Linux Shell Commands

Introduction

- Many people says that Linux is a command based operating system.
- So many of us thinks that Linux is not so user friendly OS.
- But it is not true. Linux is a GUI based OS with a Shell which is **more powerful** than its counter part (cmd) in Windows OS.
- We will be familiar with some shell commands.

Identity

- Type ***uname*** and Linux will tell his name to you
- If you want to know your name type ***whoami***

Manual

- For each command Linux contains manual. To view the manual : ***man*** name
  - ***man uname***
  - After showing manuals, prompt does not come automatically
  - Press “q” or “Ctrl+c”, “Ctrl + z” etc to go back to your prompt.

Editors

- To view files a large number of editors are available. They are:
  - kwrite
  - emacs
  - gedit
  - vi
- To view :   ***editorname filename***
  - ***kwrite file.txt***

View Text

- To view a line of text in the shell: ***echo***
  - ***echo 'welcome to linux'***
- To clear the shell : ***clear***

Directory and File Permissions

- Each file or directory has 3 security groups.
  - Owner
  - Group
  - All Others
- Each security group has 3 flags that control the access status : read, write, execute
- They are listed as 'rwx' or a "-" if the access is turned off.
  - ***rwxrwxrwx*** [read, write and executable for owner, group and all others]
  - ***rw-r--r--*** [read and write by owner, read only for group and all others]

Directory and File Permissions

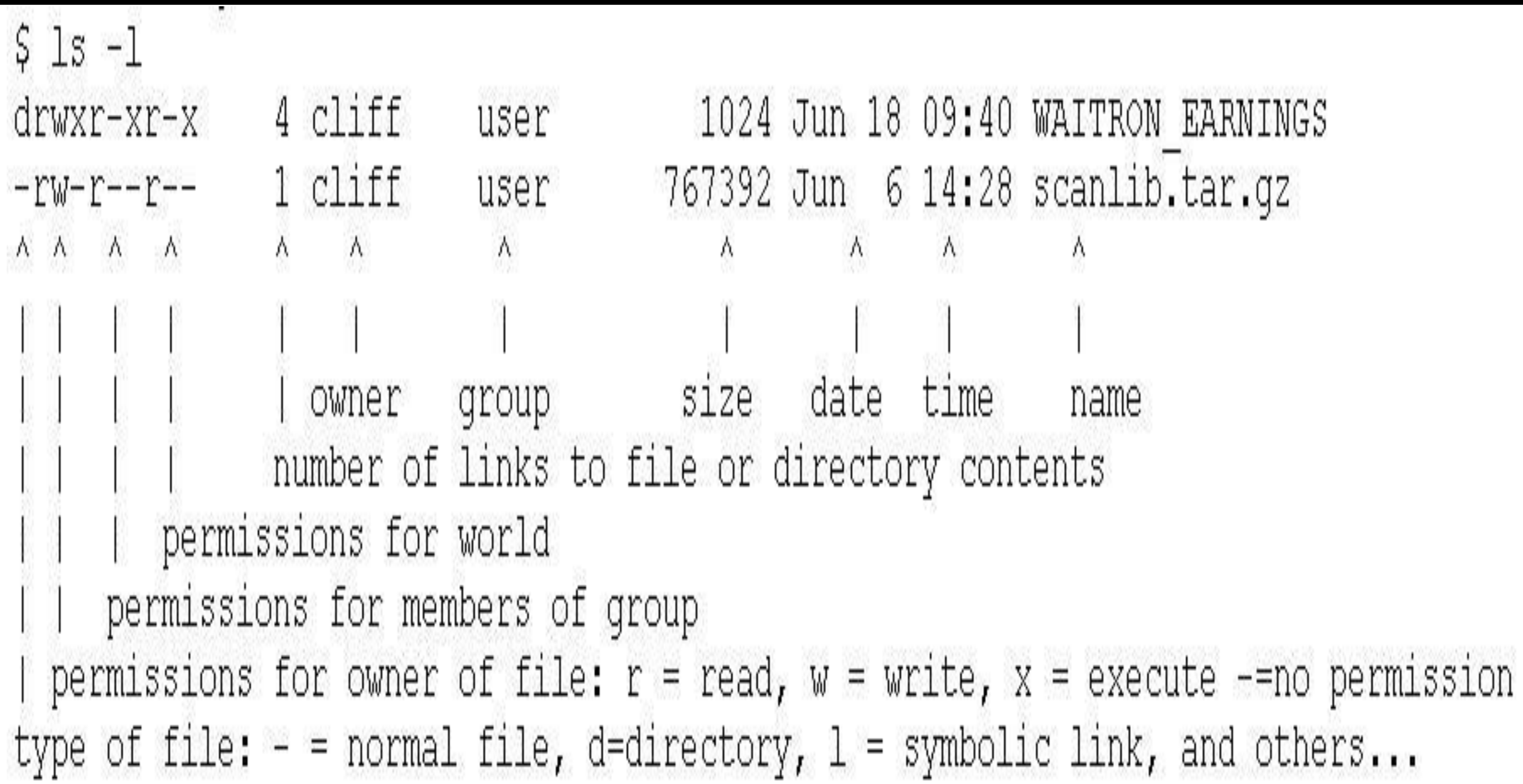
- To change the permissions type ***chmod***
  - u, g, o or all [whose permission you are changing]
  - + or - [type of change: add or subtract permission]
  - combination of r , w or x [which permission you are changing: read, write or execute]
  - file or directory [name of file or directory to change]
    - ***chmod go+rw file1 file2*** add read and write access for group and others for files 'file1' and 'file2'
    - ***chmod a+rwx file1*** add read, write and execute for everyone for 'file1'.
    - ***chmod 555 file1***

Directory and File Listings

- To list information about directory or files : ***ls***
- This command contains some options.
  - -***a*** [do not hide entries starting with .]
  - -***A*** [do not list implied . and ..]
  - -***h*** [print sizes in human readable format]
  - -***l*** [use a long listing format]
  - -***S*** [sort by file size]
  - **Permissions. Directories. Group. Size. Date. Name**

```
drwx------      2          users   4096 Nov 2 19:51  mail/
```

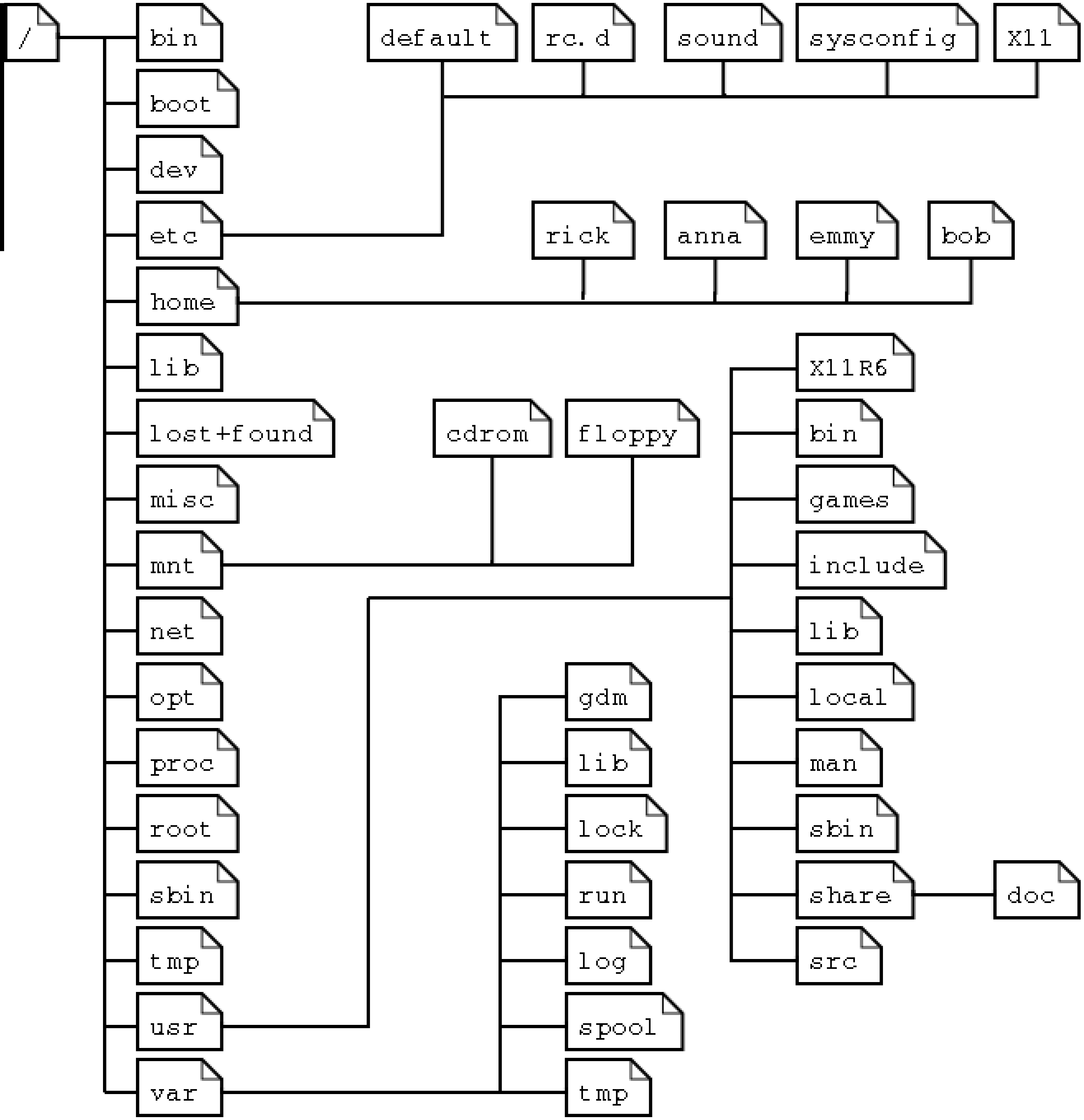
Directory and File Listings



```
$ ls -l
drwxr-xr-x    4 cliff      user        1024 Jun 18 09:40 WAITRON_EARNINGS
-rw-r--r--    1 cliff      user      767392 Jun  6 14:28 scanlib.tar.gz
^ ^  ^  ^     ^   ^       ^           ^      ^    ^      ^
| |  |  |     |   |       |           |      |    |      |
| |  |  |     | owner   group       size   date  time     name
| |  |  |     number of links to file or directory contents
| |  |  permissions for world
| |  permissions for members of group
| permissions for owner of file: r = read, w = write, x = execute -=no permission
type of file: - = normal file, d=directory, l = symbolic link, and others...
```

ls –l illustrated

Directory Structure



Directory Operations

- To print the current directory : ***pwd***
- To change the current directory : ***cd dirname***
    - The variable HOME is the default directory.
- To make a new directory : ***mkdir***
- To delete an empty directory : ***rmdir***

Directory Operations

- To move to a directory pushing the current directory to stack : ***pushd dirname***
- Effect:
  - adds a directory to the top of the directory stack
  - performs a ***cd*** to the new top directory.

Directory Operations

- To moves to the directory at the top of the stack as well as to remove the topmost entry : ***popd***
- Effect:
  - removes the top directory from the stack
  - performs a ***cd*** to the new top directory.

Directory Operations

- To display the list of currently remembered directories : ***dirs***
- The default display is on a single line with directory names separated by spaces.
- How to add to the list : ***pushd***
- How to remove from the list : ***popd***

File Operations

- To copy a file : ***cp***
- Copy source to destination or multiple sources to directory
  - -***i*** [prompt before overwrite]
  - -***r*** [copy directories recursively]
  - -***u*** [copy only when the src file is newer than the dest file or when the dest file is missing

File Operations

- To remove a file or directory : ***rm***
  - -***f*** [ignore nonexistent files, never prompt]
  - -***i*** [prompt before any removal]
  - -***r*** [remove the contents of directories recursively]
  - -***v*** [explain what is being done]

File Operations

- To move or rename a file : ***mv***
  - rename src to dest or move src(s) to directory
  - -***i*** [prompt before overwrite]
  - -***u*** [move only when the src file is newer than the dest file or when the dest file is missing
  - -***v*** [explain what is being done]

File Operations

- To determine file type : ***file filename***
- File tests each argument in an attempt to classify it. This causes the file type to be printed
  - - ***i*** [show the mime type].
  - -***v*** [Print the version of the file]
    - ***file a.txt*** : a.txt: very short file
    - ***file a.xls*** : a.xls: Microsoft Office Document
    - ***file -i a.xls*** : a.xls: \o12- application/msword

File Operations

- To concat files and print on the standard output : ***cat file1 file2 file3 …***
  - -***n*** [number all output lines]

Wildcard

- A single character which can be used as a substitute for a class of characters while searching, highly increasing the flexibility of searching.
- "*"
  - Represents zero characters, all single characters or any string.
  - Example :
    - File * : provides information about all filesystem components.
  - ls * can be combined with some other characters : *.txt, *.jpg
  - ls a* : represents everything that starts with a.
  - ls *a* : represents everything with a in the middle with anything in the start an end.

Wildcard

- “?”
  - Used to represent a single character, any single character.
  - ?? : two single successive character.
  - file ??? : any file component with name of length 3.
  - file ? ?? ??? : any file component with name of length 1/2/3.
  - file a???? : any file component of length 5 starting with a.
  - ls *.??? : list everything having extension of length 3.

Wildcard

- “[ ]”
  - Represents any of the character enclosed by [].
  - ls *[xyz]* : list any object with x/y/z and anything with starting and ending.
  - ls *.[xyz]*
  - ls [a-f]*
  - ls *[0-9]*
  - ls [a-cst]*
  - ls [a-cx-z]*
  - ls abc[0-9] [0-9] [0-9]

Wildcard

- "[ ]"
  - Represents any of the character enclosed by [].
  - ls *[xyz]* : list any object with x/y/z and anything with starting and ending.
  - ls *.[xyz]*
  - ls [a-f]*
  - ls *[0-9]*
  - ls [a-cst]*
  - ls [a-cx-z]*
  - ls abc[0-9] [0-9] [0-9]

File Viewing

- To view files in shell use: ***more*** or ***less***.
  - ***more filename***
  - ***less filename***
- The main difference between more and less is that
  - less allows backward and forward movement using the arrow keys.
  - more only uses the [Spacebar] and the [B] key for forward and backward navigation.

File Viewing

- To output the first lines of files : ***head file1 file2 file3 …***
- Print the first 10 lines of each file to standard output
- With more than one file , precede each with a header giving the file name
  - -***n*** [output the first n lines, instead of the first 10]

File Viewing

- To output the last lines of files : ***tail file1 file2 file3 …***
- Print the last 10 lines of each file to standard output
- With more than one file, precede each with a header giving the file name
  - -***n*** [output the last n lines, instead of the last 10]

File Viewing

- To sort lines of a text files : ***sort file1 file2 file3…***
- Write sorted concatenation of all file(s) to standard output.

File Viewing

- To print the number of lines, words and bytes in files :
  wc file1 file2 file3 …
- print byte, word, and newline counts for each file and a total line if more than one file is specified.
  - -***l*** [print the newline counts]
  - -***w*** [print the word counts]

Standard I/O/E

- By default, three default files known as standard files are automatically opened when a command is executed.
- They are standard input (***stdin***) ,standard output (***stdout***) and standard error (***stderr***).
- For example, the command ***ls -a*** scans the current directory and collects a list of all the files, produces a human readable list, and outputs the result to the terminal window.

Redirection

- Linux redirection features can be used to detach the default files from ***stdin***, ***stdout*** and ***stderr*** and attach other files to them.
- **Input redirection**:
  - < - get input from file instead of the keyboard
- **Output redirection**:
  - > - send output to file instead of the terminal window
- **Append output**:
  - >> - command is used to append to a file if it already exists

Piping

- The input of a command may come from the output of another command.
- This is accomplished with the ‵ | ‵ pipe operator.

- **How to view the lines 15-20 of a file named ‘a.txt’ ?**

Piping

- The input of a command may come from the output of another command.
- This is accomplished with the ' | ' pipe operator.

- **How to view the lines 15-20 of a file named 'a.txt' ?**
  - ***head -20 a.txt | tail -5***

Grep

- grep matches a pattern in a given a list of files or standard input and outputs only the matching lines.
  - ***grep*** **<**pattern> <filename>
    - ***grep abc file.txt***
- grep patterns are case sensitive by default.
- Some options
  - -***i*** [case insensitive search]
  - -***c*** [count of total matching lines]
  - -***E*** [regular expressions can be provided as patterns]
  - -***n*** [display the line numbers of the matched lines]

Find

- search for files in a directory hierarchy.
- By default, find returns all files below the current working directory.
  - ***find***
- To search a pattern : ***find -name '*txt*'***
- To search for a file type :
  - ***find -type d*** [find all directories]
  - ***fine -type f*** [find all regular files]
- Find executes the '***-print***' action by default. To change it to style such as '***ls***' : ***find -type f –ls***

Find

- To search all the directories
  - not recommended
  - ***find / -name "myfile" -type f***
- To search a specific directory
  - ***find /home/dir1 -name "myfile" -type f***
- To search multiple directories
  - ***find dir1 dir2 -name "myfile" -type f***
- To Search for all files owned by a user
  - ***find -user userid***
- To take an action
  - ***find -type f -name '*ch*' -exec chmod a+rwx {} \;***
  - ***{}*** is replaced with the name of the file
  - The ***;*** indiates the end of the command.

Find

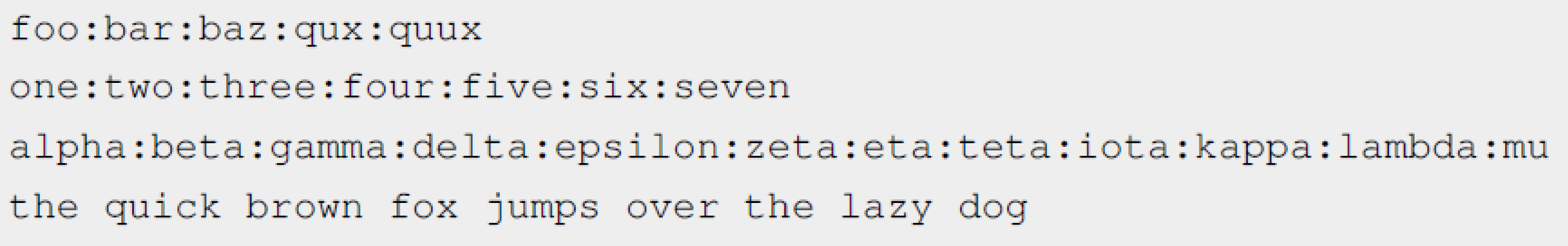
- To search all the directories
  - not recommended
  - ***find / -name "myfile" -type f***
- To search a specific directory
  - ***find /home/dir1 -name "myfile" -type f***
- To search multiple directories
  - ***find dir1 dir2 -name "myfile" -type f***
- To Search for all files owned by a user
  - ***find -user userid***
- To take an action
  - ***find -type f -name '*ch*' -exec chmod a+rwx {} \;***
  - ***{}*** is replaced with the name of the file
  - The ***;*** indiates the end of the command.

Cut

- Extract sections from each line of input
- Extraction of line segments can typically be done
  - by bytes (-b), characters (-c), or fields (-f)
  - separated by a delimiter (-d — the tab character by default).
- A range must be provided
  - N
  - N-M
  - N- (N to the end of the line)
  - -M (beginning of the line to M),
  - where N and M are counted from 1 (there is no zeroth value).

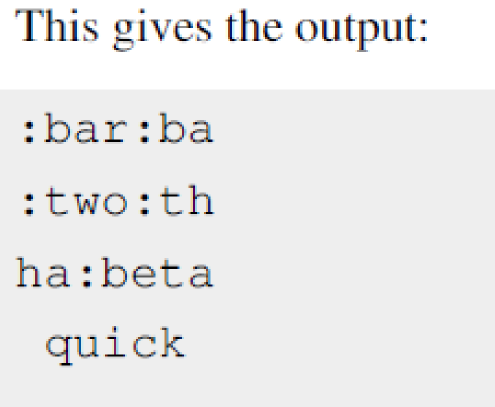
Cut

- Assuming a file named "file" containing the lines:



```
foo:bar:baz:qux:quux
one:two:three:four:five:six:seven
alpha:beta:gamma:delta:epsilon:zeta:eta:teta:iota:kappa:lambda:mu
the quick brown fox jumps over the lazy dog
```

- To output the fourth through tenth characters of each line: cut -c 4-10 file

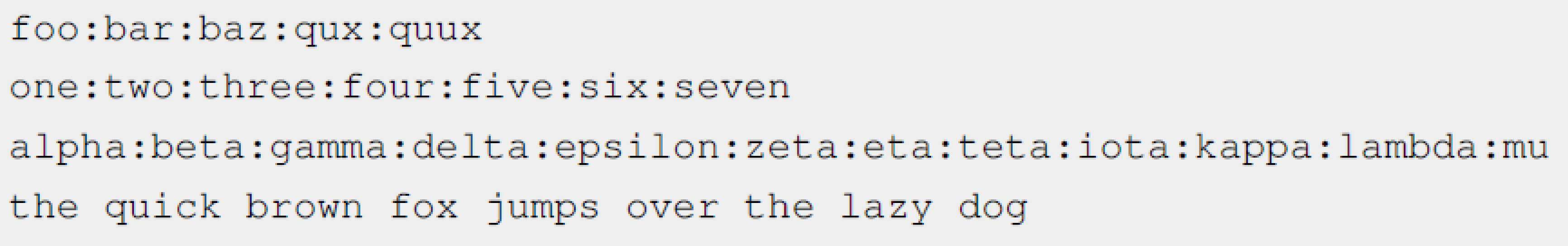


This gives the output:

```
:bar:ba
:two:th
ha:beta
 quick
```

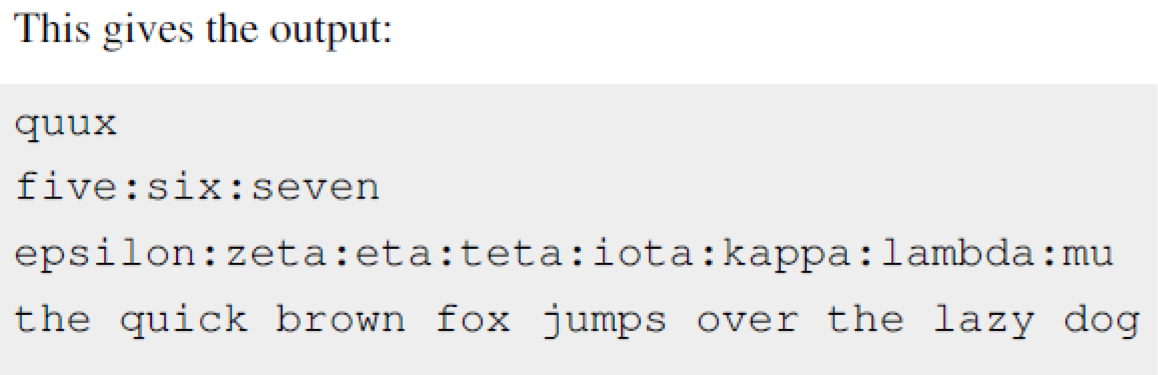
Cut

- Assuming a file named "file" containing the lines:



```
foo:bar:baz:qux:quux
one:two:three:four:five:six:seven
alpha:beta:gamma:delta:epsilon:zeta:eta:teta:iota:kappa:lambda:mu
the quick brown fox jumps over the lazy dog
```

- To output the fifth field through the end of the line of each line using the colon character as the field delimiter:
  cut -d ":" -f 5- file



This gives the output:

```
quux
five:six:seven
epsilon:zeta:eta:teta:iota:kappa:lambda:mu
the quick brown fox jumps over the lazy dog
```

[redacted]

Thanks
for your patience