

# UNIX Tutorial Six

---

## Other useful UNIX commands

### quota

All students are allocated a certain amount of disk space on the file system for their personal files, usually about 100Mb. If you go over your quota, you are given 7 days to remove excess files.

To check your current quota and how much of it you have used, type

```
% quota -v
```

### df

The **df** command reports on the space left on the file system. For example, to find out how much space is left on the fileserver, type

```
% df .
```

### du

The **du** command outputs the number of kilobytes used by each subdirectory. Useful if you have gone over quota and you want to find out which directory has the most files. In your home-directory, type

```
% du -s *
```

The **-s** flag will display only a summary (total size) and the **\*** means all files and directories.

### gzip

This reduces the size of a file, thus freeing valuable disk space. For example, type

```
% ls -l science.txt
```

and note the size of the file using **ls -l**. Then to compress science.txt, type

```
% gzip science.txt
```

This will compress the file and place it in a file called **science.txt.gz**

To see the change in size, type **ls -l** again.

To expand the file, use the **gunzip** command.

```
% gunzip science.txt.gz
```

### zcat

**zcat** will read gzipped files without needing to uncompress them first.

```
% zcat science.txt.gz
```

If the text scrolls too fast for you, pipe the output through **less**.

```
% zcat science.txt.gz | less
```

## [file](#)

**file** classifies the named files according to the type of data they contain, for example ascii (text), pictures, compressed data, etc.. To report on all files in your home directory, type

```
% file *
```

## [diff](#)

This command compares the contents of two files and displays the differences. Suppose you have a file called **file1** and you edit some part of it and save it as **file2**. To see the differences type

```
% diff file1 file2
```

Lines beginning with a < denotes file1, while lines beginning with a > denotes file2.

## [find](#)

This searches through the directories for files and directories with a given name, date, size, or any other attribute you care to specify. It is a simple command but with many options - you can read the manual by typing **man find**.

To search for all files with the extension **.txt**, starting at the current directory (.) and working through all sub-directories, then printing the name of the file to the screen, type

```
% find . -name "*.txt" -print
```

To find files over 1Mb in size, and display the result as a long listing, type

```
% find . -size +1M -ls
```

## [history](#)

The C shell keeps an ordered list of all the commands that you have entered. Each command is given a number according to the order it was entered.

```
% history (show command history list)
```

If you are using the C shell, you can use the exclamation character (!) to recall commands easily.

```
% !! (recall last command)
```

```
% !-3 (recall third most recent command)
```

```
% !5 (recall 5th command in list)
```

```
% !grep (recall last command starting with grep)
```

You can increase the size of the history buffer by typing

```
% set history=100
```



M.Stonebank@surrey.ac.uk, © 24th August 2001