
Linux fayl tizimida huquqlarni boshqarish

ass: Fayziyev V

Linux fayl tizimida huquqlarni boshqarish

1. Linux fayl tizimini tavsiflanishi
 2. Umumiy fayl tizimi vazifalarini bajarish
 3. Disk bo'limlarini boshqarish
 4. Olinadigan vositadan foydalaning
 5. Ma'lumotlarni zaxiralash
-

1. Linux fayl tizimini tavsiflanishi

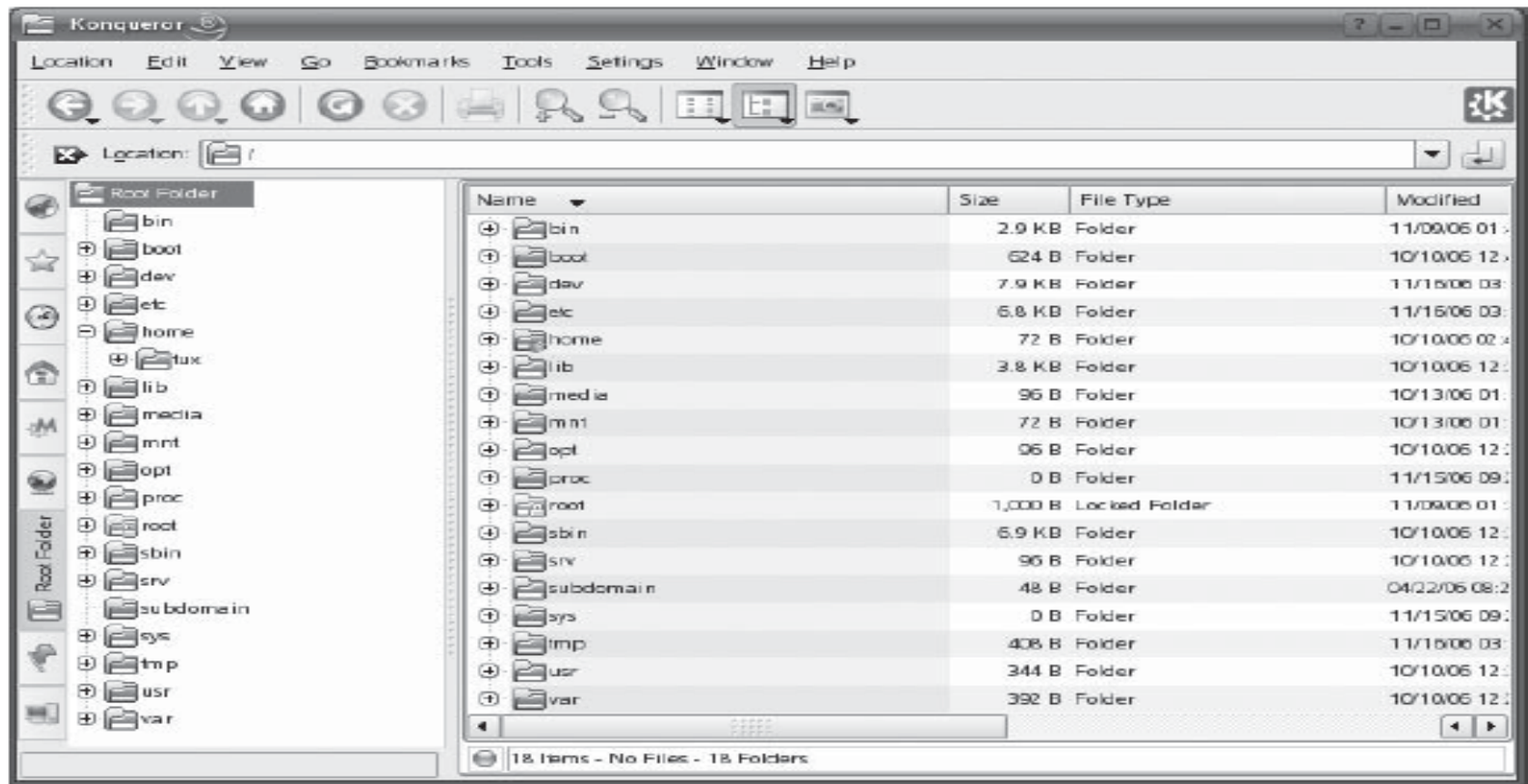
- Linux fayl tizimining roli
 - Linux fayl tizimining ierarxik tuzilishi
 - Linux tomonidan ishlatiladigan fayl turlari
-

1. Umumiy fayl tizimi vazifalarini bajarish

- Ma'lumotlar tartiblangan va osongina joylashishi mumkin Ma'lumotni istalgan keyingi vaqtda osongina olish mumkin
 - Ma'lumotlarning yaxlitligi saqlanadi
-

• Linux fayl tizimining ierarxik tuzilishi

FIGURE 6-1 The hierarchical structure of the Linux file system



Notice that the topmost directory in the structure is the / directory, also called the root directory.

- Ildiz katalog ostida bir qator kichik kataloglar joylashgan. Ushbu kataloglarni qanday nomlash bo'yicha texnik xususiyatlar Fayl tizimi ierarxiyasi standartida (FHS) mavjud.
-

/bin

Ushbu katalog Linux tizimini boshqarish va ishga tushirish uchun zarur bo'lgan bajariladigan fayllarni, shu jumladan qobiqlarni (masalan, bash) va cp va rm kabi fayl tizimini boshqarish vositalarini o'z ichiga oladi.

/boot

Ushbu katalog tizimingizni yuklash uchun
zarur bo'lgan yuklash fayllarini o'z ichiga
oladi

/dev

- Ushbu katalog tizimda o'rnatilgan turli xil apparat qurilmalarini ko'rsatish uchun ishlatiladigan maxsus fayllarni o'z ichiga oladi
 - Misol: hda, hdb ...
-

/etc

- Ushbu katalog tizim tomonidan ishlatiladigan matnga asoslangan konfiguratsiya fayllarini hamda tizimda ishlaydigan xizmatlarni o'z ichiga oladi. Linux qanday ishlashini sozlash uchun ushbu fayllarni matn muharriri yordamida tahrirlashingiz mumkin
-

/home

Ushbu katalog Linux tizimingizdagi har bir foydalanuvchi hisobi uchun uy katalogi bo'lib xizmat qiluvchi pastki kataloglarni o'z ichiga oladi.

/lib

- Ushbu katalogda /bin va /sbin dasturlari tomonidan ishlatiladigan kod kutubxonalarini mavjud. Yadro modullaringiz /lib modullari pastki katalogida ham saqlanadi.
-

/media

- Ushbu katalog ba'zi Linux distribyutorlari (masalan, SUSE Linux) tomonidan tashqi qurilmalarni, jumladan CD drayvlar, DVD drayvlar va floppi drayvlarni o'rnatish uchun ishlatiladi.
-

/mnt

Ushbu katalog ba'zi Linux distribyutorlari (masalan, Fedora yoki Red Hat) tomonidan CD drayvlar, DVD drayvlar va floppi drayvlar kabi tashqi qurilmalarni o'rnatish uchun ishlatiladi.

/opt

- Ushbu katalog tizimga o'rnatgan ba'zi dasturlar uchun fayllarni o'z ichiga oladi

/proc

- Ushbu katalog ushbu ro'yxatdagi boshqa kataloglardan biroz farq qiladi.
 - /proc aslida fayl tizimida mavjud emas
 - /proc ichida bir qancha turli pastki kataloglar mavjud, E'tibor bering, bu pastki kataloglarning har biri nom bilan emas, raqam bilan aniqlangan. Bu raqamlar tizimda ishlaydigan bog'langan jarayonning jarayon identifikatori (PID) raqamiga mos keladi
-

/root

- Bu katalog asosiy foydalanuvchining uy katalogidir. E'tibor bering, u /home-dagi boshqa foydalanuvchilar uchun uy kataloglaridan alohida joylashgan.
-

/sbin

- Bu katalog asosiy foydalanuvchining uy katalogidir. E'tibor bering, u /home-dagi boshqa foydalanuvchilar uchun uy kataloglaridan alohida joylashgan.
-

/srv

- Ushbu katalog tizimda ishlaydigan xizmatlar (masalan, httpd va ftpd) o'z fayllarini saqlaydigan pastki kataloglarni o'z ichiga oladi.

/sys

- Ushbu katalog tizimingizdagi apparat haqida ma'lumotni o'z ichiga oladi.

/tmp

- Ushbu katalog siz yoki tizim tomonidan yaratilgan vaqtinchalik fayllarni o'z ichiga oladi.

/usr

- Ushbu katalog ilova fayllarini o'z ichiga oladi. Haqiqatan ham, tizimingizda ishlatiladigan dastur fayllarining aksariyati /usr pastki katalogida saqlanadi
-

/var

- Ushbu katalog ilova fayllarini o'z ichiga oladi. Haqiqatan ham, tizimingizda ishlatiladigan dastur fayllarining aksariyati /usr pastki katalogida saqlanadi
-

• Types of Files Used by Linux

- Windows fayl tizimida fayl tizimida asosan ikkita kirish turi mavjud: kataloglar va fayllar. Sizda oddiy fayllar, yashirin fayllar, yorliq fayllari, so'zni qayta ishlash fayllari, bajariladigan fayllar va boshqalar bo'lishi mumkin.
 - Linux bilan esa fayl tizimi tomonidan ishlatiladigan turli xil fayl turlari mavjud.
-

Linux File turlari

- Oddiy fayllar: Ushbu fayllar boshqa operatsion tizimlarning fayl tizimlari tomonidan ishlatiladigan fayllarga o'xshaydi - masalan, bajariladigan fayllar, OpenOffice.org fayllari, rasmlar, matn konfiguratsiya fayllari va boshqalar.
 - Bog'lanishlar: Bu fayllar fayl tizimidagi boshqa fayllarga ishora qiluvchi ko'rsatgichlardir.
-

Linux File turlari

- FIFOs: FIFO stands for First In First Out. These are special files used to move data from one running process on the system to another. A FIFO file is basically a queue where the first chunk of data added to the queue is the first chunk of data removed from the queue. Data can only move in one direction through a FIFO.
- Sockets: Sockets are similar to FIFOs in that they are used to transfer information between sockets. With a socket, however, data can move bi-directionally.

2. Complete Common File System Tasks

- Navigating the file system
 - Managing files and directories
 - Running executable files
 - Searching the file system
-

With Linux, however, you should become proficient with the shell commands first and then use the GUI utilities for convenience

- **Reasons?**

- Most employers and co-workers won't take you seriously as a Linux administrator if you can't use the shell prompt. It just goes with the territory.
-

-
- Many Linux systems, especially those deployed as servers, don't run X Windows. Supporting a GUI environment requires a lot of CPU overhead. Many server admins prefer to devote those CPU cycles to system services instead of moving the mouse cursor on the screen. In this situation, you need to know how to do things from the shell prompt.
 - You need to know how to complete these tasks from the shell prompt to pass your Linux+ exam.
-

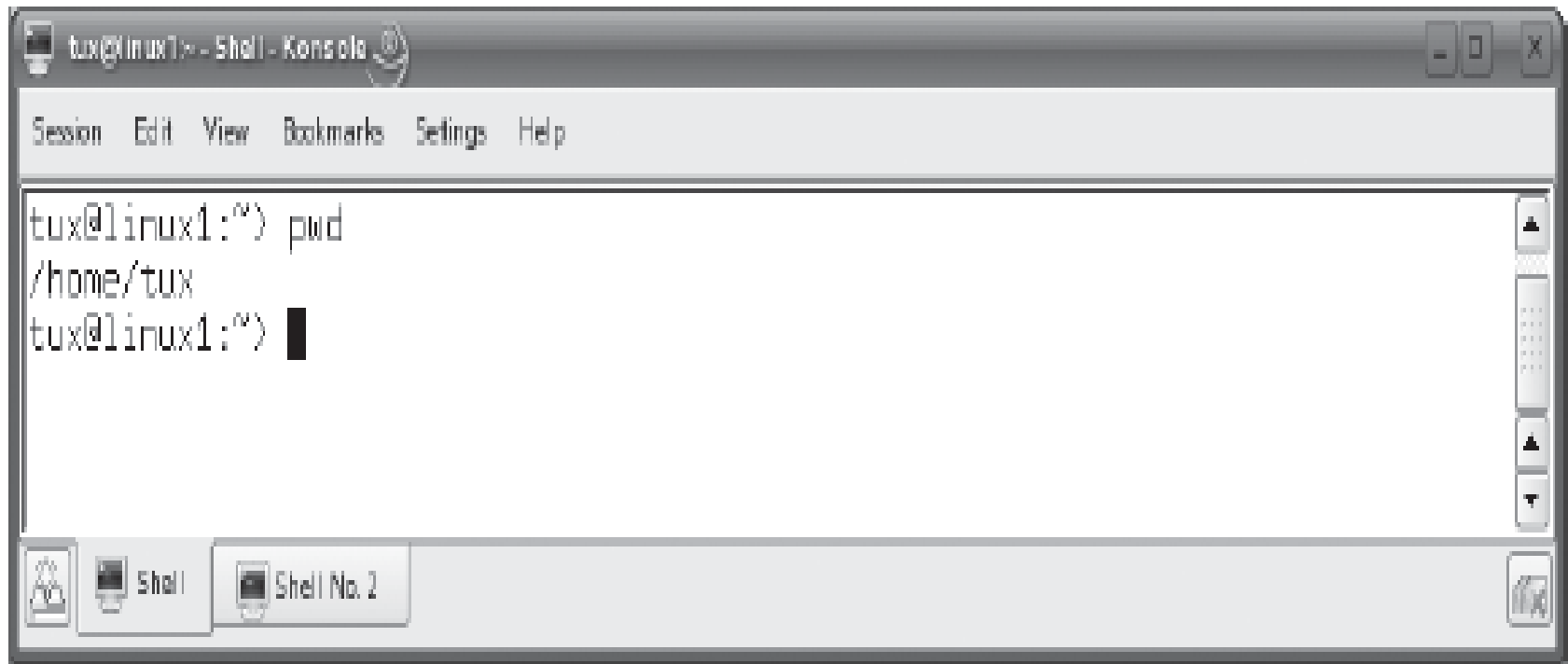
• Navigating the File System

- Pwd (Present Working Directory)
 - cd
 - ls
-

Using the pwd Command

FIGURE 6-8

Using pwd to display the current directory



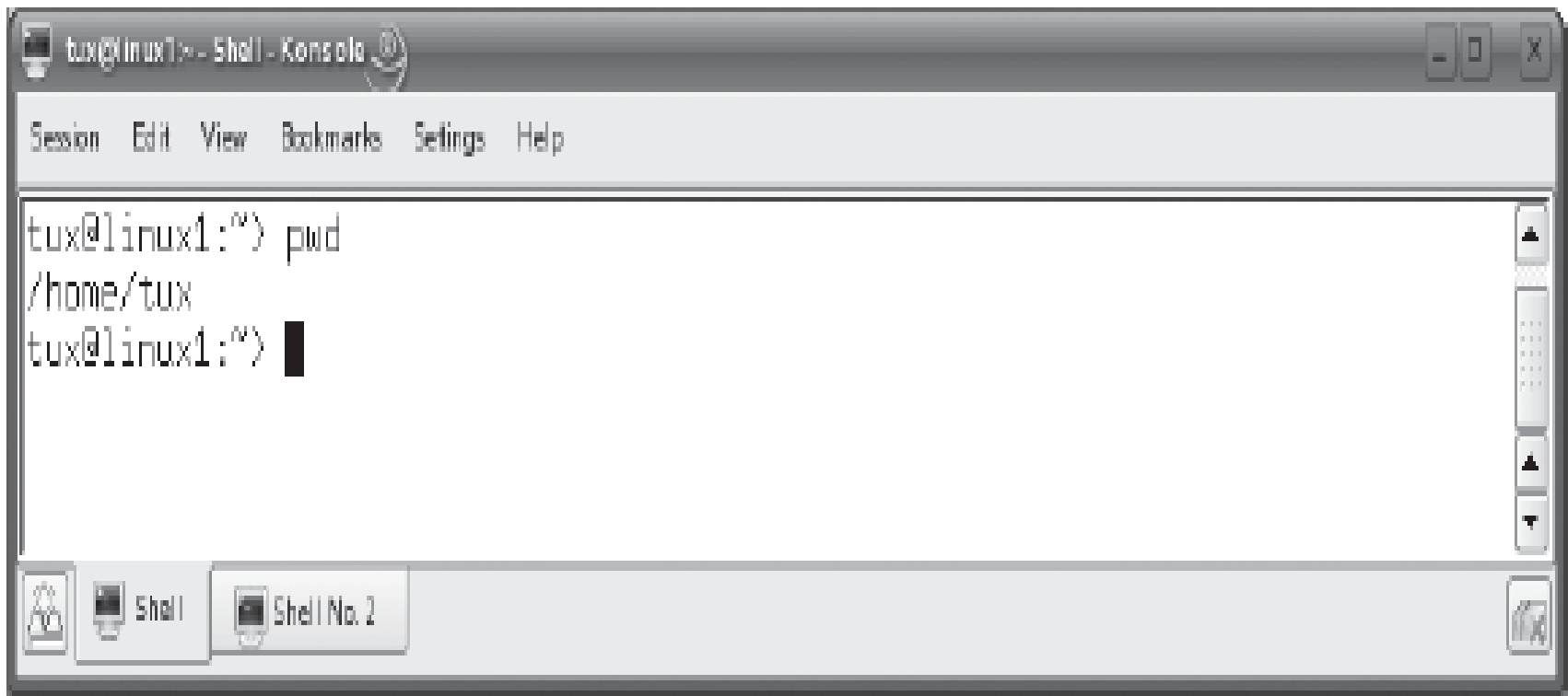
The image shows a terminal window titled "tux@linux1 ~ - Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the user "tux" at "linux1" in the root directory "~" typing the command "pwd". The output is "/home/tux". The prompt "tux@linux1:~" is followed by a cursor. The window has standard Linux window controls (minimize, maximize, close) in the top right and a taskbar at the bottom with icons for "Shell" and "Shell No. 2".

```
tux@linux1:~) pwd
/home/tux
tux@linux1:~) █
```

Using the cd Command

FIGURE 6-8

Using pwd to display the current directory



The image shows a terminal window titled "tux@linux1:~ - Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the user typing "pwd" and receiving the output "/home/tux". The prompt "tux@linux1:~" is followed by a cursor. At the bottom of the window, there are tabs for "Shell" and "Shell No. 2".

```
tux@linux1:~) pwd
/home/tux
tux@linux1:~) █
```

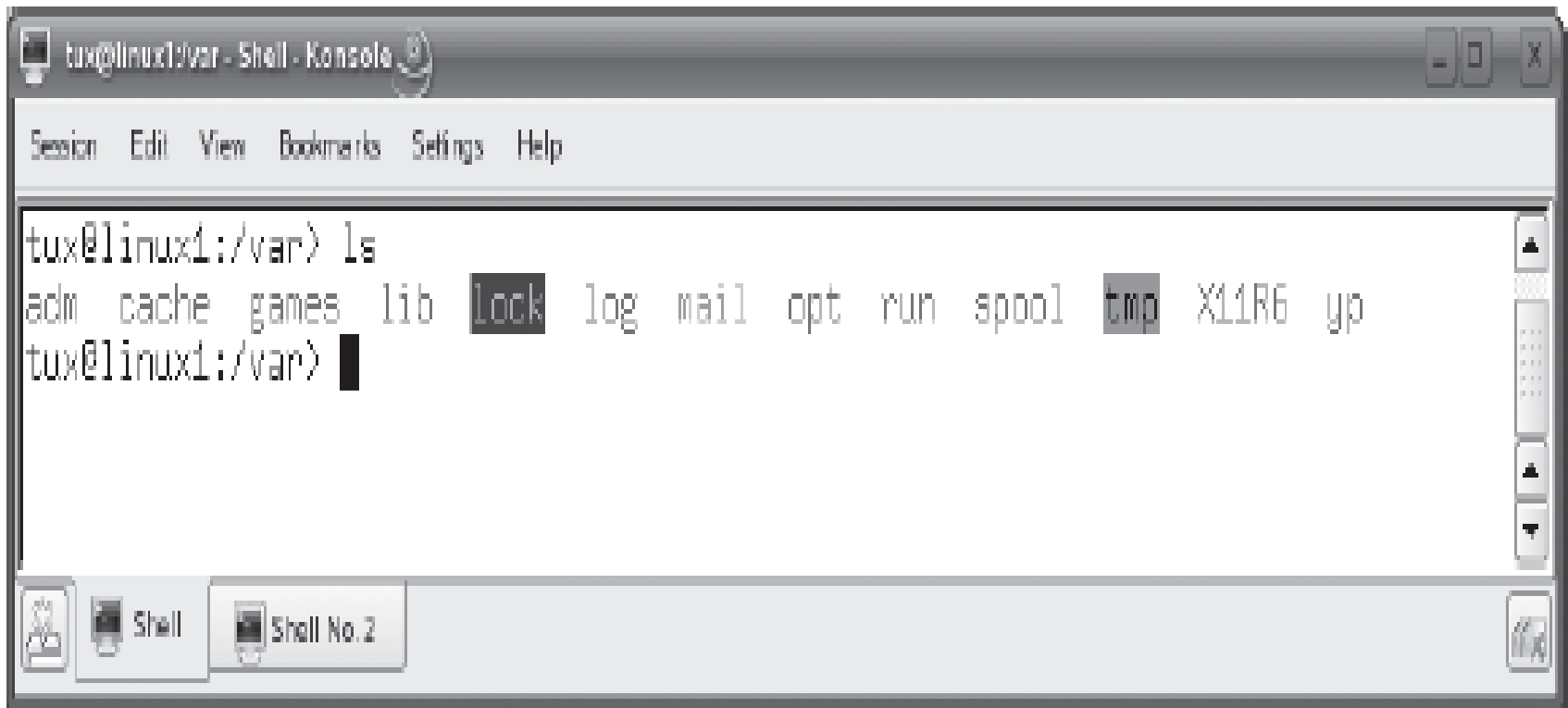
cd

- Agar siz yo'lni ko'rsatmasdan qobiq satriga CD ni kiritsangiz, u avtomatik ravishda kataloglarni hozirda tizimga kirgan foydalanuvchining uy katalogiga o'zgartiradi.
-

Is buyrug'idan foydalanish

FIGURE 6-12

Using `ls` to view the contents of the current directory



The image shows a terminal window titled "tux@linux1:/var - Shell - Konsole". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content shows the user typing the command `ls` at the prompt `tux@linux1:/var>`. The output of the command is a list of directories: `adm cache games lib lock log mail opt run spool tmp X11R6 yp`. The words `lock` and `tmp` are highlighted with a dark background. Below the terminal output, there are two tabs labeled "Shell" and "Shell No. 2".

```
tux@linux1:/var> ls
adm cache games lib lock log mail opt run spool tmp X11R6 yp
tux@linux1:/var>
```

ls

- -a Barcha fayllarni, jumladan yashirin fayllarni ko'rsatadi.
- -l Katalog tarkibining uzun ro'yxatini ko'rsatadi. Bu juda foydali variant. Undan fayl nomlari, egalik huquqi, ruxsatlar, o'zgartirish sanasi va hajmini ko'rish uchun foydalanishingiz mumkin.
- -R Katalog tarkibini rekursiv ravishda aks ettiradi; ya'ni joriy katalogning mazmuni bilan bir qatorda barcha pastki kataloglarning mazmunini ko'rsatadi.

• Fayllar va kataloglarni boshqarish

- Fayllar va kataloglarni yaratish
 - Fayl tarkibini ko'rish
 - Fayllar va kataloglarni o'chirish
 - Fayllar va kataloglarni nusxalash va ko'chirish
 - Havolalar yaratish
-

Fayllar va kataloglarni yaratish

`touch new_file`

Fayllar va kataloglarni yaratish

`mkdir new_directory`

Matn fayli tarkibini ko'rish

- cat
 - less
 - head
 - tail
-

cat

- Mushuk fayl nomi buyrug'i ko'rsatilgan matn faylini ekranda ko'rsatadi. Bu buyruq chiqishni to'xtatmaydi, shuning uchun agar siz undan uzun faylni ko'rish uchun foydalansangiz, sahifani bir marta to'xtatib turish uchun buyruqqa |ko'proq qo'shishingiz kerak bo'lishi mumkin.
-

less

- The less filename command can also be used to display the specified text file on screen, much like cat. However, the less command automatically pauses a long text file one page at time.
-

head

- The head filename command is used to display the first couple of lines of a text file on the screen.
-

tail

- The tail filename command is used to display the last couple of lines of a text file on screen. The tail command is particularly useful when displaying a log file on screen. When viewing a log file, you probably only want to see the end of the file.
 - The tail command also includes the -f option, which is very useful. You can use this to monitor the file specified in the command
-

Deleting Files and Directories

- **rmmdir**
- **rm**

rmmdir

- This utility can be used to delete an existing directory. To use it, simply enter `rmmdir directory_name`—for example, `rmmdir MyFiles`. Be aware, however, that `rmmdir` requires that the directory be empty before it will delete it.
-

rm

- The rm utility is a more powerful deletion utility that can be used to delete either a file or a populated directory. To delete a file, simply enter `rm filename`. To delete a directory, enter `rm -r directory_name`.
-

rm

- Be careful with rm! By default, it won't prompt you to confirm a deletion operation. It assumes that you really meant to delete the file or directory. If you want rm to prompt you before deleting a file or directory, include the `-i` option.
-

Copying and Moving Files and Directories

- cp
- mv

cp

- This utility is used to copy files or entire directory structures from one location in the file system to another. For example, to copy a file named /tmp/schedule.txt to your home directory, you could enter

```
cp /tmp/schedule.txt ~.
```

mv

- The mv command is used much like cp. However, it will copy the specified file to the new location in the file system and then delete the original. For example, to move a file named mylog.txt from /tmp to /var/log, you would enter `mv /tmp/mylog.txt /var/log`
-

mv

- The mv command is also used to rename files. Simply enter mv followed by the file to be renamed and then the new file name. For example, to rename schedule.txt to schedule.old, you would enter

mv schedule.txt schedule.old.

Creating Links

- Hard
- Symbolic

Hard

- A hard link is a file that points directly to the inode of another file. An inode stores basic information about a file in the Linux file system, including its size, device, owner, and permissions. Because the two files use the same inode, you can't tell which file is the pointer and which is the pointee after the hard link is created.
-

Symbolic

- A symbolic link file also points to another file in the file system. However, a file that is a symbolic link has its own inode. Because the pointer file has its own inode, the pointer and the pointee in the file system can be easily identified. For example, in the previous chapter, you saw that the vi file is symbolic-linked to the vim file.
-

syntax

In **pointee_file** **pointer_file**

- Using **ln** without any options creates a hard link
 - If you want to create a symbolic link, you use the **-s** option
-

Running Executables

For example, you've used the `man`, `vi`, and `cp` programs to accomplish various tasks on your system. These programs are executable files that exist in the Linux file system and are loaded into memory when you call them from the shell prompt

`which` command

You can view all of your environment variables by entering `env | more` or `echo $PATH` at the shell prompt.

To add the path to the executable to your PATH environment variable. Enter

```
PATH=$PATH:new_path
```



For example, if you wanted to add a directory named `apps` in your `vmk`'s home directory to the `PATH` variable, you would enter

```
PATH=$PATH:/home/vmk/apps
```

After adding the path, you must then export the new variable by entering `export PATH` at the shell prompt. You can then check your `PATH` variable by entering `echo $PATH` at the shell prompt

- **Searching the File System**

- Using find
- Using locate
- Using grep

Using find

- The find utility is fantastic tool that you can use to search the Linux file system. To use find, enter at the shell prompt

```
find path -name "filename"
```

- For example, suppose you wanted to find all of the log files stored in your file system that have a `.log` extension. You could enter

```
find / -name "*.log"
```

-
- The find utility is flexible. You can also use the `-user "username"` option to search for files owned by a **specific user**, or use the `-size "size"` option to search for files of a specified size.
 - You can use a **+** sign before the size value to search for files **larger than the specified size**, or a **-** sign before the size value to search for files **smaller than the specified size**. The find utility has many other options
-

Using locate

- The locate utility functions in much the same manner as find. However, it has one distinct advantage over find. Whenever you execute a search with find, it manually walks through each directory in the path you name in the command looking for the specified files. This process can take some time.
-

locate

- Alternatively, the locate utility builds an index of the files in the file system. Then, when you execute a search, locate simply runs a query of the index. It doesn't actually search the file system directly. The result is that locate runs much faster than find in most situations. To use locate, you must first install the findutils-locate package on your system.
-

-
- With the index updated, you can search for files by simply entering:

`locate filename`

at the shell prompt. For example, if you wanted to search for a file named `snmpd.conf`, you could enter `locate snmpd.conf`

Using grep

- Linux also provides a utility called grep that you can use to search for content within a file. Using **grep**, you can search through a file for a particular text string. To use grep, you would enter **grep search_text file**. For example, let's suppose you want to want to search through your **/var/log/messages** file for any log entries related to the **VNC** service running on your Linux system. You would enter

```
grep vnc /var/log/messages
```

grep

options:

- **-i** Ignores case when searching for the search text.
 - **-l** Doesn't display the actual matching line of text. Only displays the names of the files that contain the matching text.
 - **-r** Searches recursively through subdirectories of the path specified.
-

3. Manage Disk Partitions

- Using fdisk to create disk partitions
 - Building a file system with mkfs
 - Mounting a partition with mount
 - Checking the file system with fsck
-

- **Using fdisk to Create Disk Partitions**

The **fdisk** utility is used from the command line to create or delete partitions at the shell prompt. To use fdisk to create a new partition, first open a terminal session. Then, at the shell prompt, change to your root account by entering `su` – followed by your root user's password.

fdisk

At the shell prompt, enter `fdisk` device. For example, if you wanted to create a partition on the third SCSI hard disk in your system, you would enter

```
fdisk /dev/sdc
```



YaST2@linux1

Partition your hard disks...

This is intended for **experts**. If you are not familiar with the concepts of hard disk **partitions** and how to use them, you might want to go back and select **automatic** partitioning.

Nothing will be written to your hard disk until you confirm all your changes with the "Apply" button. Until that point, you can safely abort.

For LVM setup, using a non-LVM root device and a non-LVM swap device is recommended. Other than the root and swap devices, you should have partitions managed by LVM.

The table to the right shows the current partitions on all your hard disks.

Hard disks are designated like this

`/dev/hda` 1st EIDE disk /`dev/hdb` 2nd EIDE disk /`dev/hdc` 3rd EIDE disk

etc.

- or -

`/dev/sda` 1st SCSI disk /`dev/sdb` 2nd SCSI disk /`dev/sdc` 3rd SCSI disk

etc.

This notation always refers to the entire disk.

Partitions are designated like this:

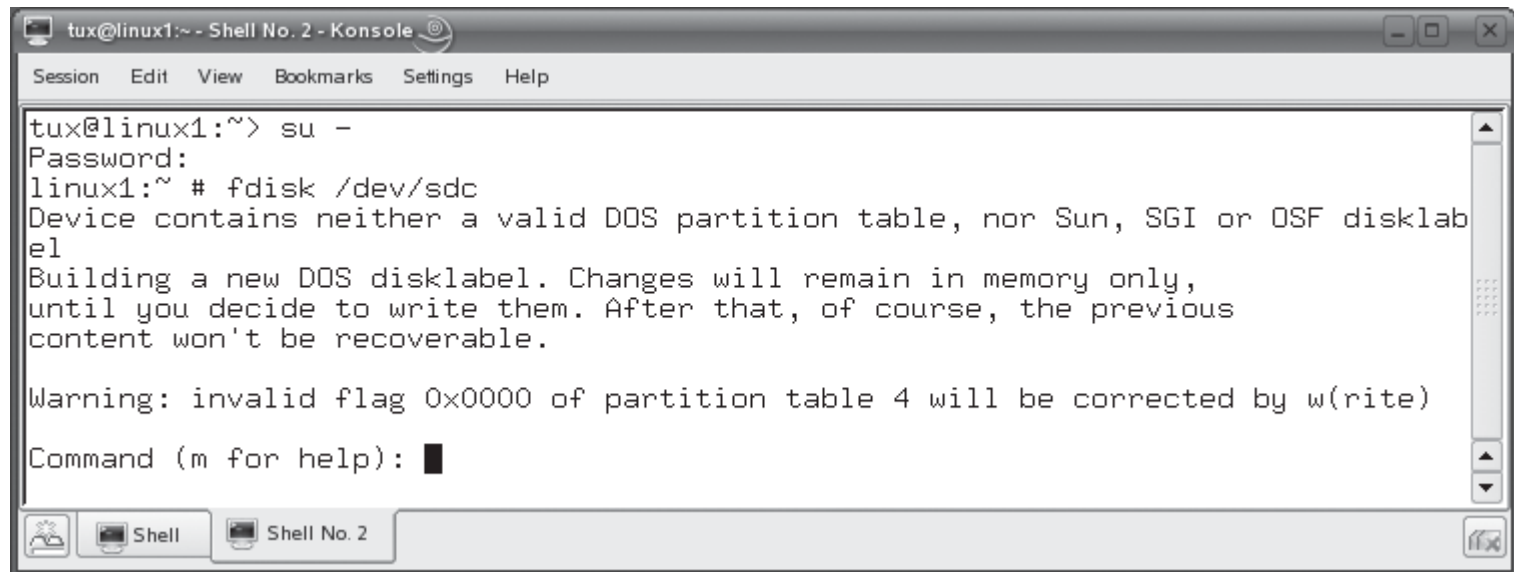
Expert Partitioner

Device	Size	F	Type	Mount	Mount By	Start	End	Used By	Label
/dev/sda	8.0 GB		VMware,-VMware Virtual S			0	1043		
/dev/sda1	203.9 MB		Linux native	/boot	K	0	25		
/dev/sda2	517.7 MB		Linux swap	swap	K	26	91		
/dev/sda3	6.0 GB		Linux native	/	K	92	875		
/dev/sda4	1.2 GB		Linux native	/var	K	876	1043		
/dev/sdb	4.0 GB		VMware,-VMware Virtual S			0	521		
/dev/sdb1	3.9 GB		Linux native	/media/usb	K	0	520		

Buttons: Create, Edit, Delete, Resize, LVM..., RAID..., Crypt File..., Expert., Quit, Apply

Taskbar: 1 tux@linux1 SUSE Linux YaST2@lin 01:19 pm
2 tux - Konquer YaST Control 11/16/06

With fdisk running, you have a Command: prompt that you can use to enter **fdisk** commands



```
tux@linux1:~> su -  
Password:  
linux1:~ # fdisk /dev/sdc  
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel  
Building a new DOS disklabel. Changes will remain in memory only,  
until you decide to write them. After that, of course, the previous  
content won't be recoverable.  
  
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)  
  
Command (m for help): █
```

Before creating a partition, you should press **P** to view any existing partitions on the disk. This will help you determine if there is sufficient space and, if there is, what number must be assigned to a new partition created on the disk.

To create a new partition, you press **N**. You can then specify whether you want to create a primary disk partition or an extended disk partition.

To create a primary partition, press **P** when prompted.

To create an extended partition, press **E**

For example: You could press **T** and specify a partition ID of **82** if you wanted to change the partition to a swap partition.

-
- **Building a file system with mkfs**
 - **Even though we've created a partition with fdisk, we can't use it yet. That's because it hasn't been formatted with a file system yet. This is accomplished using one of the following commands:**
 - **mkfs**
 - **mkreiserfs**
-

mkfs

- This utility is used to make an **ext2** or **ext3** file system on a partition. You can even use it to create a FAT file system on the partition (which is used by Windows and DOS). You specify which file system you want to use by entering the **-t** option and the type of file system. For example, if you wanted to create an ext3 file system on the first partition on your third SCSI hard disk drive in your system, you would enter **mkfs -t ext3 /dev/sdc1**

mkreiserfs

- This utility is used to make a Reiser file system on a partition. For example, if you wanted to create a Reiser file system on the first partition on your third SCSI hard drive in your system, you would enter

```
mkreiserfs /dev/sdc1
```

When you do, a screen is displayed and a proposal is presented. Many of these parameters can be customized using `mkreiserfs` options.

- **Mounting a Partition with mount**

One of the key differences between Linux and other operating systems (such as Windows) is the fact that you need to mount a file system before you can use it. This is another one of those “gotchas” that trip up new Linux administrators. You even have to mount CDs, DVDs, and floppies before you can use them!

Such as Windows, reference partitions on the disk using a drive letter, such as C: or D ...

Not so with Linux! Linux instead uses a virtual file system (**VFS**) that creates a single hierarchy that encompasses all partitions on all storage devices in the system. Switching to the directory where the partition is mounted switches you to that partition.

mount

To use mount to mount a partition, first switch to your root account using **su**. Then enter

```
mount -t file_system_type device mount_point
```

at the shell prompt. For example,

```
mount -t reiserfs /dev/sdc1 /mnt/extraspace
```

If you don't know what type of file system is used by the partition, you can use the **-a** option with mount instead of **-t**. This will cause mount to try to mount the partition using all supported file system types until one is successful.

umount

If, for some reason, you wanted to unmount a partition, you can use the **umount** command. To unmount a partition, simply enter **umount** followed by the device or the mount directory for the partition. For example,

```
umount /dev/sdc1
```

or

```
umount /mnt/extraspace.
```

You can also use command-line utilities to create and mount swap partitions on your Linux system. You can create swap partitions using the `mkswap` command. First, create the partition that will be used as a swap partition using `fdisk`. Then enter `mkswap` device at the shell prompt. For example, to convert the second partition on the second IDE hard drive in your system to a swap partition, you would enter `mkswap /dev/hdb2` at the shell prompt. After converting the partition to a swap partition, you need to enable it using the `/sbin/swapon` command.

- **Checking the File System with fsck**

Syntax: **fsck** device

For example if you wanted to check the file system on the first partition of the third SCSI hard drive in your system, you would enter **fsck /dev/sdc1**. The utility will then check the file system and report any errors encountered (or the lack thereof)

4. Use Removable Media

- Working with floppies
 - Working with optical devices
 - Working with USB and FireWire devices
-

- Working with Floppies

Syntax: `mount -t vfat device mount_point`

On other distributions, such as Red Hat or Fedora, the default mount point for floppy diskettes is

`/mnt/floppy`

EX: `mount -t vfat /dev/fd0 /media/floppy`

Therefore, you can simply enter `mount /dev/fd0` at the shell prompt and the floppy will be automatically mounted using these parameters.

Before you remove a mounted floppy diskette from a drive, you must also unmount it. Just as with partitions, this is done using the `umount` command. To unmount a floppy, you enter `umount device` or `umount mount_point`. For example, to unmount your **A:** drive, you would enter `umount /dev/fd0`

- **Working with Optical Devices**

- `/media/cdrom`
 - `/media/cdrecorder`
 - `/media/dvd`
-

On distributions such as Red Hat or Fedora, your directories for mounting optical devices are located in `/mnt`.

To mount a CD on a Linux system, insert it in your CD drive. Change to your root account and then enter `mount -t iso9660 device mount_point`. For example, `mount -t iso9660 /dev/hdc /media/cdrom`

As with any mounted file system, you should use `umount` to unmount a CD or DVD before `ejecting` it.

- Working with USB and FireWire Devices

mount **-t** auto **device** mount_point

ex:

mount **-t** **auto** /dev/sdd /media/ THUMBDRIVE

5. Back Up Data

- Selecting a backup medium
 - Selecting a backup strategy
 - Using Linux backup utilities
-

- **Selecting a Backup Medium**

- “old days”: floppies or a tape drive
- Today, USB or FireWire hard drive



- **Selecting a Backup Strategy**

- Selecting a backup type
 - Selecting a backup schedule
 - Determining what to back up
-

Selecting a Backup Type

- Full
 - Incremental: During an incremental backup, only the files that have been modified since the **last backup** (full or incremental) are backed up. After being backed up, each file is flagged as having been backed up.
 - Differential: only the files that have been modified since the last **full backup** are backed up
-

Selecting a Backup Schedule

- You can use whatever backup schedule works best for you. However, most admins work on a weekly rotation, as discussed previously. Pick one day for your full backup and then the remaining days of the week for your incremental or differential backups. You should also schedule your backups to occur when the load on the system is at its lightest. Late in the evening or in the early morning are usually best, depending on your organization's schedule.
-

Determining What to Back Up

- /etc
 - /home
 - /opt
 - /root
 - /var
 - /srv
-

- **Using Linux Backup Utilities**

- Using tar
- Using cpio



Using tar

- The tar utility has been around for a very long time and is a very commonly used Linux backup tool. The acronym “tar” stands for tape archive. The tar utility takes a list of specified files and copies them into a single archive file (.tar). The .tar file can then be compressed with the gzip utility on your Linux system, resulting in a file with a .tar.gz extension.
-

tar

- Syntax: `tar -cvf filename directory`
 - The `-c` option tells tar to create a new archive.
 - The `-v` option tells tar to work in verbose mode, displaying each file being backed up on screen.
 - The `-f` option specifies the name of the tar archive to be created.
 - Ex: `tar -cvf /media/usb/backup.tar /home`
-

-
- To restore a tar archive, simply enter

`tar -xvf filename`

ex: `tar -xvf /media/usb/backup.tar`

- If the archive has been zipped, you can also use the `-z` option to unzip the archive before extracting it
-

Using cpio

- The cpio utility can also be used to make archive files just like tar. A key **difference** between tar and cpio is the fact that you must provide cpio with a **list of files and directories** to back up from the standard input
- Syntax: **cpio -ov file,directory > file.cpio**
- The **-o** option tells cpio to create a new archive. The **-v** option simply tells cpio to run verbosely, displaying the name of each file and directory as it's processed

-
- we could extract the archive we just created by entering `cpio -iv < file.cpio`
-