12. Sort, Grep, wc, diff, find, file

Create the following files using vi

Keep same cases upper or lower exactly as shown:

Create a file called **numdays** with the following:

7 monday

4 Tuesday

6 wednesday

3 thursday

5 friday

5 friday

8 Saturday

2 sunday

You should already have a file called **days** from the last exercise with the days written in it but if not, create it as follows:

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

Lab 12.1 - File

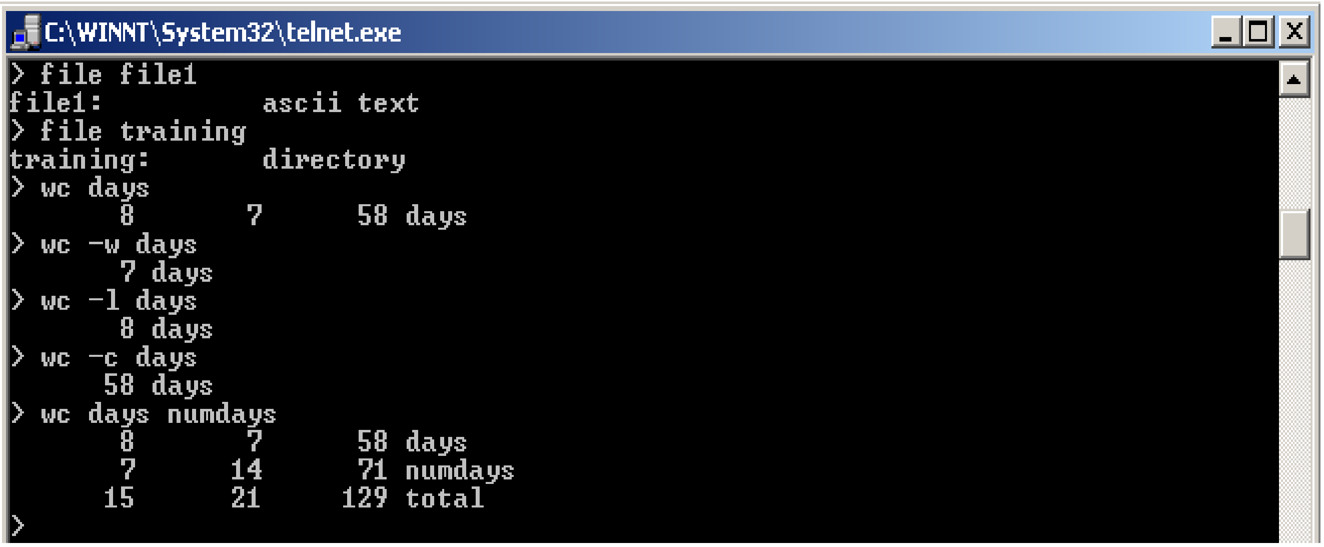
The **file** command allows you to find out what files are, for example to determine whether an unknown file is in the text format, and therefore suitable for direct viewing.

```
file days
file *  (will show all the files in your current directory to determine what kinds of files they are.
```

Lab 12.2 - wc

wc: used to display a count of lines, words and characters in a file. The wc utility reads one or more input files and by default writes the number of lines words and characters contained in each input file to the standard output.

wc days	//number of lines words and characters in the file days
wc days numdays	// number of lines words and characters in the file days and numdays
wc –l days	//count the number of lines
wc –w days	//count the number of words
wc –c days	//count the number of characters



```
> file file1
file1:          ascii text
> file training
training:       directory
> wc days
       8       7      58 days
> wc -w days
       7 days
> wc -l days
       8 days
> wc -c days
      58 days
> wc days numdays
       8       7      58 days
       7      14      71 numdays
      15      21     129 total
>
```

Figure 12.1: Example of using file and wc

Lab 12.3 - Grep

grep – Search a file for a pattern

globally search for a **r**egular **e**xpression and **p**rint all lines containing it.

Syntax: **grep "pattern" file(s)**

```
grep day days numdays
```

-l: print out the names of files where the pattern was found

```
grep –l day days numdays
```

-n: print the matched line and its line number. Show the line numbers where the pattern is not found.

```
grep –n day days
```

-c: print only the count of matching lines

```
grep –c day days
```

-v print all lines that do not match the pattern entered. For example, all lines that do not contain '234'

```
grep -v Wed days
```

-i match with either upper or lowercase

grep -i wed days numdays (note that days has uppercase wed)

Lab 12.4 - Sort

The sort utility sorts the contents of a file; it arranges the lines in the order of ASCII character codes. For example, sort sorts the file days in alphabetical order .

```
sort days
```

The list of arguments which the sort command can have include:

-n numeric sort-included for numeric lists only. For example.

```
sort -n numdays
```

-r reverse order for both numeric and text files i.e. descending versus ascending.
For example:

```
sort -r days
```

-u unique – this option outputs repeated lines only once. For example:

```
sort -u numdays
```

Lab 12.5 – find – Finding Files

The **find** command scans part of the Unix filestore for files and or directories with specified characteristics:

```
touch chat     creates a blank file called chat
find . –name “c*”
```

The above example will find all files and directories whose name starts with c in the current directory and below, in any sub-directories.

The following will find all files from the current directory & sub-directories beginning with “c” that have been modified in the last thirty days:

```
find . –name  “c*” –mtime –30
```

The following will find all files from the current directory & sub-directories beginning with “c” that are older than 1 day:

```
find . –name  “c*” –mtime +1
```

The following will find all files from the current directory & sub-directories whose name doesn’t begin with a “c” (not c*).

```
Find . ! –name “c*”
```

Lab 12.6 - exec

This option can be used to execute a command on the files found, for example the following will find all files called **chat** in the current directory and sub-directories and delete them:

```
find . –name “chat” –exec rm {}  \;  // be careful when using rm
```

Lab 12.7 - ok

This is similar to -exec except that it asks for confirmation before executing the command, for example the following will ask for delete confirmation each file it finds:

```
touch chat
find . –name “chat” –ok rm {}  \;
< rm ... ./chat >? n
```

Lab 12.8 - diff command

The diff command allows us to compare two versions of the same file, one being an edited version of the other. Diff compares the two files and reports on all lines that are changed, added or deleted.

For example type the following lines into a file called **spot1**:

```
This is my first file
Line 1
```

& the following lines into a file called **spot2**:

```
This is my second file
Line 1
```

Then compare the two by issuing the diff command:

```
diff spot1 spot2


1c1
< This is my first file
---
> This is my second file
```