Working in the shell

Covered topics: **shell variables, exporting variable, builtin commands, commands, aliases, output redirection, command line piping, quoting, command substitution basic grep usage, commands like mv, cp and mkdir.**

Exercise 1: Understanding the Shell
Objective: Familiarize yourself with the shell environment and basic navigation commands.

1. Login to your system.
2. Check which shell you are currently using by executing the command:
```
echo $SHELL
```

3. List your current directory contents using the command:
```
ls -l
```

4. Change to your home directory using:
```
cd ~
```

5. Create a new directory named "test_directory":
```
mkdir test_directory
```

6. Navigate into "test_directory":
```
cd test_directory
```

7. Create an empty file named "example.txt":
```
touch example.txt
```

8. Display the current working directory:
```
pwd
```

Exercise 2: Working with Variables
Objective: Learn how to create and use shell variables.

1. Create a variable called MY_VAR and assign it a value of your choice (e.g., your name).
```
MY_VAR="YourName"
```

2. Print the value of the variable using:
```
echo $MY_VAR
```

3. Create a variable that stores the current date using the date command.
```
CURRENT_DATE=$(date)
```

4. Print the value of the CURRENT_DATE variable.
```
echo $CURRENT_DATE
```

5. Create a variable that holds the path to your home directory.
```
HOME_DIR=$HOME
```

6. Use the variable to navigate to your home directory:
```
cd $HOME_DIR
```

Exercise 3: Creating and Using Aliases
Objective: Define and use aliases to simplify command usage.

1. Create an alias called ll that stands for ls -la. Run the following command:
```
alias ll='ls -la'
```

2. Test your new alias by running:
```
ll
```

3. Create another alias called “space” that tells you how much diskspace you have. Run:
```
alias space='df -h'
```

4. Check if your aliases are set by running:
```
alias
```

5. To make your aliases permanent, open your .bashrc file (or .bash_profile, depending on your shell) in a text editor:
```
nano ~/.bashrc
```
Add your alias definitions to the bottom of the file.

6. Save and exit the text editor. Then, reload your .bashrc file:
```
source ~/.bashrc
```

Exercise 4: Running Common Commands
Objective: Familiarize yourself with common commands to manipulate files and directories.

1. Create a text file named "notes.txt" and add some text to it using echo:
```
echo "This is a note." > notes.txt
```

2. Append more text to "notes.txt":
```
echo "Another note." >> notes.txt
```

3. Display the contents of "notes.txt" using:
```
cat notes.txt
```

4. Copy "notes.txt" to a new file called "notes_backup.txt":
```
cp notes.txt notes_backup.txt
```

5. Change the name of notes_backup.txt to NOTES_backup.txt :
```
mv notes_backup.txt NOTES_Backup.txt
```

6. Delete "notes.txt":
```
rm notes.txt
```

Exercise 5: Combining Commands with Pipes and Redirection
Objective: Understand how to combine commands using pipes and redirection.

1. List all files in your current directory and save the output to a file called "file_list.txt":
```
ls -l > file_list.txt
```

2. Use grep to find files with a specific pattern (e.g., ".txt") in "file_list.txt":
```
grep ".txt" file_list.txt
```

3. Combine ls and grep using a pipe to directly filter the output:
```
ls -l | grep ".txt"
```

4. Sort the contents of "file_list.txt" and save the sorted output into a new file called "sorted_file_list.txt":
```
sort file_list.txt > sorted_file_list.txt
```

1. Create a file called *flist* that holds all the filenames from your logindirectory. (use the **find** command)

2. Enter the following command:

```
echo "a trial" > file1
```

What is the content of *file1*?

Enter the following command:

```
echo "another trial" > file1
```

What is the content of *file1*?

Enter the following command:

```
echo "and another trial" >> file1
```

What is the content of *file1*?

3. Use redirection to get the content of the file *file1* in a new file called *double_text*.

4. To print a file you can use the *lp* command.
   What is the result of the following command:

```
cat /etc/passwd > lp
```

6. What is the result of the following command?

```
find /usr -print > flist
```

Change the command so that all errors are redirected to */dev/null*.

Change the command again to that all errors go to *flist* as well.

(help: man find)

7. Use the *ls* command and commandline piping to view all filenames on the system.

8. The *wc* command list the number of output-characters, words and lines.

Example:

```
wc -l long
```

lists the number of lines.

The *who* command lists the users that are logged in.

Use the who command and the wc command to query the number of users that are logged in at the moment.

Explain the following the results:

1. Enter the following commands:

```
type reboot
type cd
```

2.
```
cd /tmp ; pwd
cd ; pwd
```

3.
```
echo z > file1
echo a >> file1
cat file1
z
a
sort file1
a
z
sort file1 > file1
cat file1
```

4.
```
value=10
echo $value
10
bash
echo $value

exit
echo $value
10
export value
bash
echo $value
10
```

5. PATH=
```
ls
bash: ls: No such file or directory
```

6. history|wc -l
49 *(your outcome may differ)*
HISTSIZE=5
history|wc -l

5

7. `laptop=HP`
   `echo 'my laptop is of the brand $laptop'`
   `my laptop is of the brand $laptop`
   `echo "my laptop is of the brand $laptop"`
   `my laptop is of the brand HP`
   `echo "my laptop is of the brand \$laptop"`
   `my laptop is of the brand $laptop`

8. echo "my username is whoami"
   my username is whoami
   echo "my username is `whoami`"
   my username is rocky
   echo "my username is $(whoami)"
   my username is rocky
   echo 'my username is `whoami`'
   my username is `whoami`
   echo "my usename is $LOGNAME"
   my usename is rocky

9. set|wc -l
   91
   env|wc -l
   22
   number=300
   set|wc -l
   92
   env|wc -l
   22
   export number
   env|wc -l
   23