

# Chapter 2

# FILE COMMANDS

*Get on the  
Fast Track!*



TM

**SYS-ED/  
Computer  
Education  
Techniques, Inc.**

## Objectives

You will learn:

- Purpose and function of file commands.
- Interrelated usage of commands.



**1 File Command**

The command file is used to determine the type of the file. It reads/analyzes the file and displays the file type.

- If a file is ASCII, the system looks at the first 512 bytes and displays ASCII.
- If a file is executable, the command displays a version number.

A magic file is used to search the file and determine its type. These files have a 'magic number' stored in a specific location which informs the UNIX operating system that the file is a binary executable and its type.

Syntax:

`file [ -bciknsvzL ] [ -f namefile ] [ -m magicfiles ] file`

**Example:**

<code>file *</code>	Display file types for all files in a directory.
---------------------	--

---

## 2 CUT

This command cuts out selected fields of each line of a file.

Syntax:

```
cut [-b] [-c] [-f] list [-n] [-d delim] [-s] [file]
```

Option	Description
-b list	The list following -b specifies byte positions. When -b and -n are used together, list is adjusted so that no multi-byte character is split. When b is used, the input line should contain 1023 bytes or less.
-c list	The list following -c specifies character positions.
-f list	The list following -f is a list of fields assumed to be separated in the file by a delimiter character. Lines with no field delimiters will be passed through intact, unless -s is specified. If -f is used, the input line should contain 1023 characters or less.
list	A comma-separated or blank-character-separated list of integer field numbers (in increasing order), with optional - to indicate ranges.
-n	Do not split characters. When -b list and -n are used together, list is adjusted so that no multi-byte character is split.
-d delim	The character following -d is the field delimiter (-f option only). Default is tab. Space or other characters with special meaning to the shell must be quoted. delim can be a multi-byte character.
-s	Suppresses lines with no delimiter characters in case of -f option. Unless specified, lines with no delimiters will be passed through untouched.
file	A path name of an input file. If no file operands are specified, or if a file operand is -, the standard input will be used.

### Examples:

```
ls -al | cut f1 -d ' '
```

Displays the first field with a delimiter of space.

```
ls -al | cut -c1-4 -d ' '
```

Displays columns 1 thru 4 of the ls output.

**3 PASTE**

Merge corresponding or subsequent lines of files.

Syntax:

paste [-s] [-d list] file

Option	Description
-s	Concatenate all of the lines of each separate input file in command line order. The NEWLINE character of every line except the last line in each input file will be replaced with the TAB character, unless otherwise specified by the -d option.
-d list	Unless a backslash character (\) appears in list, each character in list is an element specifying a delimiter character. If a backslash character appears in list, the backslash character and one or more characters following it are an element specifying a delimiter character as described below. These elements specify one or more delimiters to use, instead of the default TAB character, to replace the NEWLINE character of the input lines. When the list is exhausted the first element from the list is reused.

**Examples**

ls | paste -s

Take the input from ls and paste that input into columns.

4 FIND

Finds one or more files assuming that filenames have been and exist.

Syntax:

find path expressions

Option	Description								
Path	A path name of a starting point in the directory hierarchy.								
-atime n	True if the file was accessed n days ago. The access time of directories in path is changed by find itself.								
-cpio device	Always true; write the current file on device in cpio format (5120-byte records).								
-ctime n	True if the file's status was changed n days ago.								
-exec command	True if the executed command returns a zero value as exit status. The end of command must be punctuated by an escaped semicolon. A command argument {} is replaced by the current path name.								
-follow	Always true; causes symbolic links to be followed. When following symbolic links, find keeps track of the directories visited in order that it can detect infinite loops. This expression should not be used with the -type l expression.								
-fstype type	True if the filesystem to which the file belongs is of type type .								
-group gname	True if the file belongs to the group gname. If gname is numeric and does not appear in the /etc/group file, it is taken as a group ID.								
-inum n	True if the file has inode number n.								
-links	True if the file has n links.								
-local	True if the file system type is not a remote file system type as defined in the /etc/dfs/fstypes file. nfs is used as the default remote filesystem type if the /etc/dfs/fstypes file is not present.								
-ls	Always true; prints current path name together with its associated statistics. These include (from left to right): <table border="1" data-bbox="492 1209 1430 1360"> <tbody> <tr> <td>Inode number</td> <td>size in kilobytes (1024 bytes)</td> </tr> <tr> <td>protection mode</td> <td>number of hard links</td> </tr> <tr> <td>User</td> <td>Group</td> </tr> <tr> <td>size in bytes</td> <td>modification time</td> </tr> </tbody> </table>	Inode number	size in kilobytes (1024 bytes)	protection mode	number of hard links	User	Group	size in bytes	modification time
Inode number	size in kilobytes (1024 bytes)								
protection mode	number of hard links								
User	Group								
size in bytes	modification time								

If the file is a special file the size field will instead contain the major and minor device numbers.

Option	Description
-mount	Always true; restricts the search to the file system containing the directory specified. Does not list mount points to other file systems.
-mtime n	True if the file's data was modified n days ago.
-name pattern	True if pattern matches the current file name. Normal shell file name generation characters may be used. A backslash (\) is used as an escape character within the pattern. The pattern should be escaped or quoted when find is invoked from the shell.
-ncpio device	Always true; write the current file on device in cpio -c format (5120 byte records).
-newer file	True if the current file has been modified more recently than the argument file.
-nogroup	True if the file belongs to a group not in the /etc/group file.
-nouser	True if the file belongs to a user not in the /etc/passwd file.
-perm [-]onum	True if the file permission flags exactly match the octal number onum see chmod). If onum is prefixed by a minus sign (-), only the bits that are set in onum are compared with the file permission flags, and the expression evaluates true if they match.
-print	Always true; causes the current path name to be printed.
-size n[c]	True if the file is n blocks long (512 bytes per block). If n is followed by a c, the size is in bytes.
-type c	True if the type of the file is c, where c is b, c, d, D, f, l, p, or s for block special file, character special file, directory, door, plain file, symbolic link, fifo (named pipe), or socket, respectively.
-user uname	True if the file belongs to the user uname . If uname is numeric and does not appear as a login name in the /etc/passwd file, it is taken as a user ID.

**Examples:**

System would search for any file named mypage.htm in the current directory and any subdirectory.

```
find -name 'mypage.htm'
```

System would search for any file named mypage.htm on the root and all subdirectories from the root.

```
find / -name 'mypage.htm'
```

System would search for any file beginning with file in the current directory and any subdirectory.

```
find -name 'file*'
```

System would search for any file that is larger then 1000k.

```
find -name '*' -size +1000k
```

5 GREP

GREP finds text within a file.

Syntax:

grep [-b] [-c] [-h] [-i] [-l] [-n] [-s] [-v] [-w] text filenames

Option	Description
-b	Precede each line by the block number on which it was found. This can be useful in locating block
-c	Print only a count of the lines that contain the pattern.
-h	Prevents the name of the file containing the matching line from being appended to that line. Used when searching multiple files.
-i	Ignores case.
-l	Displays the files that have the text not the text within the file.
-n	Precede each line by its line number in the file (first line is 1).
-s	Suppress error messages about nonexistent or unreadable files.
-v	Specifies that files are being looked for which do not contain the text.
-w	Search for the expression as a word as if surrounded by \< and \>.
text	The text that is being searched for / not being searching for.
filenames	The name of the file that is being searched.

5.1 Patterns used for Searching

Character	Description
.	Matches a single character.
*	Wild character.
[]	Matches any character contained within the bracket.
^	Represents the beginning of the line.
\$	Represents the end of the line, therefore \$. would pull up any lines that ended with .
\	The next character is to be included in the search for C\ C.

Examples:

Searches all .htm files in the current directory for any reference of UNIX.

```
grep "unix" *.htm
```

Displays all files with an 'a' in a line and the line number.

```
grep -n 'a' *
```

6 SORT

Sorts the lines in a text file.

Syntax:

sort [-b] [-d] [-f] [-i] [-m] [-M] [-n] [-r] [-u] [+fields] filename [-o outputfile]

Option	Description
-b	Ignores spaces at beginning of the line.
-d	Uses dictionary sort order and ignores the punctuation.
-f	Ignores caps.
-i	Ignores nonprinting control characters.
-m	Merges two or more input files into one sorted output.
-M	Treats the first three letters in the line as a month.
-n	Sorts by the beginning of the number at the beginning of the line.
-r	Sorts in reverse order.
-u	If line is duplicated only display once.
+fields	Sorts by fields, usually by tabs.
filename	The name of the file that needs to be sorted.
-o outputfile	Sends the sorted output to a file.
-t	Field separator
-k keydef	The keydef argument is a restricted sort key field definition.  The format of this definition is: -k field_start [type] [,field_end [type] ]

Examples

sort -r file.txt

sort -r -k 6,8 tt

7 CMP

Compares two files and indicates which line numbers are different.

Syntax:

cmp [-c] [-i N] [-l] [-s] [-v] firstfile secondfile

Option	Description
-c	Output differing bytes as characters.
-i N	Ignore differences in the first N bytes of input.
-l	Write the byte number (decimal) and the differing bytes (octal) for each difference.
-s	Write nothing for differing files; return exit statuses only.
-v	Output version info.
firstfile	First file that is to be compared.
secondfile	Second file that is to be compared.

Examples:

Compares file1 to file2 and outputs results.

cmp file1.txt file2.txt

cmp -c -l filea fileb

8 DIFF

Displays two files and prints the lines that are different.

Syntax:

```
diff [-b] [-i] [-t] [-w] [-c] [-C] [-e] [-f] [-h] [-n] [-D string] [-l] [-r] [-s] [-S name] [fileone filetwo ]
[directoryone directorytwo]
```

Option	Description
-b	Ignores spacing differences.
-i	Ignores case.
-t	Expands TAB characters in output lines. Normal or -c output adds character(s) to the front of each line that may adversely affect the indentation of the original source lines and make the output lines difficult to interpret. This option will preserve the original source's indentation.
-w	Ignores spaces and tabs.
-c	Produces a listing of differences with three lines of context. Output begins with identification of the files involved and their creation dates, then each change is separated by a line with a dozen *'s. The lines removed from file1 are marked with '-'; those added to file2 are marked '+'. Lines that are changed from one file to the other are marked in both files with '!'. The lines removed from file1 are marked with '-'; those added to file2 are marked '+'. Lines that are changed from one file to the other are marked in both files with '!'.
-C	Produces a listing of differences identical to that produced by -c with number lines of context.
-e	Produces a script of only a, c, and d commands for the editor ed , which will recreate file2 from file1.
-f	Produces a similar script, not useful with ed , in the opposite order.
-h	Is well suited for changed stretches which are short and well separated and files of unlimited length. Options -c, -e, -f, and -n are unavailable with -h. diff does not descend into directories with this option.
-n	Produces a script similar to -e, but in the opposite order and with a count of changed lines on each insert or delete command.
-D string	Creates a merged version of file1 and file2.
-l	Produce output in long format.
-r	Applies diff recursively to common subdirectories encountered.
-s	Reports files that are the identical; these would not otherwise be mentioned.
-S name	Starts a directory diff in the middle, beginning with the file name.
filenameone	File one for comparing.
filenametwo	File two for comparing.
directoryone	Directory one for comparing.
directorytwo	Directory two for comparing.

**Examples:**

```
diff help dir2 -
```

Compares the directory named help with the directory named dir2.



9 JOIN

The join command forms, on the standard output, a join of the two relations specified by the lines of file1 and file2.

Syntax:

```
join [-a filenumber | -v filenumber ] [-1 fieldnumber ] [-2 fieldnumber ] [-o list ] [-e string ] [-t char ] file1 file2
```

Option	Description
-a filenumber	In addition to the normal output, produce a line for each unpairable line in file filenumber, where filenumber is 1 or 2. If both -a 1 and -a 2 are specified, all unpairable lines will be output.
-v filenumber	Instead of the default output, produce a line only for each unpairable line in filenumber, where filenumber is 1 or 2. If both -v 1 and -v 2 are specified, all unpairable lines will be output.
-1 fieldnumber	Join on the fieldnumberth field of file 1. Fields are decimal integers starting with 1.
-2 fieldnumber	Join on the fieldnumberth field of file 2. Fields are decimal integers starting with 1.
-j fieldnumber	Equivalent to -1fieldnumber -2fieldnumber.
-j1 fieldnumber	Equivalent to -1fieldnumber.
-j2 fieldnumber	Equivalent to -2fieldnumber Fields are numbered starting with 1.
-o list	Each output line includes the fields specified in list. Fields selected by list that do not appear in the input will be treated as empty output fields.  Each element of which has the either the form filenumber.fieldnumber, or 0, which represents the join field.  The common field is not printed unless specifically requested.
-e string	Replace empty output fields with string.
-t char	Use character char as a separator.

10 PG

File filters for CRTs.

Syntax:

pg [-number] [-p string] [-c] [-e] [-f] [-n] [-r][-s] [+ linenumber] [+ pattern /]

[filename]

Option	Description
-number	An integer specifying the size (in lines) of the window that pg is to use instead of the default. (On a terminal containing 24 lines, the default window size is 23).
-p string	pg uses string as the prompt. If the prompt string contains a %d, the first occurrence of %d in the prompt will be replaced by the current page number when the prompt is issued. The default prompt string is ``:``.
-c	Home the cursor and clear the screen before displaying each page. This option is ignored if clear_screen is not defined for this terminal type in the terminfo(4) data base.
-e	pg does not pause at the end of each file.
-f	Normally, pg splits lines are longer than the screen width, but some sequences of characters in the text being displayed generate undesirable results. The -f option inhibits pg from splitting lines.
-n	Commands must be terminated by a <new line> character. This option causes an automatic end of command as soon as a command letter is entered.
-r	Restricted mode. The shell escape is disallowed. pg prints an error message but does not exit.
-s	pg prints all messages and prompts in the standard output mode.
+linenumber	Start up at linenumber.
+/pattern/	Start up at the first line containing the regular expression pattern.
Filename	A path name of a text file to be displayed. If no filename is given, or if it is -, the standard input is read.

11 COMMANDS

h help.	q or Q quit
<blank> or <newline> next page.	l next line.
d or <^D> display half a page more or <^L> redisplay current page.	f skip the next page forward.
n next file.	p previous file.
\$ last page.	w or z set window size and display next page.
s savefile save current file in savefile.	/pattern/ search forward for pattern.
?pattern? or ^pattern^ search backward for pattern	!command execute command

**Examples:**

Would execute and display the file myfile.txt.

```
pg myfile.txt
```

