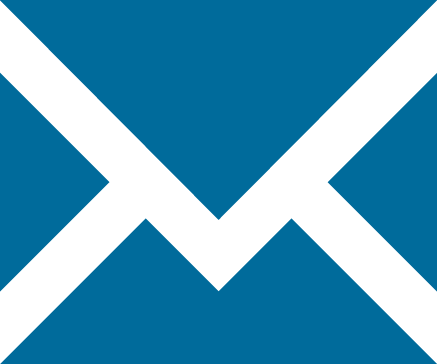
|  |
| --- |
|  |

|  |
| --- |
| **QA Consulting.** |
| Linux Fundamentals – Exercise Workbook |
| Linux System Administration exercises |

#### Prepared by

Devdatta Gonsai

[devdatta.gonsai@qa.com](mailto:devdatta.gonsai@qa.com)

Contents

[Virtual Machine Foundations 1](#_Toc509323131)

[Creating a Virtual Machine 1](#_Toc509323132)

[Working with the Terminal 2](#_Toc509323133)

[Terminal Basics 1 2](#_Toc509323134)

[Terminal Basics 2 3](#_Toc509323135)

[Making Scripts – Basic 4](#_Toc509323136)

[Conditional Statements 6](#_Toc509323137)

[Conditionals - Basic 6](#_Toc509323138)

[Conditionals – Intermediate 6](#_Toc509323139)

[Loops Introduction 7](#_Toc509323140)

[Loops – Basic 7](#_Toc509323141)

[Loops – Intermediate 7](#_Toc509323142)

[Loops – Advanced 8](#_Toc509323143)

[Scripting 9](#_Toc509323144)

[Scripting – Intermediate 1 9](#_Toc509323145)

[Scripting – Intermediate 2 9](#_Toc509323146)

[Scripting – Advanced 1 10](#_Toc509323147)

[Scripting – Advanced 2 10](#_Toc509323148)

[Scripting – Advanced 3 10](#_Toc509323149)

[Scripting – Advanced 4 10](#_Toc509323150)

[Working with Sudo and su 11](#_Toc509323151)

[Sudo and Su - Basic 11](#_Toc509323152)

[Sudo and Su - Advanced 11](#_Toc509323153)

[Redirection and Permissions 12](#_Toc509323154)

[Redirection – Basic 12](#_Toc509323155)

[Redirection - Advanced 12](#_Toc509323156)

[Permissions with chmod – Basic 12](#_Toc509323157)

# Virtual Machine Foundations

## Creating a Virtual Machine

Copy over the Ubuntu virtual disk image from the LocalInstall folder to your desktop. Once this is done, ensure you have VirtualBox installed on your machine. If this is not the case, the installer is found within the LocalInstall folder.

Create a Linux virtual machine within VirtualBox. Give it a name, and ensure that is of type “Linux” and it’s **Ubuntu 64-bit**. It needs to be an Ubuntu image and requires the following specifications:

* Memory Size – 4096 MB
* Create a new virtual hard disk of type VDI
* 50.0 GB hard disk space - dynamically allocated

Once this is done, you will need to open the settings for your newly created virtual machine and add the following configurations:

* Storage – under the IDE Controller in the Storage Tree, there should be an empty disk – select this, and click the disk image in the right pane. Choose the virtual disk image you copied from the LocalInstall.
* Network – ensure that Adapter 1 is enabled and attach it to Bridged Network. Re-initialise the MAC address under the ‘Advanced’ drop down with the green arrows.
* Shared Folders – create a folder on your Desktop named ‘shared’. Add the path to this folder and name it ‘shared’ when you click the folder with a green plus sign to add a share.

Once this is done, double-click your machine to run it through the install process. A couple of things to note:

* Ensure you erase the disk on the virtual machine for the install
* Set region to United Kingdom/London
* Set the root user as admin[your initials] with the password of ‘Pa$$w0rd’

# Working with the Terminal

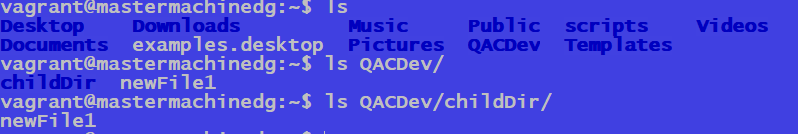
## Terminal Basics 1

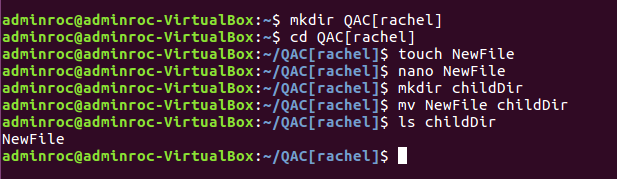
Hint: you may need the commands cd, mkdir, touch, cp and nano

Create a new directory within your current working directory and name it ‘QAC [your first name]’.

Go into this new directory and create a new file. Use a text editor to add some text, save and close.

Create another directory within your QAC directory and name it ‘childDir’. Copy the file you created into this new directory within the QAC directory.





## Terminal Basics 2

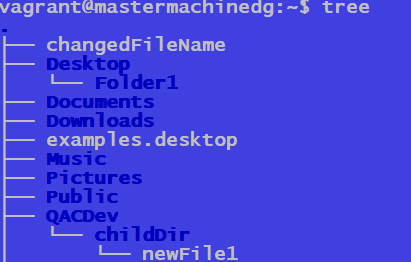
Hint: you may need the commands cd, cat, apt-get and mv

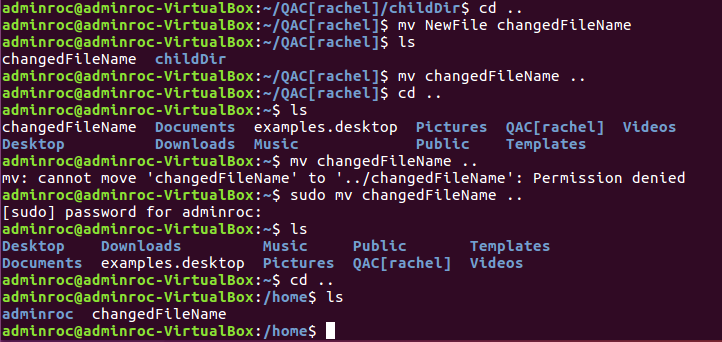
Print the contents of the file you created without going into the file to edit it.

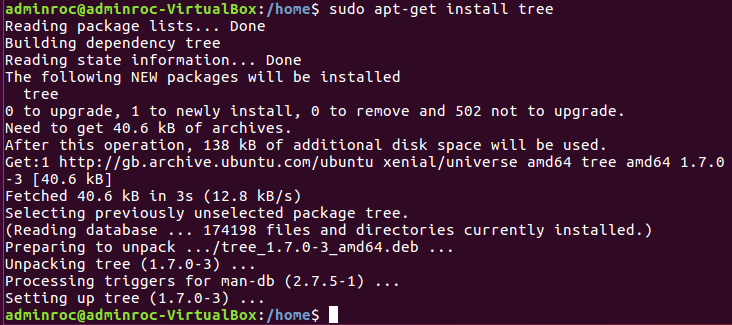
Change the name of this file to ‘changedFileName’ and move it to the home directory.

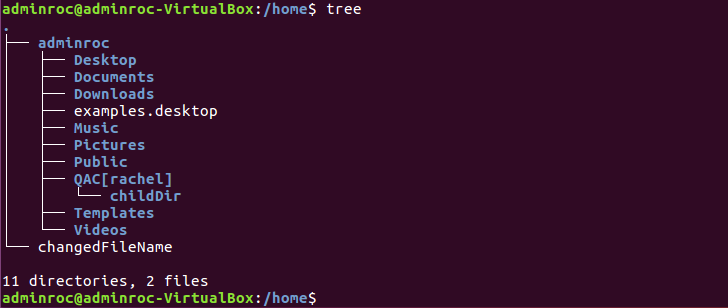
Navigate to the home directory, and print out your current working directory.

Install the tree package for Ubuntu. Print out the structure of your folders with tree package.





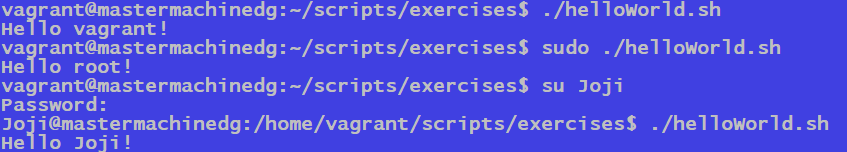


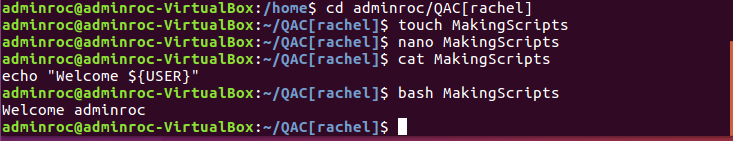


## Making Scripts – Basic

Write a script that will say “Hello [user]!”, where user is the current user running the script.

Hint: create a script and use the echo keyword

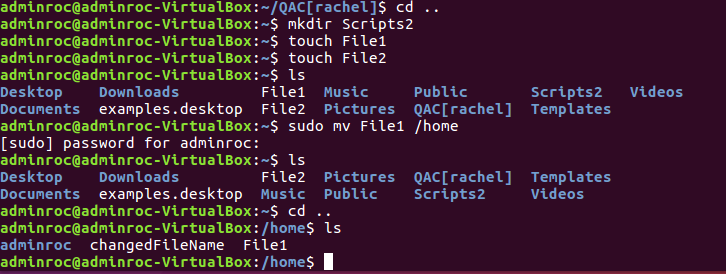




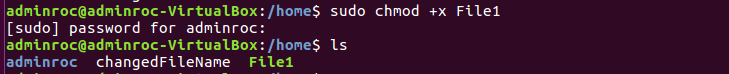
Next, create a script that automates the following:

* Creation of a new directory, and the creation of 2 text files within this
* Move one of the files within the directory to your home directory

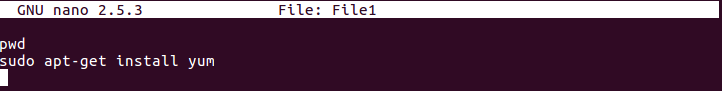
directory



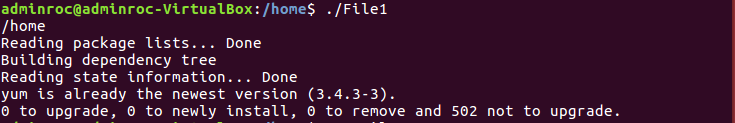
* Change the type of the second text file to a script



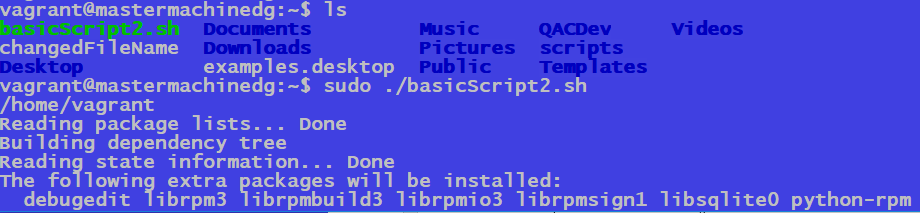
* Add commands to this file to print the working directory and download the yum package manager through apt



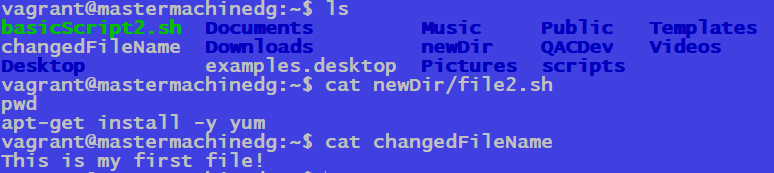
* Execute the second file (./ runs the file)



Hint: Use mkdir, mv, pwd, apt-get, chmod and dot slash notation



…



Sudo touch scriptExample.sh (.sh is not needed but is useful for the user, that it is a script file)

Sudo nano scriptExample.sh (to edit it)

~!/bin/bash (at top of the nano screen, all the bash commands will reside here)

echo “hello world”

To make a file executable we use sudo chmod +x scriptExample.sh (allows us to change the mode, to be readable, or users to write or execute it. +x adds the executable permission to this file) (a+x would give permission to every single user)

Sudo ./scriptExample (now allows us to run command)

Will now print “hello world”

We can also use sudo bash scriptExample to run command

Grep search for particular commands/strings within a file?

Sudo cat /etc/group | grep sudo (only want to search within the group of those that are sudo)

Use pipe key to take output of one command to be used by another command. Taking standard output of etc/group file (using cat) and then passed it onto a new command (grep sudo)

Sudo su Steven (changes user to Steven)

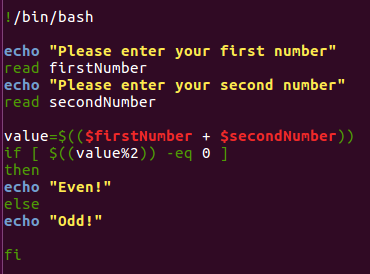
Redirection-

Standard output of a command is the terminal. We can take an output (of ls) and change standard output (that was the terminal) into a file by using >>

Ls >> file1

## Conditionals – Basic and Intermediate

Write a script that takes a user input and outputs whether it is an odd or even number. Extend upon your previous script so that it takes two values and adds them together. From this, the script should output whether the sum of the two values is odd or even

. 

# Loops Introduction

## Loops – Basic

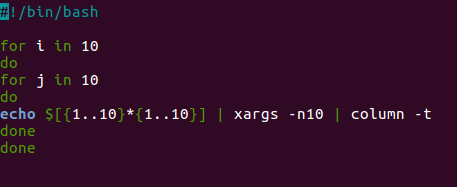
Create script that implements a while loop which will take a user input of a number and output all numbers from 1 to the inputted value.

Hint: Use read and a while loop



## Loops – Intermediate

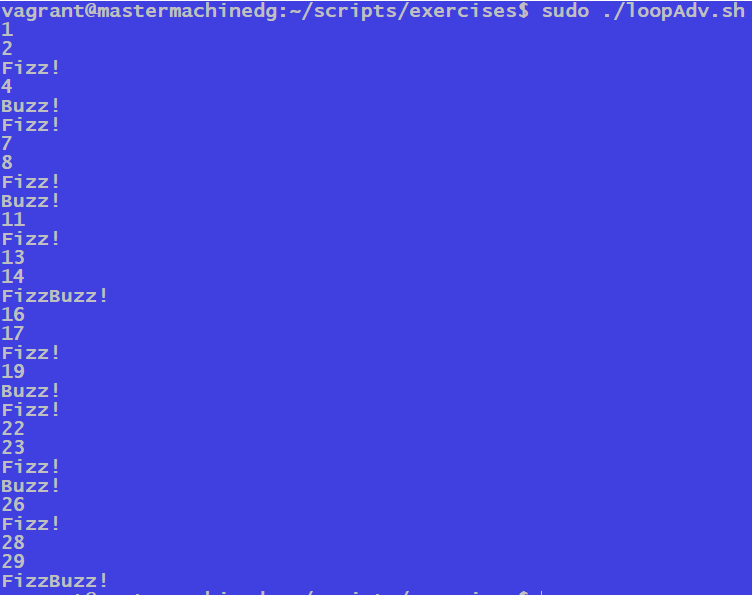
Create a script that implements a for loop, which will print out the times tables from 1 to 10. Have a go into formatting the table too so it is presented a little nicer!



xargs splits the result list into sublists and calls utility for every sub-list

## Loops – Advanced

Create a script that implements an array, which will hold values from 1 to 30. From here, loop through all items in the array. If a number divides by 3, echo “Fizz!” instead of the number. If the number divides by 5, echo “Buzz!” instead of the number.



Once you have finished this, take it further by asking for a user input rather than having the limit of 30 in the script.

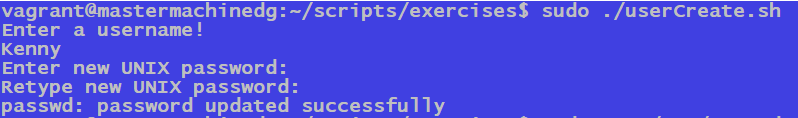
# Scripting

## Scripting – Intermediate 1

Create a script file that will take user input for a username and password, create a new user based on this information.

From here, test the user has been created by first checking the /etc/passwd file for the user, and then switching to the user

Hint: Use read, useradd, passwd, grep and su

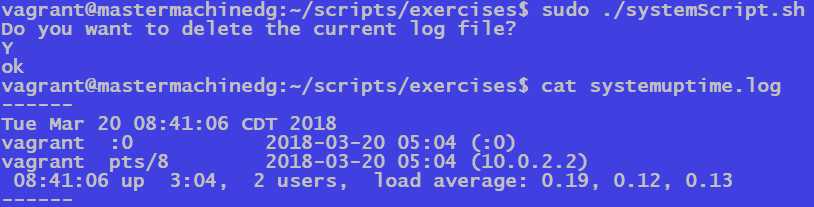


## Scripting – Intermediate 2

Write a script that upon invocation shows the time and date, lists all logged-in users, and gives the system uptime. The script then saves this information to a logfile.

Add to this script by making the log look a little nicer when adding content to it. Add functionality to ask the user whether they want to delete the previous log file – if they answer yes, then delete the file if it’s there.

Hint: Use who, date, redirection and conditional statements



## Scripting – Advanced 1

Create a script that prints out all the prime numbers between 1 and a given user input. The script should prompt for an input and based on this, print all prime numbers between 1 and that input.

## Scripting – Advanced 2

Create a script file that counts the number of lines of a file specified through user input and log it into a new file created within the script named [user specified file name] \_log.txt.

## Scripting – Advanced 3

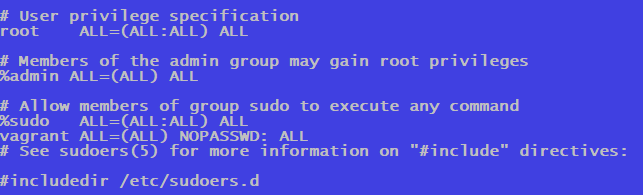
Create a bash script that emulates the gameplay of Rock, Paper, Scissors against the machine. This may include if statements, reading user input, any type of loops, case statements, and any other functions/commands you may be aware. For a suitable random number generator, investigate the use of $RANDOM or shuf. Once this is done, extend the game by making it a 2-player interactive game rather than against the machine.

## Scripting – Advanced 4

Create a script that emulates the gameplay of Tic Tac Toe. Like the previous Rock Paper Scissors exercise, this will include conditionals, loops, read and functions.

# Working with Sudo and su

## Sudo and Su - Basic

1. Log in as root using the su command
2. Run the command visudo – this will allow you to edit the /etc/sudoers file. We can give sudo access to a user in the sudo group. You should see a screen like the below:  
     
   
3. Add a new user with your user adding script created in the previous exercise.
4. Now, add the new user to the sudo group with the usermod command.
5. Switch to this new user and try and run visudo. If this doesn’t work, use sudo alongside the command. Did this work? Why?

## Sudo and Su - Advanced

Edit the script to allow new users to be added to the sudo group. Run the script once more to create a user

It should now create the user with a password, as well as add them to the sudo group.

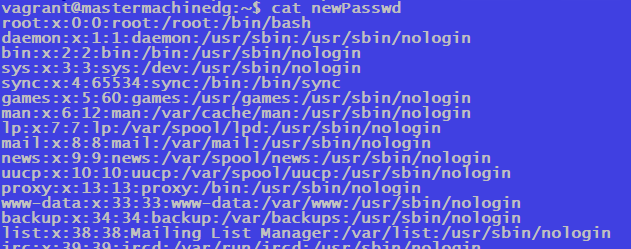
Confirm this in your script by outputting all people in the sudo group onto the terminal.

# Redirection and Permissions

## Redirection – Basic

Read the contents of the /etc/passwd command and redirect it to a file called newPasswd in your home directory.

The new file you created should have an output similar to the below:



## Redirection - Advanced

Write a script that automates this process – it should append any new users onto the file, rather than constantly write the same content into the file every time the script is ran.

## Permissions with chmod – Basic

Write a script that will copy the /etc/group file into your home directory and check the permissions on it. The script will then remove read permissions from the copy in your home directory and run the command to count lines in the copy.