

EXAM BASIC TOOLS – 2020/10/02

Quentin Le Roux

BASH SHELL

1. What can we use the command "echo" for ?

"echo" is a bash shell built-in command that prints a given input to the terminal (or returns the input when redirected via a pipeline > or >>, e.g. to a file). Examples of inputs can be values (such as "Hello" or 2) and variables (noted with a \$ symbol, variables are evaluated to their assigned value before being printed or returned, etc).

2. What is the difference between ">" and ">>" operators ?

> and >> are output redirection operators. They both redirect the output from a command or program to a file. However, > overwrites the content of the target file (if it exists) while >> appends such content to that of the target file. In the case where the file is non-existent, an empty file of the same name will be created by both > and >>.

3. What is the difference between "cd .." and "cd -" ?

"cd" is an abbreviation for change directory. It is a shell command that, when given the argument "..", will change the current directory to the one above it (e.g. if pwd = "a/b/c", "cd .." will move the current directory up by one to "a/b").

Given the argument "-" the command will change the current directory to the one that was previously selected (e.g. in our example above, "cd -" will move us back the current directory to "a/b/c", entering "cd -" again will move us to "a/b").

4. What folder will you be in after running "cd -" if your current directory is "/home/user" ?

It will depend. If we open a new instance of a command line and type "cd -", we will get the error message "bash: cd: OLDPWD not set" as we have not changed directory yet in the current instance of the shell.

Now, if we are in "/home/user" (for instance after typing "cd ~"), running "cd -" will return us to that previous folder. Example:

```
> cd ~/my_dir #change pwd to /home/user/my_dir
> cd ~ #change pwd back to /home/user
> cd - #change pwd back to my_dir
```

5. How to get the list of all files in the directory "~/my_directory" if your current directory is "/etc/perl/Net" and put this information in the file "file_list" ? (Write your answer in one bash line)

If the file is to be created in "/etc/perl/Net":

```
> ls ~/my_directory > file_list
```

If the file is to be created in "~/my_directory

```
> ls ~/my_directory > ~/my_directory/file_list
```

6. Does the command "chmod g+x my_file.txt" allow to the group to delete the file "my_file.txt" ? Explain why ?

No. "chmod g+x my_file.txt" only changes the file mode (read, write, execute) of the file my_file.txt by adding 'execute' rights to group users. However, execute rights only grant the right to execute the program to the group users.

Since we don't know if the group users have write access to the file and to the folder containing the file (which is how group users are allowed to delete the file), we can't say that they are allowed to delete the file "my_file.txt".

7. How to delete (in one line) all files except the .txt ones in the directory below ?

```
find <name_of_directory_below> -type f ! -name '*.txt' -delete
```

GIT

1. How can you get the list of Untracked files (and all changes as well) in your local directory ?

Using "git status" will provide a list of Untracked files as well as the changes to be committed (if "git add" has been used since the last commit).

2. Write the list of commands that you would use to add a new file "new_file" to your github repository (remote)

```
> git add <new_file>
> git commit -m "<useful and descriptive comment>"
> git push origin <name_of_the_remote_branch_to_push_to>
```

3. How can you get the full history of your commits in your local directory ?

Depending on the level of detail I want, I would use either "git hist" or "git log".

4. Is it possible to restore an old version of particular file (maybe already deleted) even after few commits? If possible, explain how (the commands)

It is possible.

If the file has been deleted, we can easily search for it using:

```
> git log --diff-filter=D --summary
```

Otherwise we can use:

```
> git log -- <file>
```

This will allow us to identify the commits hashes (e.g. 85cc747...) where the file was deleted OR modified in the repository. We then proceed with:

```
> git checkout <commit_hash>~1 <file>
```

The "~1" indicates we want to retrieve the old version of that particular file as it was before the problematic commit(s) (for instance, if the file was deleted with the <commit_hash>, using "git checkout <hash> <file>" will throw an error report as the file does not exist after that commit. Instead we use the ~1).

SHELL SCRIPT

Write a shell script (.sh) that generates a random integer number between 0 and 100, and asks the user to guess it. On each user try, the program will tell if the number to find is larger or smaller than the one that was typed. Program ends only when user guesses the correct number. Your program must interact with the user at every step.

Please find attached a screenshot of the working program in my terminal as well as the .sh file itself. Please also find a reproduction of the code (without the comments) below:

```
#!/bin/bash

rng_number=$((1 + $RANDOM % 100))

echo "Guess a number between 1 and 100:"

while read user_input; do
    if [[ $user_input -eq $rng_number ]]; then
        echo -e "\nWell done! You guessed $rng_number!"
        break;
    elif [[ $user_input -gt $rng_number ]]; then
        echo -e "\nYour guess is too big. Try again:"
    else
        echo -e "\nYour guess is too low... Try Again:"
    fi
done
```