

# Module 07

## Navigating the Filesystem

# Exam Objective

## 2.3 Using Directories and Listing Files

### Objective Description

Navigation of home and system directories and listing files in various locations.

# Introduction



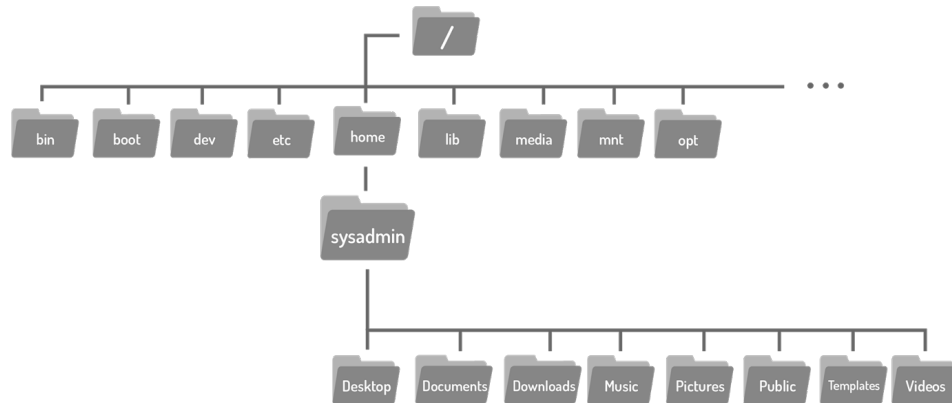
# Introduction

- In Linux, everything is stored in files.
- *Files* are used to store data such as text, graphics, and programs.
- *Directories* are a type of file used to store other files.
- Directories are used to provide a hierarchical organization structure.

# Directory Structure

# Directory Structure

- On a Windows system, the *top level* of the directory structure is called *My Computer*.
- The Linux directory structure, called a *filesystem*, also has a top level called the *root directory* (symbolized by the slash / character).



# Directory Structure

- To view the contents of the root directory, use the `ls` command with the `/` character as the argument:

```
sysadmin@localhost:~$ ls /  
bin    dev    home   lib     media   opt     root   sbin     selinux  sys    usr  
boot   etc    init   lib64   mnt     proc    run    sbin???  srv      tmp    var
```

- Notice that there are many directories with descriptive names including `/boot`, which contains files to boot the computer.

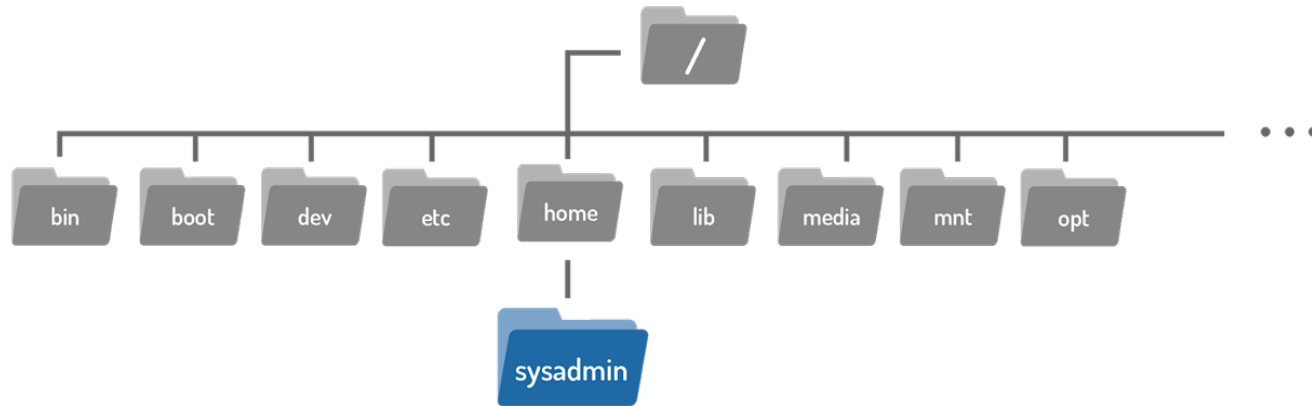
# Home Directory

- On most Linux distributions there is a directory called `home` under the root `/` directory.
- Under this `/home` directory there is a directory for each user on the system.
- When a user opens a shell, they should automatically be placed in their home directory.
  - The user has the full control to create and delete additional files and directories in their home directory.
  - Most other directories in a Linux filesystem are protected with *file permissions*.
- The home directory has a special symbol used to represent it, the tilde `~` character.



# Home Directory

- The directory name is the same as the name of the user.
- So, a user named `sysadmin` would have a *home directory* called `/home/sysadmin`:



# Current Directory

- The `pwd` (*print working directory*) command can be used to determine where the user is currently located within the filesystem.
- The `pwd` command prints the working directory, which is the current location of the user within the filesystem.

```
pwd [OPTIONS]
```

```
sysadmin@localhost:~$ pwd  
/home/sysadmin
```

# Changing Directories

- When a user opens a shell, they typically begin in their home directory.
- To navigate the filesystem, use the `cd` (*change directory*) command.

```
cd [options] [path]
```

- To move from the home directory into the `Documents` directory use the directory name as an argument to the `cd` command:

```
sysadmin@localhost:~$ cd Documents
sysadmin@localhost:~/Documents$
```

- After changing directories, the new location can also be confirmed in the new prompt, again shown in blue.

# Changing Directories

- When used with no arguments, the `cd` command will take the user to their home directory.

```
sysadmin@localhost:~/Documents$ cd  
sysadmin@localhost:~$
```

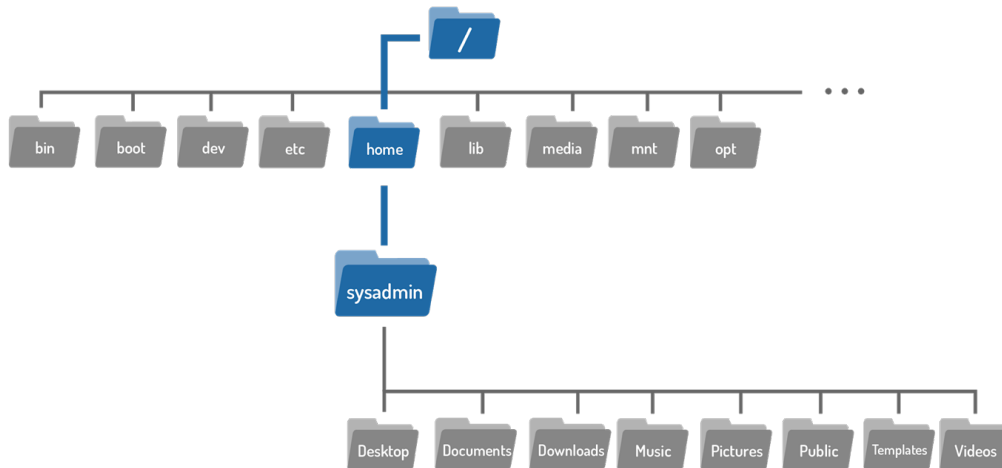
- If the user tries to change to a directory that does not exist, the command returns an error message:

```
sysadmin@localhost:~$ cd Junk  
-bash: cd: Junk: No such file or directory
```

# Paths

# Paths

- A path is a list of directories separated by the / character.
- There are two types of paths: *absolute* and *relative*.
- For example, `/home/sysadmin` is a path to the home directory:



# Absolute Paths

- Absolute paths allow the user to specify the exact location of a directory.
- Absolute paths always starts at the root directory, and therefore it always begins with the / character.
- The path `/home/sysadmin` is an absolute path; it tells the system to:
  - Begin at the root / directory > move into the home directory > then into the sysadmin directory
- If the path `/home/sysadmin` is used as an argument to the `cd` command, it moves the user into the home directory for the sysadmin user.

```
sysadmin@localhost:~/Documents$ cd /home/sysadmin  
sysadmin@localhost:~$
```

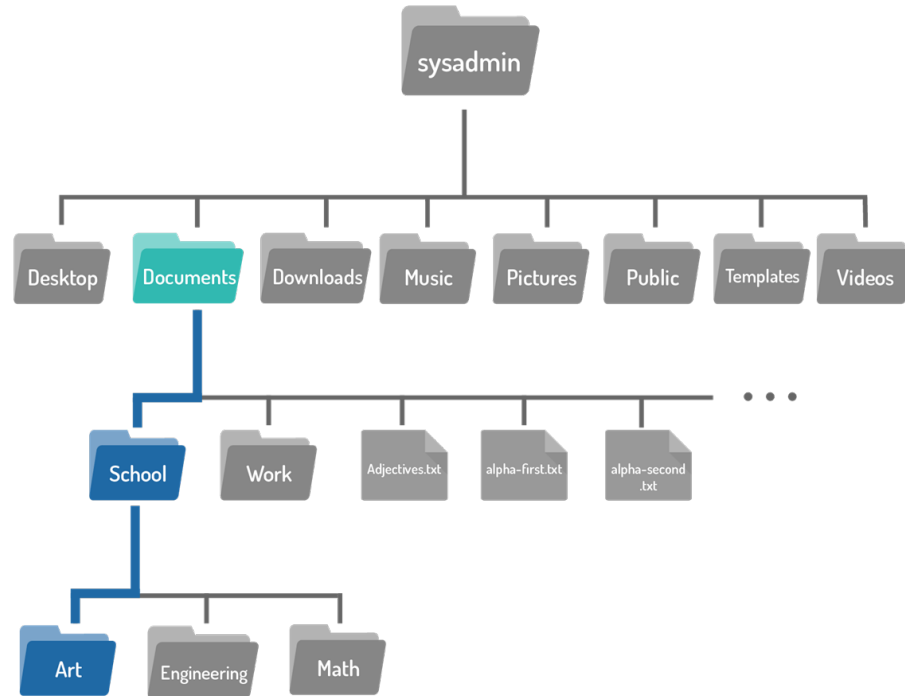
# Relative Paths

- A relative path gives directions to a file relative to the current location in the filesystem.
- The user must currently be in a directory that contains objects in the path
- Relative paths start with the name of a directory:

```
sysadmin@localhost:~/Documents/$ cd School/Art  
sysadmin@localhost:~/Documents/School/Art$
```



# Relative Paths



# Paths - Shortcuts: The .. Characters

- Two period .. characters always represents one directory higher relative to the current directory, sometimes referred to as the parent directory.
- For example, to move from the `Art` directory back to the `School` directory:

```
sysadmin@localhost:~/Documents/School/Art$ cd ..  
sysadmin@localhost:~/Documents/School$
```

- The double dot can also be used in longer paths as well:

```
sysadmin@localhost:~/Documents/School$ cd ../../Downloads
```

# Paths - Shortcuts: The . Character

- The single period . character always represents the current directory.
- For the `cd` this shortcut is not very useful, but it comes in handy for commands covered in subsequent sections.

# Listing Files in a Directory

# Listing Files in a Directory

- The `ls` (list) command is one of the most powerful tools for navigating the filesystem.

```
ls [OPTION]... [FILE]...
```

- The `ls` command is used to display the contents of a directory and can provide detailed information about the files.
- When used with no options or arguments, the `ls` command lists the files in the current directory:

```
sysadmin@localhost:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates
Videos
```

- The `ls` command can also be used to list the contents of any directory in the filesystem, provide the path to the directory as an argument.

# Listing Hidden Files

- A hidden file is any file (or directory) that begins with a dot `.` character.
- The `ls` command omits *hidden files* by default.
- To display all files, including hidden files, use the `-a` option to the `ls` command:

```
sysadmin@localhost:~$ ls -a
.          .bashrc    .selected_editor  Downloads  Public
..         .cache     Desktop          Music      Templates
.bash_logout .profile  Documents        Pictures   Videos
```

- Most of the hidden files are *customization files*, designed to customize how Linux, your shell or programs work.
  - For example, the `.bashrc` file in the home directory customizes features of the shell

# Long Display Listing

- Each file has details associated with it called *metadata*, this can include information such as the size, ownership, or timestamps.
- Use the `-l` option to the `ls` command to view this information.
- For example, below, a listing of the `/var/log` directory:

```
sysadmin@localhost:~$ ls -l /var/log/
total 832
-rw-r--r-- 1 root    root    17869 Mar 14 17:48 alternatives.log
drwxr-x--- 2 root    adm     4096 Mar 14 17:48 apache2
drwxr-xr-x 2 root    root     4096 Mar 14 17:45 apt
-rw-r----- 1 syslog adm      380 Jul 28 03:45 auth.log
```

- In the output above, each line describes metadata about a single file.

# Long Display Listing

2019 © Network Development Group Inc.

- The following describes each of the fields of data in the output of the `ls -l` command:

- **File Type:**

```
-rw-r--r-- 1 root  root  17869 Mar 14 17:48 alternatives.log
drwxr-x--- 2 root  adm   4096 Mar 14 17:48 apache2
```

- The first character of each line indicates the type of file. The file types are:

Symbol	File Type	Description
d	directory	A file used to store other files.
-	regular file	Includes readable files, images files, binary files, and compressed files.
l	symbolic link	Points to another file.
s	socket	Allows for communication between processes.
p	pipe	Allows for communication between processes.
b	block file	Used to communicate with hardware.
c	character file	Used to communicate with hardware.



# Long Display Listing

- **Permissions:**

```
drwxr-xr-x 1 root root      0 Apr 11 21:58 upstart
```

- The next nine characters demonstrate the permissions of the file.
- Permissions indicate how certain users can access a file.

- **Hard Link Count:**

```
-rw-r----- 1 syslog adm 23621 Aug 23 15:17 auth.log
```

- This number indicates how many hard links point to this file.

# Long Display Listing

- User Owner:

```
-rw-r----- 1 syslog adm 416 Aug 22 15:43 kern.log
```

- Every file is owned by a user account.
- This is important because the owner has the rights to set permissions on a file.

- Group Owner:

```
-rw-rw-r-- 1 root utmp 292584 Aug 20 18:44 lastlog
```

- Indicates which group owns this file, this is important because any member of this group has a set of permissions on the file.

# Long Display Listing

- **File Size:**

```
-rw-r----- 1 syslog adm 1087150 Aug 23 15:17 syslog.1
```

- Size of files in bytes.

- **Timestamp :**

```
drwxr-xr-x 1 root root 32 Jul 17 03:36 fsck
```

- Indicates when a file's contents were last modified.

- **File Name:**

```
-rw-r--r-- 1 root root 47816 Jul 17 03:36 bootstrap.log
```

- The name of the file or directory.

# Human Readable Sizes

- The `-l` option to the `ls` command displays file sizes in bytes.
- For text files, a byte is 1 character so small files are easy to read, but for larger files, it is hard to comprehend how large the file is:

```
sysadmin@localhost:~$ ls -l /usr/bin/omshell
-rwxr-xr-c 1 root root 1561400 Oct 9 2012 /usr/bin/omshell
```

- The file size is hard to determine in bytes. Is 1561400 a large file or small?
- To present the file size in a more human readable size, like megabytes or gigabytes, add the `-h` option (with the `-l` option) to the `ls` command:

```
sysadmin@localhost:~$ ls -lh /usr/bin/omshell
-rwxr-xr-c 1 root root 1.5M Oct 9 2012 /usr/bin/omshell
```

# Listing Directories

- When the command `ls -d` is used, it refers to the current directory, and not the contents within it.

```
sysadmin@localhost:~$ ls -d
```

```
.
```

- The `.` represents the current directory.
- The `ls -l` command lists the contents of the directory:

```
sysadmin@localhost:~$ ls -l
```

```
total 0
```

```
drwxr-xr-x 1 sysadmin sysadmin 0 Apr 15 2015 Desktop
```

```
drwxr-xr-x 1 sysadmin sysadmin 0 Apr 15 2015 Documents
```

```
drwxr-xr-x 1 sysadmin sysadmin 0 Apr 15 2015 Downloads
```

# Listing Directories

- To use the `ls -ld` command in a meaningful way requires the addition of the `-l` option:

```
sysadmin@localhost:~$ ls -ld
drwxr-xr-x 1 sysadmin sysadmin 224 Nov  7 17:07 .
```

# Recursive Listing

- *Recursive listing* is when you want to display all of the files in a directory as well as all of the files in all subdirectories under a directory.
- To perform a recursive listing, use the `-R` option to the `ls` command:

```
sysadmin@localhost:~$ ls -R /etc/ppp
/etc/ppp:
chap-secrets  ip-down.ipv6to4  ip-up.ipv6to4  ipv6-up  pap-secrets
ip-down       ip-up            ipv6-down      options  peers

/etc/ppp/peers:
Output Omitted...
```

- Note that in the example above, the files in the `/etc/ppp` directory were listed first and the files in the `/etc/ppp/peers` directory were listed after.

Be careful with this option; for example, running the command on the root directory would list every file on the file system.

# Sort a Listing

- By default, the `-ls` command sorts files alphabetically by file name.
- Sometimes it may be useful to sort files using different criteria.
- To sort files by size, we can use the `-S` option (capital letter `s`).
- While the `-S` option works by itself, it is most useful when used with the `-l` option so the file sizes are visible:

```
sysadmin@localhost:~$ ls -lS /etc/ssh
total 160
-rw-r--r-- 1 root root 125749 Apr 29 2014 moduli
-rw-r--r-- 1 root root 2489 Jan 29 2015 sshd_config
-rw----- 1 root root 1675 Jan 29 2015 ssh_host_rsa_key
```



# Sort a Listing

- It may also be useful to use the `-h` option to display human-readable file sizes:

```
sysadmin@localhost:~$ ls -lSh /etc/ssh
total 160K
-rw-r--r-- 1 root root 123K Apr 29 2014 moduli
-rw-r--r-- 1 root root 2.5K Jan 29 2015 sshd_config
-rw----- 1 root root 1.7K Jan 29 2015 ssh_host_rsa_key
Output Omitted...
```

- The `-t` option sorts files based on the time they were modified:

```
sysadmin@localhost:~$ ls -tl /etc/ssh
total 160
-rw----- 1 root root 668 Jan 29 2015 ssh_host_dsa_key
-rw-r--r-- 1 root root 607 Jan 29 2015 ssh_host_dsa_key.pub
-rw----- 1 root root 227 Jan 29 2015 ssh_host_ecdsa_key
Output Omitted...
```

# Sort a Listing

- For more detailed modification time information you can use the `--full-time` option to display the complete timestamp (including hours, minutes, seconds):

```
sysadmin@localhost:~$ ls -t --full-time /etc/ssh
total 160
-rw----- 1 root root    668 2015-01-29 03:17:33.000000000 +0000 ssh_host_dsa_key
-rw-r--r-- 1 root root    607 2015-01-29 03:17:33.000000000 +0000 ssh_host_dsa_key.pub
-rw----- 1 root root    227 2015-01-29 03:17:33.000000000 +0000 ssh_host_ecdsa_key
```

- It is possible to perform a reverse sort by using the `-r` option. It can be used alone, or combined with either the `-S` or `-t` options:

```
sysadmin@localhost:~$ ls -lrS /etc/ssh
total 160
-rw-r--r-- 1 root root    179 Jan 29  2015 ssh_host_ecdsa_key.pub
-rw----- 1 root root    227 Jan 29  2015 ssh_host_ecdsa_key
```