Lecture 20: Input/ Output Redirection & Filters (contd.)

.Use of grep as a filter

.Finding a particular file in the directory:

```
$ ls | grep "emp.dat"
```

The output of the **Is** command is a list of files in the directory and this output is directly given to the **grep** command as input file. The pattern **"emp.dat"** will be searched in the list of file names produced by the **Is** command.

Note that the symbol "I" **is used to join the two commands in the pipe line.**

.To find out whether a given user has logged in or not:

```
$ who I grep "mano"
```

who command will produce a list of user-names who are currently logged in. This output **of who** will become the input for **grep** and **grep** will search for the pattern **"mano".**

Note that, no error message is produced, if the search using the **grep** command fails.

Classroom Exercise:

Write the command to count the number of persons in the 'CD' department.

.sort

- Arranges the output
- $ sort [- option] [+pos] [-pos] [-0 output file name] [filename]
- options:
  - -d Dictionary order: only letters digits and blanks
  - -n Sort on the first numeric field
  - -r Reverse order of sort
  - -u Eliminate duplicate lines in sorted output
  - -t Specify field separator character (within double quotes after -t)
  - –0 Stores output in a file

The **sort** command takes lines from input file and sorts them, displaying the output on the standard output device. By default the sorting is done on the first field in ascending order.

Each line from the input file is treated as a series of one or more fields separated from each other by a space, or tab by default or any other field delimiter if specified (see example on pg 4.15).

The command line format is:

```
$ sort [option] [filename]
```

sort without a filename takes input from the standard input and arranges each line alphabetically and produces the output on the standard output.

```
$ sort        <Enter>
    Alka
    Lalit
    Bhupinder
```

```
Puneet
Malvika
<CTRL+D>
```

the output will be:

```
Alka
Bhupinder
Lalit
Malvika
Puneet
$
```

Examples:

.To sort a file with a given filename

```
$ cat> namelist        <Enter>
Puneet
Malvika
Lalit
Bhupinder
Alka
<CTRL +D>

$ sort namelist        <Enter>
Alka
Bhupinder
Lalit
```

```
Malvika
Puneet
$
```

.To display the output of the file specified in the reverse order:

```
$ sort -r namelist      <Enter>

Puneeta
Malvika
Lalit
Bhupinder
Alka
$
```

.To sort the file in numeric order rather than ASCII mode, we use -n option (note that this is required when the entries in the number field are not right justified):

```
$ cat >nunbers          <Enter>
6
9
10
2
<CTRL+D>

$ sort -n numbers                <Enter>

2
6
```

```
9
10
$
```

To arrange each record in order of a specific field and not in the usual order of the first character onwards we must specify the field using a command line format as follows:

```
$ sort [ +pos1] [ -pos2] <filename>
```

where

sort skips pos1 fields and stops considering fields after pos2th field.

If pos2 is omitted it extends the fields from pos1 to the end of the line.

Example:

.Sort the file on the third field, skipping the firsttwo fields:

```
$ sort +2       -3 employee    <Enter>
```

.Sorting the employee file on 2nd and within that on 4th field:

```
$sort +2 –3     +3 -4 employee         <Enter>
```

The field separator must be specified while sorting on multiple fields. Sort command has no knowledge about the character by which each field is separated from the other. By default, spaces and tabs are considered as the field separators.

The -t option is used to specify the field delimiter, in case the delimiter is other than a tab or space character.

```
$ sort -t":"      +2 -3 /etc/passwd       <Enter>
```

The passwd file in /*etc* directory has colon (:) as field separator. Therefore, to sort this file on the third field we must specify the field separator as colon.

The output of the sort command is displayed on the terminal by default. In case this output is to be directed to a disk file, the -0 option should be used.

```
$ sort -t ":"      +2 -3 -0 newpasswd    /etc/passwd                <Enter>
```

.Using sort as a Filter

.To display all the logged in users in a sorted manner:

```
$ who | sort               <Enter>
```

.To display the employees belonging to MKT department in a sorted manner from the employee file :

```
$ grep "MKT" employee | sort          <Enter>
```

Here grep will filter out the records of the employee belonging to MKT department and the output will become the input to the sort command.