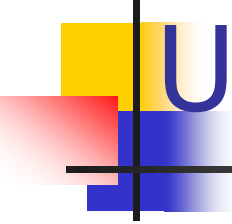


SEEM3460 Tutorial

Unix Files and Attributes

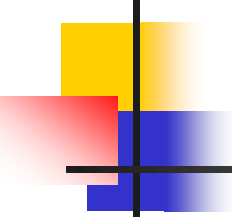
Before we start:

Please log on to the server



Unix File Attributes

- Typing **ls -l** at the command prompt, users will see a whole host of properties associated with each file displayed.
- The main properties displayed are:
 - The file type and access permissions
 - The link count of the file
 - The file's owner and group
 - The size of the file
 - The date on which the file was last modified
 - The name of the file
- **ls -ld** is used to show the attributes of the directory



'-' - regular file
'd' - directory
's' - symbolic link
'p' - named pipe
'c' - character special
'b' - block special

link count

1

spate

↑

user

fcf

↑

group

-rw-r--r-

user group and other permissions

file size

137564

Feb 13 09:05

date of last modification

file name

layout.tex



File type

- A file's type can be determined by using the **file** utility.
 - The file utility attempts to describe the contents of the fileName arguments, including the language that any text is written in. When using file on a symbolic link file, file reports on the file that the link is pointing to, rather than the link itself.
- The first character of Field 1 indicates the type of the file.

Character	File type
-	regular file
d	directory file
b	buffered special file (such as a disk drive)
c	unbuffered special file (such as a terminal)
l	symbolic link
p	pipe
s	socket



File Permissions

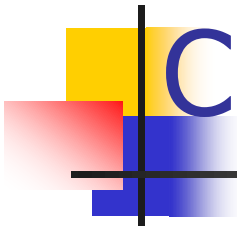
- The next nine characters of Field 1 indicate the file's permission settings.
- These nine characters should be thought of as being arranged in three groups of three characters, where each cluster of three letters has the same format.
 - r: read permission
 - w: write permission
 - x: execute permission
 - If a dash occurs instead of a letter, then permission is denied.
 - Three clusters: User, Group, Others
 - E.g., rw-r--r--



File Permissions

- The meaning of the read, write, and execute permissions depends on the type of the file, as shown below.

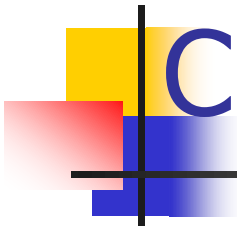
	Regular file	Directory file	Special file
Read	The process may read the contents.	The process may read the directory (i.e., list the names of the files that it contains).	The process may read from the file using the <code>read()</code> system call.
Write	The process may change the contents.	The process may add files to or remove files from the directory.	The process may write to the file, using the <code>write()</code> system call.
Execute	The process may execute the file (which makes sense only if the file is a program)	The process may access files in the directory or any of its subdirectories.	No meaning.



Change File Permissions

- The **chmod** utility is used to change the permission of files. For example, to remove read permission from others:
 - `chmod o-r <filename>`
- The table below shows some other examples of the use of chmod.

Requirement	Change parameters
Add group write permission.	g+w
Remove user read and write permission.	u-rw
Add execute permission for user, group, and others.	a+x
Give others read permission.	o+r
Add write permission for user, and remove read permission from group.	u+w, g-r

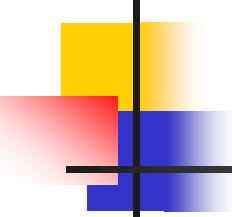


Change File Permissions

- The `chmod` utility allows you to specify the new permission setting of a file as an octal number. Each octal digit represents a permission triplet.
 - For example, `rwxr-x---` will be `750`.

	User	Group	Others
setting	<code>rwX</code>	<code>r-X</code>	<code>---</code>
binary	<code>111</code>	<code>101</code>	<code>000</code>
octal	<code>7</code>	<code>5</code>	<code>0</code>

- How about `rwXrwXrwX`? And `rw-r--r--`?



'-' - regular file
'd' - directory
's' - symbolic link
'p' - named pipe
'c' - character special
'b' - block special

link count

1

spate

↑

user

fcf

↑

group

-rw-r--r-

user group and other permissions

file size

137564

Feb 13 09:05

date of last modification

file name

layout.tex



Unix File Naming Rules

- A UNIX filename may be up to **255** characters in length. You may use any printable characters you want in a filename except the slash (/)
 - You avoid the use of any character that is special to a shell (like `<`, `>`, `*`, `?`, or the tab), as these can confuse both the user and the shell.
- There is no requirement that a file end in an extension.
- You can not name a file period (.) or period period (..).
- Directories are files in Unix, so directory names follow these same rules also.



File Modification Time

- Type **stat** at the command prompt, users will see the access time, modify time, and change time of a file.
 - sepc92:> stat <filename>
 - The **access** time is when a file was last accessed (read or modified).
 - The **modify** time is when a file last had its contents modified.
 - The **change** time is when a file last had its metadata changed (such as file permissions or the name of the file).
- The **find** utility can be used to search for file names matching with a certain pattern.
 - sepc92:> find -name "*.c"



File Modification Time

- The **find** utility supports an option based on the last modification time.
 - sepc92:> find -name "*.c" -mtime -1
 - the modification time is less than 1 day ago.
 - Also for -atime and -ctime
- The **touch** command can set the modification and access times to predefined values.
 - touch options expression filename
 - touch filename (without options and expression)
 - both times are set to the current time and creates the file
 - The expression consists of MMDDhhmm (month, day, hour and minute). Options can be -a or -m.
 - Touch -a -t 03161430 filename



Disk Usage of Files & Dirs

- Use **du** (Disk Usage) command to check the information of disk usage of files and directories on a machine.
 - Using the **"-h"** option with the **"du"** command provides results in "Human Readable Format". This means you can see sizes in Bytes, Kilobytes, Megabytes, Gigabytes, etc.
- To get the summary of a grand total disk usage size of a directory uses the option **"-s"**.
- Display the disk usage based on a modification of time, use the flag **"--time"**.