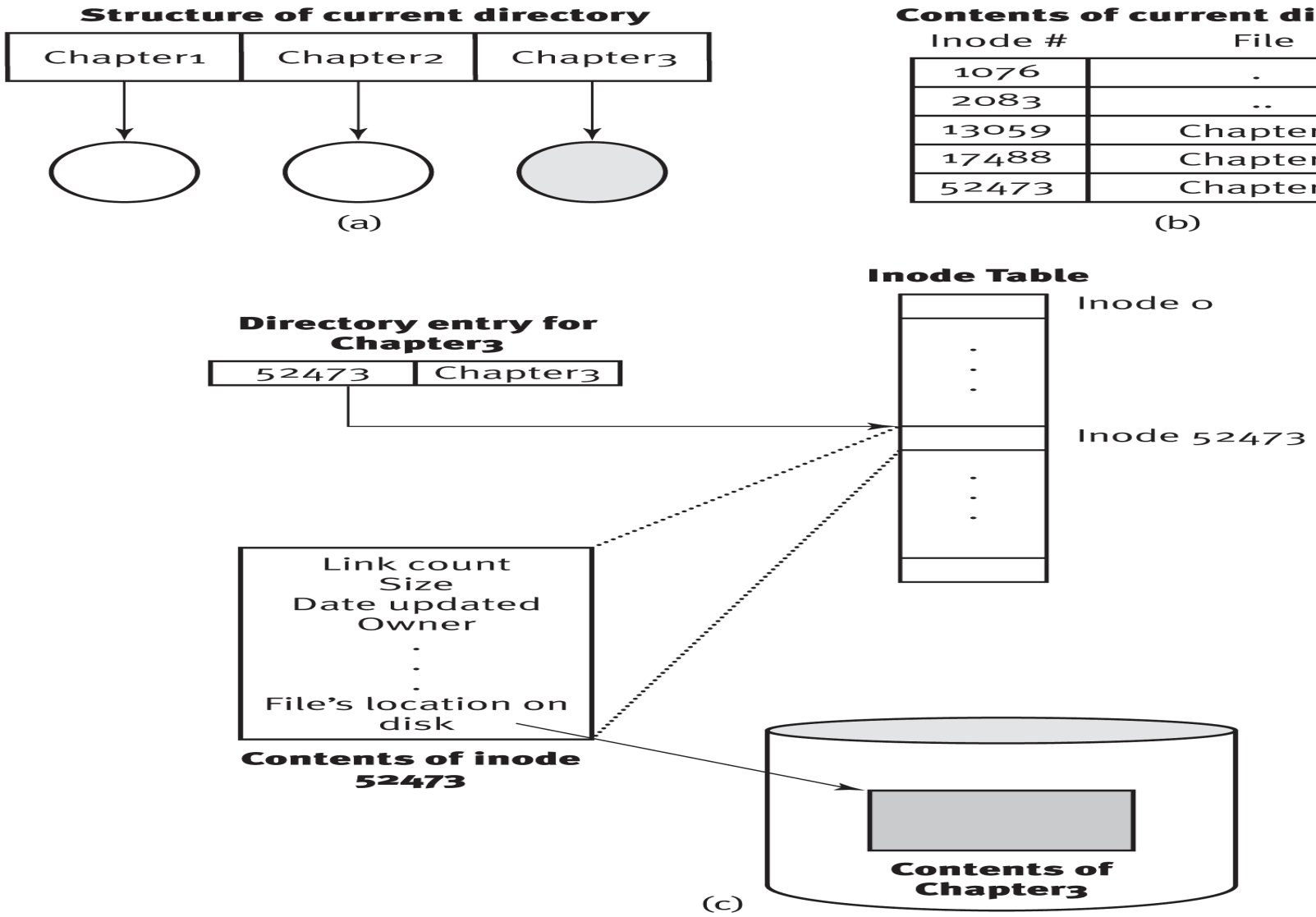
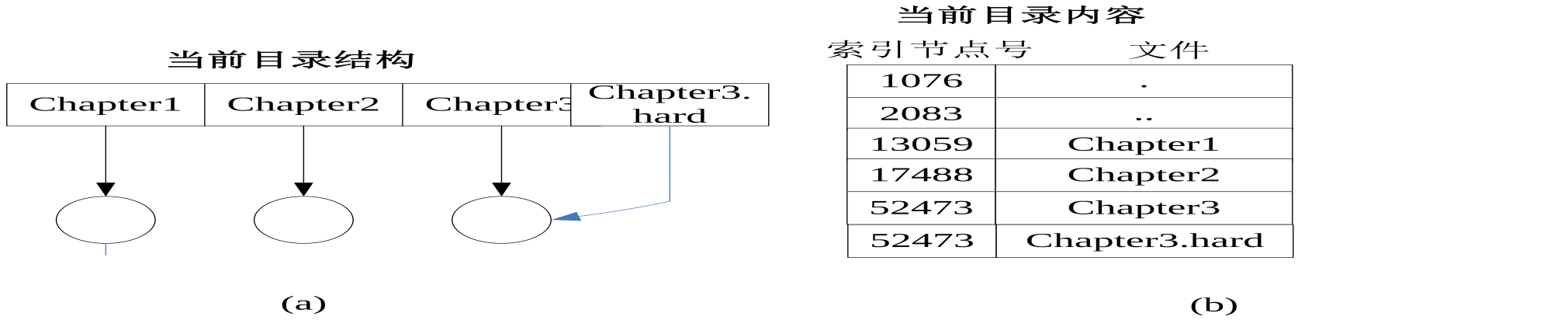
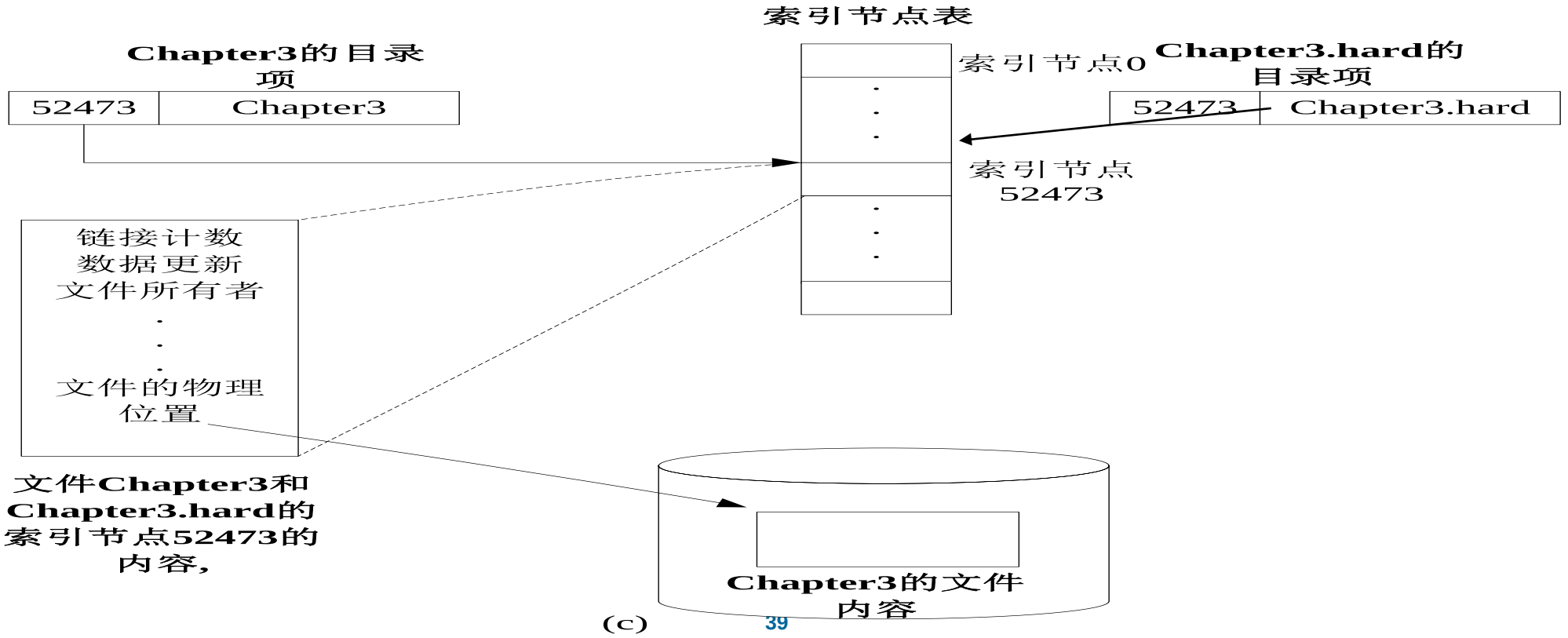


Figure 11.1 (a) Logical structure of current directory; (b) contents of current directory; and (c) relationship among a directory entry, inode, and file contents

In Chapter3 Chapter3.hard





硬链接

- 当前目录下创建一个名为memo6.hard的硬链接，指向文件~/memos/memo6
  - ln ~/memos/memo6 memo6.hard
  - 如下图
- 为~/linuxbook/examples/dir1目录下的所有文件创建链接
  - $ ln –f ~/linuxbook/examples/dir1/* ~/linuxbook/example/dir2
  - $ ls –l dir1
  - $ ls –l dir2

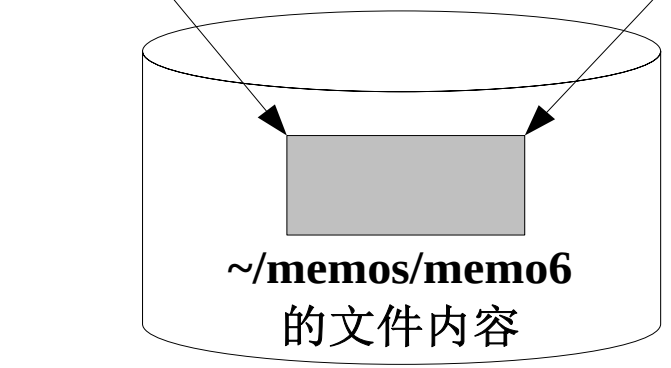
硬链接

当前目录内容

索引节点号	文件
1076	.
2083	..
13059	Chapter1
17488	Chapter2
52473	Chapter3
83476	memo6.hard

目录~/memos内容

	文件
1076	.
2083	..
83468	memo1
...	...
83476	memo6
...	...



硬链接

- 在主目录下创建一个/users/ji/unixbook/examples/demo1链接
  - $ ln –f ~/users/ji/unixbook/examples/demo1 ~

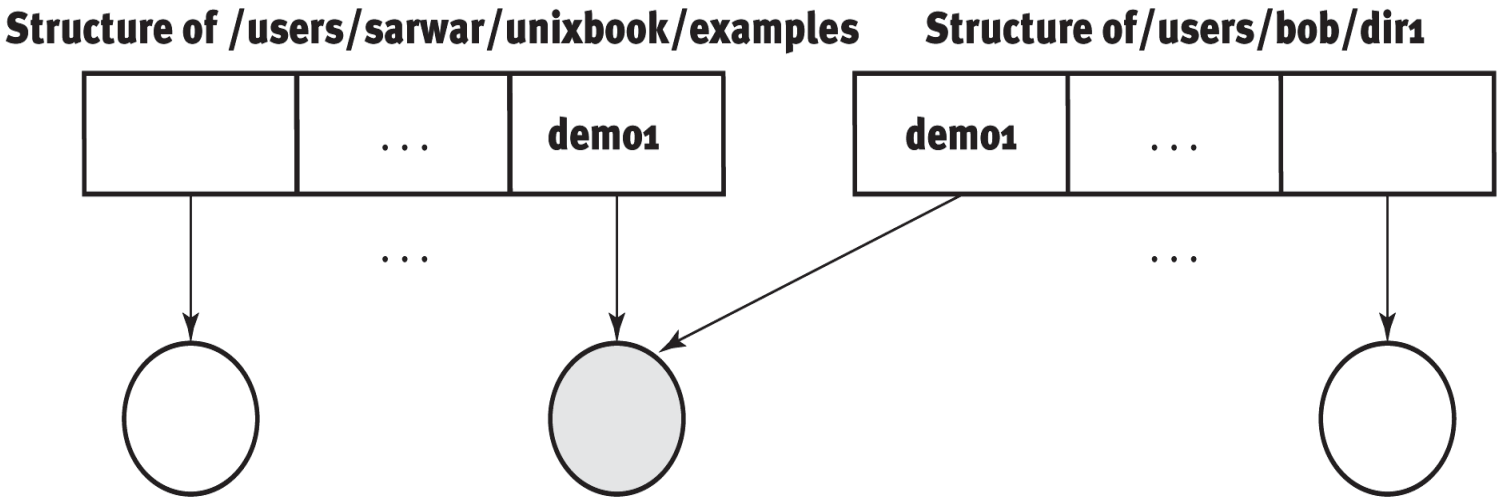


Figure 11.4 A hard link between /users/sarwar/unixbook/examples/demo1 and /users/bob/dir1

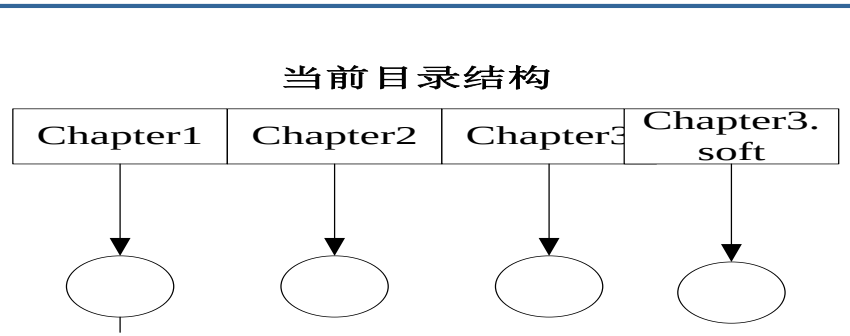
硬链接

- Hard Links特点：
  - 不可跨越文件系统
  - 只有超级用户才可以建立目录硬链接Only superuser can create hard links to directories
  - 不占用空间(极少)

符号链接

- 系统为共享的用户创建一个link类型的新文件，将这新文件登记在该用户共享目录项中，这个link型文件包含连接文件的路径名。该类文件在用ls命令长列表显示时，文件类型为l。
- 当用户要访问共享文件且要读link型新文件时，操作系统根据link文件类型性质将文件内容作为路径名去访问真正的共享文件。
- ln –s Chapter3 Chapter3.soft
- readlink命令用来获取一个符号链接指向的目标文件的路径。
  - 当一个软链接指向的是一个另外的软链接，而另外一个软链接又指向其他的目标。 这时可以使用-f选项直接获取最终的非软链接的目标。

符号链接

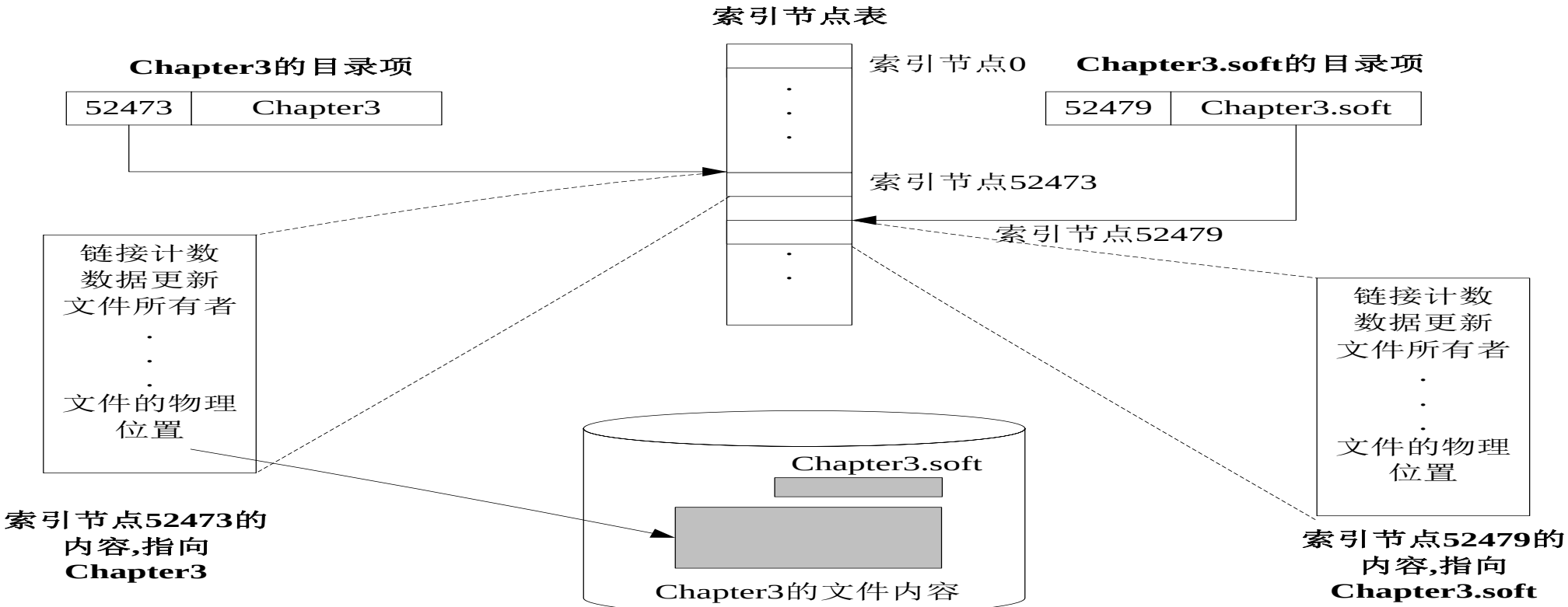


(a)

当前目录内容

索引节点号	文件
1076	.
2083	..
13059	Chapter1
17488	Chapter2
52473	Chapter3
52479	Chapter3.soft

(b)

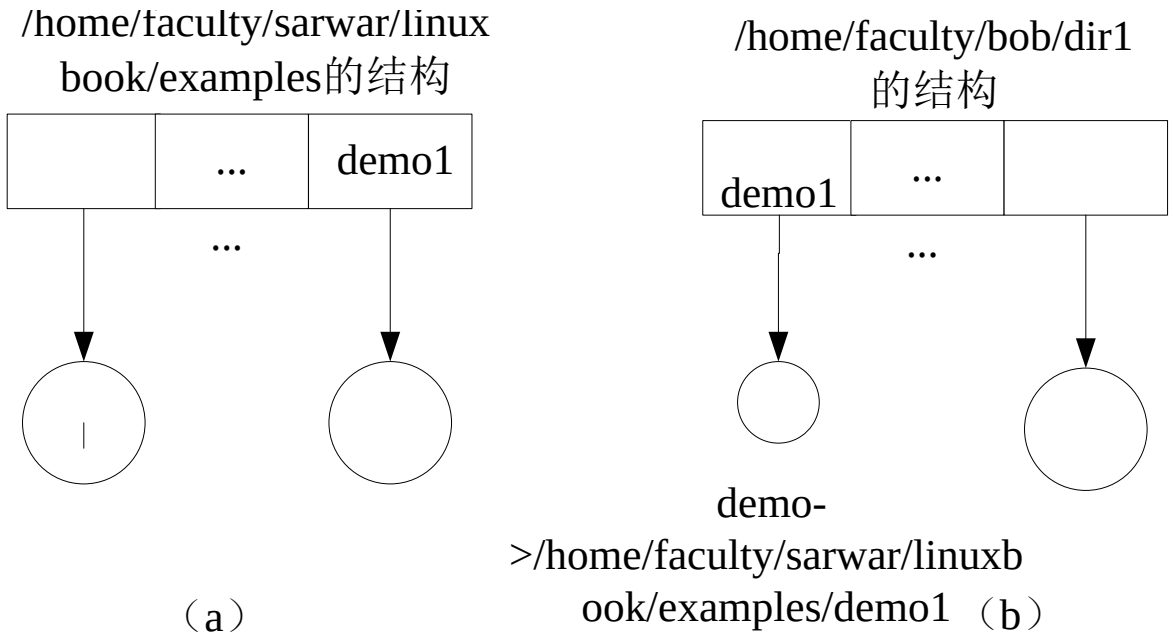


(c)

符号链接

- 用户ji可以运行下面命令在bob家目录下的dir1子目录里建立名为demo1的软链接指向/home/faculty/ji/linuxbook/examples/demo1文件。

ln –sf /home/faculty/ji/linuxbook/examples/demo1 ~



符号链接特点

- 可跨越文件系统，甚至跨越网络(NFS)
- 如果链接指向的文件从一个目录移动到另一个目录，就无法通过符号链接访问它
- 占有少量空间，存inode 的信息

小结

- 本章需要掌握：
  - 命令：chmod、ln
  - 知识：文件权限、用户类型、文件共享，硬链接，符号连接
- 本章需要了解：
  - 命令：umask、chattr、 lsattr

End of chpater