

## The cut Command

One particular field from any file or from output of any command can be extracted and displayed using the cut command. A single character can also be extracted by using the -c option of the cut command.

### Example 4.22

Let us consider the file mast. If we want to view only the second field of the file, then the command is:

```
$ cut -d":" -f2 mast
kane and abel
kane and abel
kane and abel
the naked face
the naked face
where eagles dare
the ring
the ring
the almighty
$_
```

In the above command “-d” represents the delimiter or the field separator. The option “-f” is followed by the list of fields that have to be displayed.

The cut command also takes its input from the output of another command.

### Example 4.23

```
$ cat mast | cut -d":" -f3
j archer
j archer
j archer
s sheldon
s sheldon
a maclean
d steele
d steele
I wallace
$_
```

## The paste Command

The paste command merges the contents of two files into a single file. It reads a line from each file in the file list specified and combines them into a single line.

### Example 4.24

If we have three files called `firstname`, `secondname`, `lastname` whose contents are as follows:

```
$ cat firstname
george
victoria
sylvia
$_
$ cat secondname
mathew
thomas
peter
$_
$ cat lastname
thomas
peter
mary
$_
```

The paste command given below will merge the contents of the three files.

```
$ paste firstname secondname lastname
george mathew thomas
victoria thomas peter
sylvia peter mary
$_
```

The paste command also takes the output from another command as input.

## The tee Command

When using pipes, in certain situations we might get undesired result or an error message. We will not be able to identify the exact position of the error. It is also possible that we want the output of a particular command in a long pipeline to be stored in a file for later use. The tee command is used in these circumstances.

## Example 4.25

```
$ cat mast | sort | tee temp | cut -d":" -f1
b001
b001
b001
b003
b003
b005
b006
b006
b008
$ _
$ cat temp
b001:kane and abel:jarcher
b001:kane and abel:jarcher
b001:kane and abel:j archer
b003:the naked face:s sheldon
b003:the naked face:s sheldon
b005:where eagles dare maculae
b006:the ring:d steele
b006:the ring:d steele
b008:the almighty:i wallace
$ _
```

In the above example the sorted output of the cat command is stored in a file named temp and only the first field is displayed on the screen.

### The tr Command

The tr command reads standard input and, for each input character, maps it to an alternate character, deletes the character, or leaves the character alone. The output is written to the standard output.

The output of several commands is separated from each other by more than a single space. This command when used with the "-s" option squeezes multiple spaces into a single space.

## Example 4.26

Let us assume that we have a file called bag whose contents are as follows:

```
$ cat bag
This bag contains twenty apples.
$ _
```

The following command converts the contents of the entire file into uppercase.

```
$ cat bag | tr "[a-z]" "[A-Z]"
THIS BAG CONTAINS TWENTY APPLES.
$ _
```

### Usage of Pipes and Filters together

Pipes and filters are very effective tools of the LINUX operating system. Pipes and filters increase the efficiency and utility of the LINUX operating system.

Let us try and understand the usage of pipes and filters together effectively with examples.

An important usage of the grep command using the "^d" pattern is to list the directories.

```
$ ls -l | grep "^d"
```

If we want to display only the first field of the first ten lines of a sorted text file, the command is:

```
$ cat text | sort | head | cut -d" " -f1
```

If we want to extract the file owners and view it pagewise, the command is:

```
$ ls -l | tr -s " " | cut -d" " -f3 | pg
```