# **Computer Architecture and Technology Convergence Assignment**

**Student Name:** Sarah McNelis **Student Number:** G00398343

Table of Contents

[**Computer Architecture and Technology Convergence Assignment** 1](#_Toc70767852)

[Q1: Binary Arithmetic 2](#_Toc70767853)

[Q1.1 2](#_Toc70767854)

[Q1.2 2](#_Toc70767855)

[Q1.3 4](#_Toc70767856)

[Q1.4 5](#_Toc70767857)

[Q1.5 6](#_Toc70767858)

[Q2: Linux Assignment 7](#_Toc70767859)

[Q2.1: 7](#_Toc70767860)

[Q2.2 21](#_Toc70767861)

[Q2.3 42](#_Toc70767862)

[Q2.4: 47](#_Toc70767863)

[**References** 54](#_Toc70767864)

## Q1: Binary Arithmetic

### Q1.1

When adding two binary numbers there are only three possible outputs, 0, 1 or 2. As explained in *How to Add Binary Numbers* (2021). Wikihow. Available at: <https://www.wikihow.com/Add-Binary-Numbers#:~:text=%20Adding%20Binary%20Numbers%20Using%20Place%20Value%20,or%203%20%28if%20you%20carried%20from...%20More%20> [Accessed April 12, 2021] when the sum is 0 then you write 0, when it is 1 then you write 1, however, when the sum is 2 you write the equivalent binary number for 2 which is 10. Therefore, 1 and 1 is 0 with a carry of 1. Similarly, if you were adding 1 and 1 and there had also been a carry of 1 the answer would be 3 which in binary is 11. Please find my addition table below.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **OVERFLOW** |  |  |  |  |  |  |  |  |
| **CARRY** |  |  |  |  | 1 | 1 |  | 1 | 1 |  |
| **O1** |  |  |  |  |  | 0 | 1 | 0 | 1 | 1 |
| **O2** | + |  |  |  |  | 1 | 1 | 0 | 1 | 1 |
| **SUM** | = |  | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

### Q1.2

For this question I am using what is known as the ‘double-dabble’ method for converting decimal numbers into binary numbers. As explained in *How to Convert Decimal Number into Binary* (2020). Electricalbaba. Available at: <https://electricalbaba.com/how-to-convert-decimal-number-into-binary/#:~:text=An%20easy%20method%20of%20converting%20a%20decimal%20number,remainder%20after%20each%20division%20in%20the%20reverse%20order> [Accessed April 12, 2021] and by *RapidTables.* Available at: <https://www.rapidtables.com/convert/number/how-decimal-to-binary.html> [Accessed April 12, 2021] this is done by progressively dividing the integer by 2 and recording the carry each time. This process is repeated until the integer reaches 0. Then you select the remainders/carry in reverse order from the most significant bit (MSB) to the least significant bit (LSB). Next step is then converting the positive binary number into a negative binary number. This is done by inverting the number and adding one as described by Beck, K. (2020) *How to Convert Negative Numbers to Binary.*  Sciencing. Available at: <https://sciencing.com/convert-negative-numbers-binary-5124016.html> Accessed [April 12, 2021].

#### Q1.2.1 First Calculation: -31

1. I am starting off by converting the number as a positive integer. I then divided this number by 2 to get the base 2 or binary number. I continue to do this until the integer reaches zero. Then I combine the remainders starting from MSB to LSB. I should note here that I have added three zeros to the beginning of this binary number to make up 8 bits.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Decimal** | **Divisor** | **Answer** | **Remainder** | **Digit Significance** |
| 31 | 2 | 15.5 | 1 | LSB |
| 15 | 2 | 7.5 | 1 |  |
| 7 | 2 | 3.5 | 1 |  |
| 3 | 2 | 1.5 | 1 |  |
| 1 | 2 | 0.5 | 1 | MSB |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +31 | = | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  | MSB |  |  |  | LSB |

1. Next step is converting the positive binary number to a negative binary number. This is done by inverting the numbers and adding 1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +31 | = | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|  |  | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Inverted Number | = | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Add 1 | + |  |  |  |  |  |  |  | 1 |
| -31 | = | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| Answer: -31 = 11100001 | | | | | | | | | |

#### Q1.2.2 Second Calculation: -59

1. Again, I am starting off by converting the number as a positive integer. I then divided the number by 2 to get the base 2 or binary number. I continue to do this until integer reaches zero. Then I combine the remainders starting from the most significant bit (MSB) to the least significant bit (LSB). For this number I have added 2 zeros to the beginning of this binary number to make up 8 bits.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Decimal** | **Divisor** | **Answer** | **Remainder** | **Digit Significance** |
| 59 | 2 | 29.5 | 1 | LSB |
| 29 | 2 | 14.5 | 1 |  |
| 14 | 2 | 7 | 0 |  |
| 7 | 2 | 3.5 | 1 |  |
| 3 | 2 | 1.5 | 1 |  |
| 1 | 2 | 0.5 | 1 | MSB |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +59 | = | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|  |  |  |  | MSB |  |  |  |  | LSB |

1. Next step is converting the positive binary number to a negative binary number. This is done by inverting the numbers and adding 1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +59 | = | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
|  |  | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| Inverted Number | = | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Add 1 | + |  |  |  |  |  |  |  | 1 |
| -59 | = | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Answer: -59 = 11000101 | | | | | | | | | |

### Q1.3

I would interpret the bit pattern 11101001 as a negative binary number. As stated in *Negative numbers: Two’s complement.* BBC. Available at: <https://www.bbc.co.uk/bitesize/guides/zjfgjxs/revision/5#:~:text=The%20smallest%20positive%20number%20is%20the%20smallest%20binary,use%20two%27s%20complement%20to%20create%20negative%20binary%20numbers> [Accessed April 12, 2021] negative numbers start with integer 1 and positive numbers start with 0. Furthermore, this webpage explains that the bit at the far left of the bit pattern or the MSB is used to indicate positive or negative and the remaining bits are used to store the actual size of the number. However, adding two positive binary numbers can often give a negative value. This is known as overflow when the total number is bigger than 8 digits. The overflow digit then gets dropped as the CPU cannot store it anywhere according to *Overflow.* BBC. Available at: <https://www.bbc.co.uk/bitesize/guides/zjfgjxs/revision/3> [Accessed April 12, 2021].

### Q1.4

There are 6 types of logic gates: AND, OR, NOT, NAND, NOR, XOR. Logic gates are the building blocks of an electronic circuit as explained in *Logic Gates,* Tutorialspoint. Available at: <https://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm> [Accessed April 12, 2021]. They can only be distinguished by their shape. In the diagram I can see an XOR gate, a NOT gate and an AND gate.

|  |  |  |
| --- | --- | --- |
| My workings: | | |
| This is an XOR gate. If both signals are the same then the output is 0. Otherwise, the output is 1. | This is a NOT gate. It accepts in one signal and outputs the opposite signal. | This is an AND gate. If both signals are 1 then the output is 1. Otherwise, the output is 0. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1.4 Truth Table** | | | | |
| **X** | **Y** | **X SUM (NOT GATE)** | **B (AND GATE)** | **D (XOR GATE)** |
| 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |

Based on my workings and on the truth table above when both X and Y are 0 then both outputs at D and B are also 0. Similarly, when both X and Y are 1 then both outputs at D and B are also 0. Therefore, the conclusion I have reached is that when both inputs are the same then both outputs are 0. Otherwise, the outputs are 1. I then decided to use a screenshot of the diagram in a google image search. I subsequently discovered that this diagram is of a half-subtractor circuit. A half-subtractor is a multiple output combinational logic circuit where two inputs correspond to two 1-bit binary numbers and the two outputs correspond to the difference bit which in this case is D and the borrow bit at B according to Teja, R. (2021) *Binary Adder and Subtractor*. Electronicshub. Available at: <https://www.electronicshub.org/binary-adder-and-subtractor/> [Accessed April 19, 2021]. Furthermore, Teja (2021) explains that the half subtractor is similar to the half adder except that X is complemented before applied to the AND gate which then implements the borrow output at B.

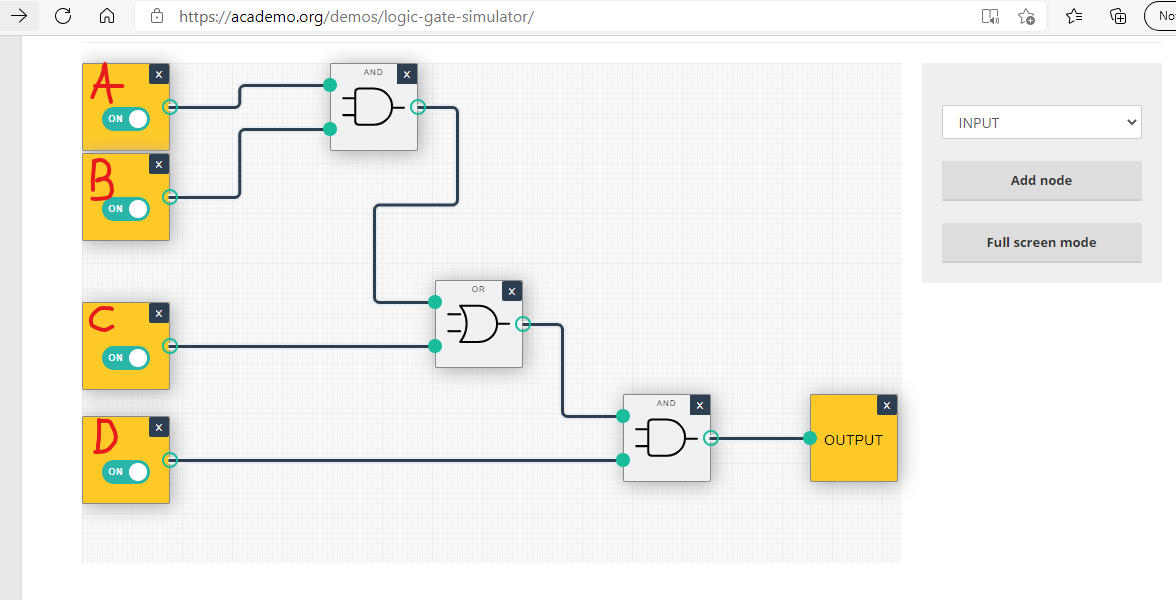
### Q1.5

For this question I have used an online logic gate simulator as suggested from: *Logic Gate Simulator.* Academo. Available at: <https://academo.org/demos/logic-gate-simulator/> [Accessed on April 12, 2021]. I have taken a screenshot of the logic circuit for the Boolean equation: (AB + C)D which I have drawn using the simulator.

My Workings:

1. The first part of this equation is AB. This refers to an AND gate/A.B. This means that if both inputs are 1 the output will be 1. Otherwise, the output is 0.
2. Next we have AB + C. The plus symbol refers to an OR gate which means that if both inputs are 0 then the output is also 0. Otherwise, the output is 1. The two inputs in this case are the sum of AB and C.
3. Finally, the last part of the equation is (AB + C)D. So, the first input is the sum of the OR gate and the second input is D. Again, I am using an AND gate here which will output 1 if both inputs are 1. Otherwise, the output will be 0.

My Diagram for logic equation (AB + C)D:



## Q2: Linux Assignment

### Q2.1:

For this assignment I used the Linux terminal on the AWS VM to input the following commands. This is my interpretation of what each output is. I have also included a screenshot of each command.

1. The echo command is commonly used to display a message or to output results of another command as described by *Echo Command in Linux with Examples* (2019). Linuxize. Available at: <https://linuxize.com/post/echo-command-in-linux-with-examples/> [Accessed April 19, 2021]. In other words, the echo command takes in an argument and prints that exact argument out again. In this case, when I passed in the argument hello world into the linux terminal using the echo command, it printed out hello world.



1. By entering passwd into the command line you are opting to change your password according to Murray, C. (2020) *How to Handle Account Passwords in Linux with Passwd Command*. Linuxhandbook. Available at: <https://linuxhandbook.com/passwd-command/> [Accessed April 19, 2021]. After entering passwd it said changing password for sarahmcnelis, and I was prompted to enter my current password. And then it looks for a new password.



1. The date command displays and sets the systems date/time. When I entered date into the command line, the output displayed the day of the week, the month, the day of the month, the current time, the time zone and the year, as stated in *Date Command in Linux: How to Set, Change, Format and Display Date* (2020). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-date-command> [Accessed April 19, 2021]. As described on this website the date command can also be used to change date/time settings on the system.



1. When I typed hostname into the command line it outputs the hostname of the machine, which in this case is the IP address of the VM that I am using. This command can be used to obtain and set the host, domain, or node name of the current system as said by Emmet (2020) *Using the Hostname Command.* Pimylifeup. Available at: <https://pimylifeup.com/hostname-command/#:~:text=The%20hostname%20command%20can%20be%20used%20to%20retrieve,and%20connect%20to%20machines%20within%20a%20network%20quickly> [Accessed April 19, 2021]. Furthermore, Emmet (2020) explains that you can change your hostname. However, in order to complete this action you must have superuser privileges.



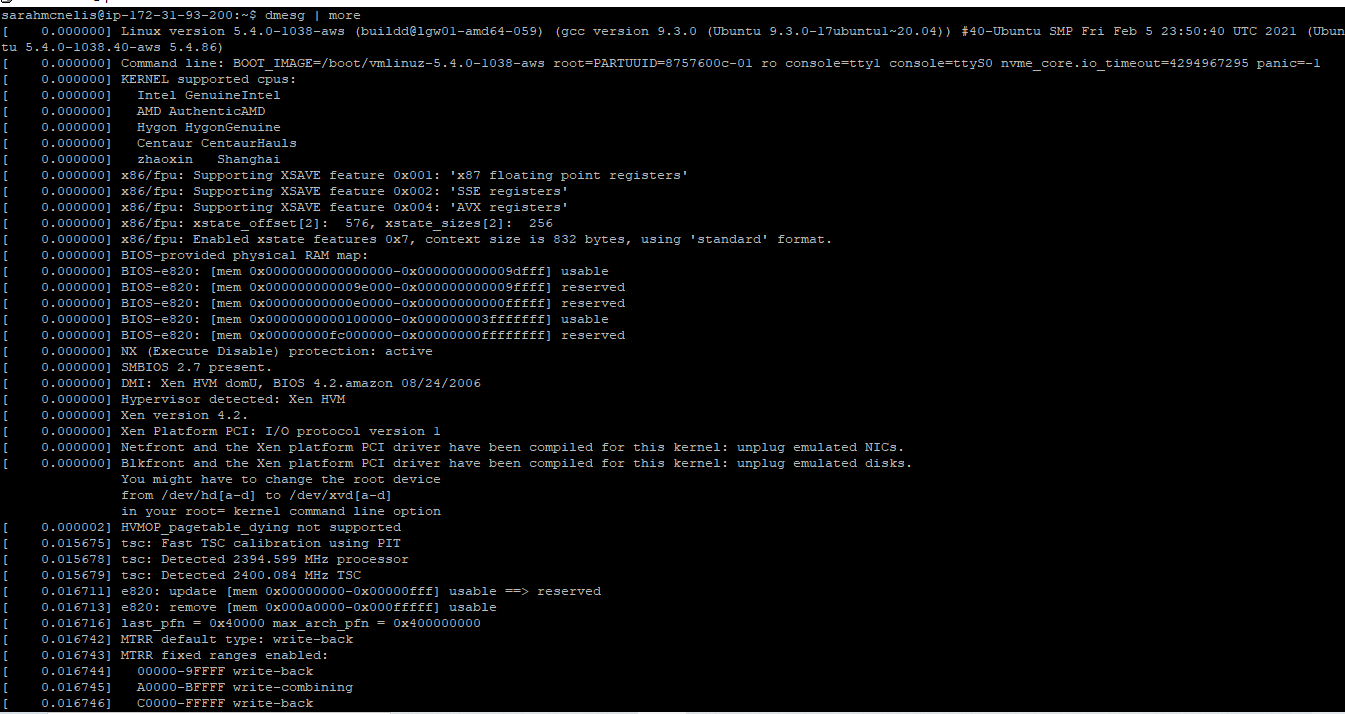
1. The arch command is used to print the computer’s architecture according to *Arch Command in Linux with Examples* (2019). Geekforgeeks. Available at: <https://www.geeksforgeeks.org/arch-command-in-linux-with-examples/#:~:text=arch%20command%20is%20used%20to%20print%20the%20computer,to%20Install%20Tilix%20Terminal%20Emulator%20in%20Ubuntu%2FFedora%2FArch%20Linux%3F> [Accessed April 19, 2021]. As you can see from the image below, the machine I am working on is 64-bit version of the x85 instruction set architecture (ISA).



1. The uname command can be used to find out the hostname of the system, the hardware architectures supported by the currently used kernel and the exact release of your system according to *Uname Command – Print Unix System Information.* Unixtutorial. Available at: <https://www.unixtutorial.org/commands/uname#:~:text=uname%20command%20–%20print%20Unix%20system%20information%201,uname%20-v.%20...%203%20Common%20uname%20usage.%20> [Accessed April 19, 2021]. In this case, typing uname -a into the command line outputs the basic information about my machine. As you can see I am working on a linux terminal from an IP address using Amazon Web Services (AWS), Ubuntu refers to the OS I am using, date, time and year it was created, and the instruction set architecture.



1. The dmesg command displays the messages from the kernel ring buffer according to Shrestha, N. (2015) *7 ‘dmesg’ Commands for Troubleshooting and Collecting Information of Linus Systems.* Tecmint. Available at: <https://www.tecmint.com/dmesg-commands/> [Accessed April 19, 2021]. Shrestha (2015) states that when typing dmesg | more into the command line it will display a log of all loaded drivers in the kernel on a single page. This can be seen in the screenshot below.



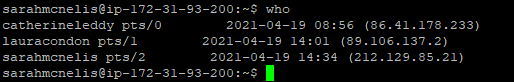
1. The purpose of the uptime command is to find out how long the system has been active. When I entered this into the linux command line it returned the current time, the amount of time the system has been in running state, the number of users currently logged into the system and the load time for the past 1, 5 and 15 minutes as stated in *Linux Uptime Command with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/linux-uptime-command-with-examples/> [Accessed April 19, 2021].



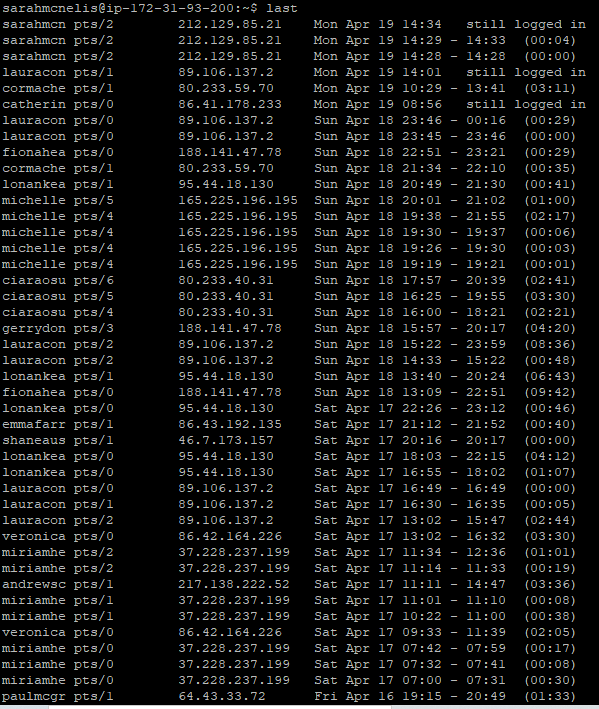
1. The whoami command displays the name of the currently logged in user as explained by *Whoami Command in Linux* (2020). Linuxize. Available at: <https://linuxize.com/post/whoami-command-in-linux/> [Accessed April 19, 2021]. This can be a useful way of knowing which user you are currently working on if you were switching between users for example.

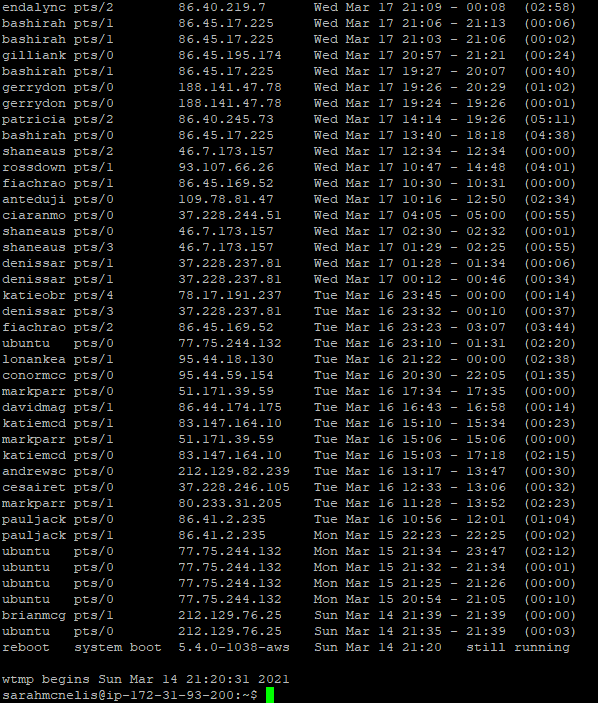


1. The who command displays all the users logged into the system according to *Linux who command* (2019). Computer hope. Available at: <https://www.computerhope.com/unix/uwho.htm> [Accessed April 19, 2021]. I confirmed this when I entered who into the command line. It gave me a full list of the users currently logged in displaying the usernames, lines and time of all the currently logged-in sessions.

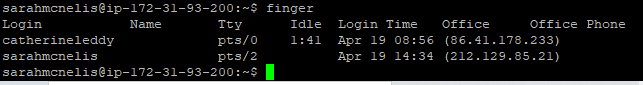


1. The last command gives a list of all the users logged in and out since the file /var/log/wtmp was created as stated by *Last Command in Linux with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/last-command-in-linux-with-examples/#:~:text=The%20last%20command%20in%20Linux%20is%20used%20to,login%20in%20%28and%20out%29%20time%20and%20their%20host-name> [Accessed April 19, 2021]. As the list was quite large I have taken a screenshot of the top and the bottom. From the first screenshot we can see that I have been logged in since 14.34. At the bottom of the second screenshot, we can see that the wtmp was created on March 14.

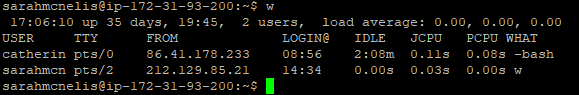




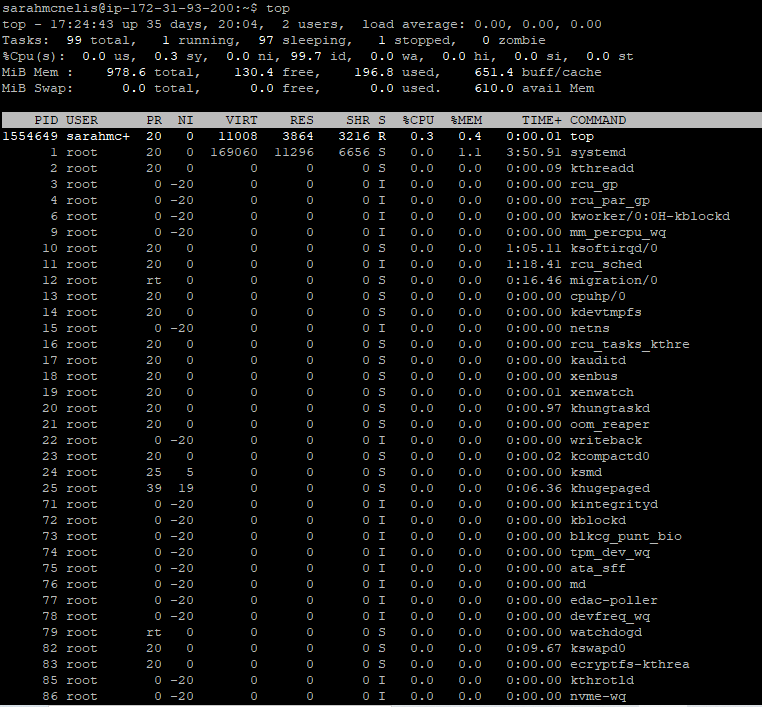
1. The finger command is generally used by system administrators which gives details of all the users logged in according to *Finger Command in Linux with Examples* (2020). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/finger-command-in-linux-with-examples/> [Accessed April 19, 2021]. Geeksforgeeks (2020) further explains that this command also gives details such as login name, username, idle time, login time and in some cases even email addresses. As you can see from the screenshot below it has displayed myself and another user who are currently logged into the VM.



1. When I entered w into the command line it populated a list of users. According to *W Command in Linux* (2019). Linuxize. Available at: <https://linuxize.com/post/w-command-in-linux/> [Accessed April 19, 2021], the w command is a quick way of seeing who is logged into the system and what they are doing. As you can see from the screenshot below, myself and another user are currently logged on and you can see what commands we are running. As you can see my current argument is the w command. Linuxize (2019) explains that the w command shows the username, the terminal being used, the IP addresses/hostname, the time logged on, the time the user last interacted with the terminal, the time used by all processes in the terminal(JCPU), the time used by the user’s current process (PCPU) and what the user’s current process is.



1. The purpose of the top command is to show the Linux processes. As explained by *Top command in Linux with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/top-command-in-linux-with-examples/> [Accessed April 19,2021], this command gives a real-time summary of system information and also outputs a list of processes which are currently managed by the Linux Kernel. This list was quite large, so the below screenshot does not capture the full list. I was also required to press q to quit otherwise it would keep running the command.



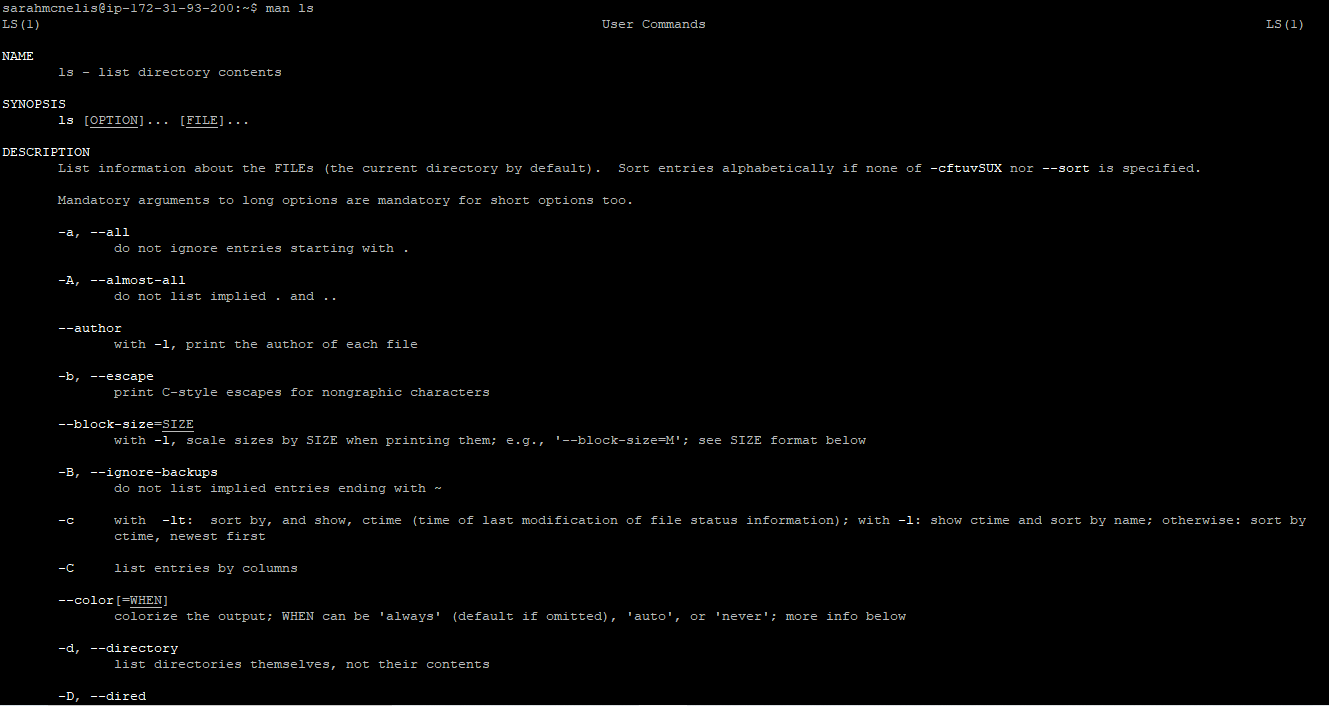
1. The command echo $SHELL returns the value of the shell environment according to *Difference between “echo $SHELL” and “which bash”* (2017). Unix.stackexchange. Available at: <https://unix.stackexchange.com/questions/43499/difference-between-echo-shell-and-which-bash#:~:text=echo%20%24SHELL%20shows%20the%20value%20of%20the%20SHELL,alone%20unless%20it%20is%20unset%20when%20they%20start> [Accessed April 19, 2021]. As you can see from my screenshot below, this command displayed the shell I am using under my username. Mishra, S. (2018) (*#!/Bin/bash) What exactly is this?*. Medium. Available at: <https://medium.com/@codingmaths/bin-bash-what-exactly-is-this-95fc8db817bf#:~:text=%2Fbin%2Fbash%20is%20the%20most%20common%20shell%20used%20as,developed%20and%20better%20syntax.%20WHAT%20IS%20IT%20%3F> [Accessed April 19, 2021], explains that bin/bash/ is the most common shell used by default in linux systems.



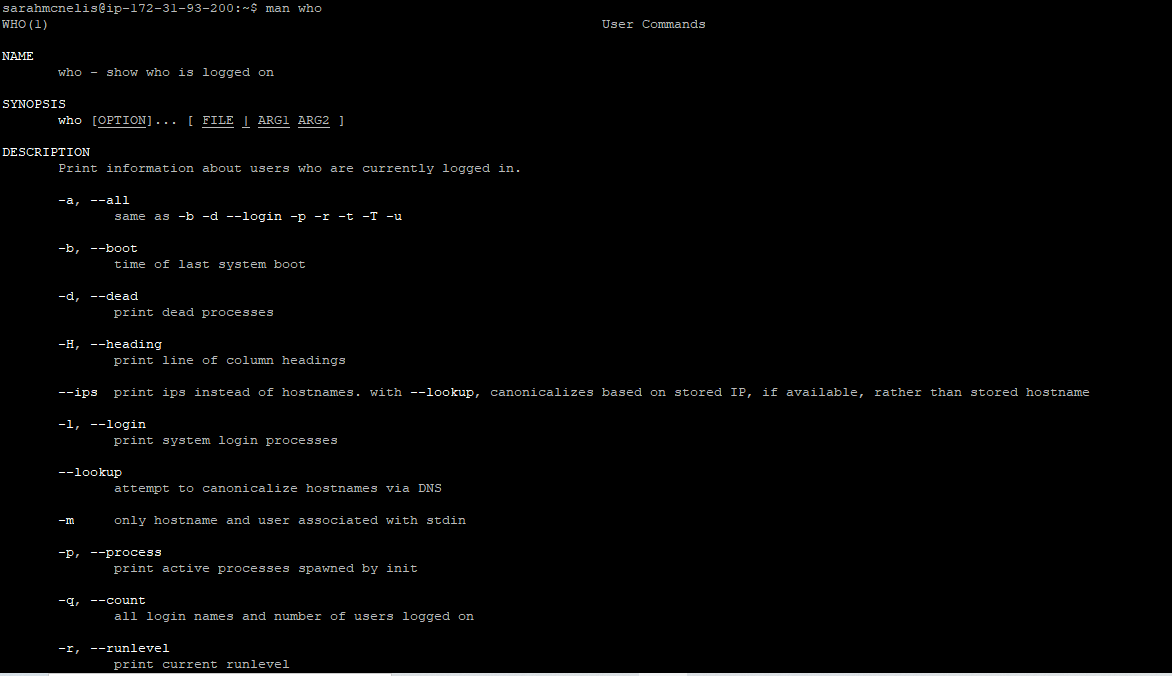
1. Typing echo {con, pre} {sent, fer} {s, ed} into the command line will output a list of words. There is not a lot of documentation online about this particular command, but my interpretation is that it takes the first two elements from the first list and uses the contents of the other two lists to make up a word. It outputs all possibilities that could be created. Therefore, my conclusion would be that by using the echo command to input part of a word will return a number of possibilities of the full word.



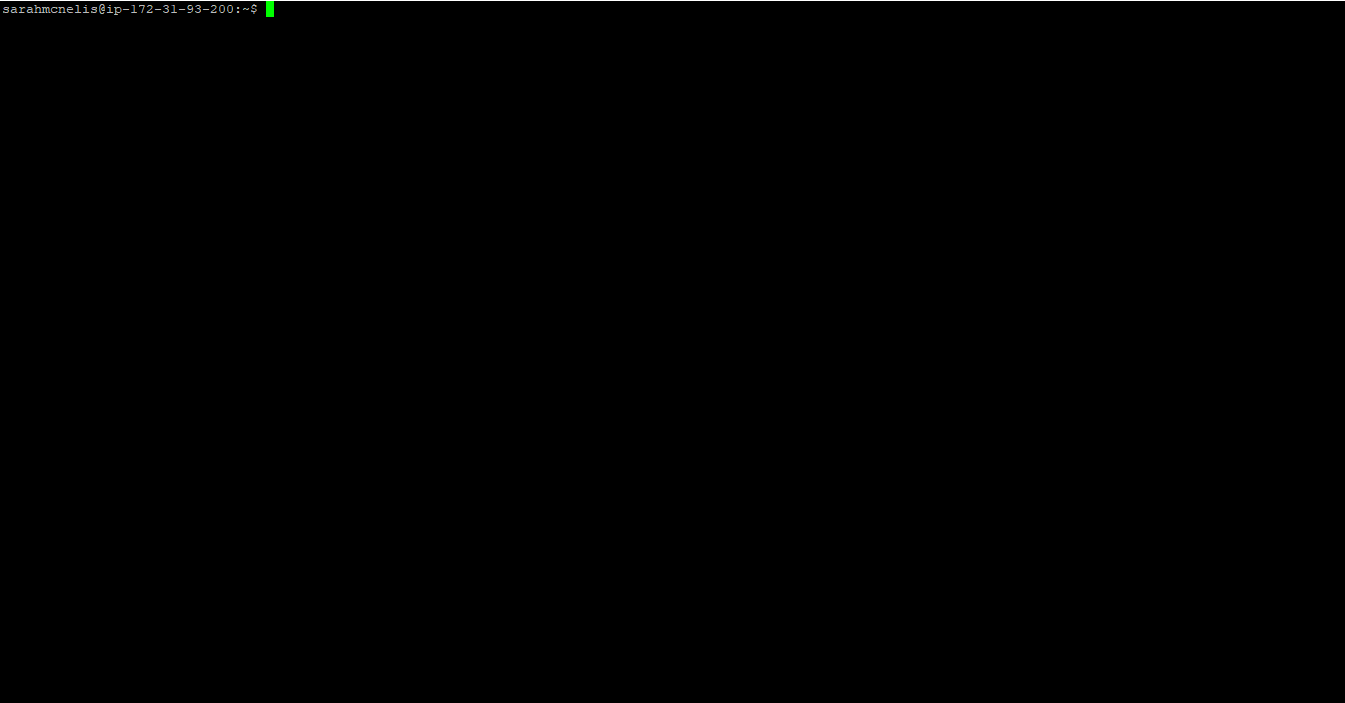
1. The man command is used to display the user manual of any command that can be run in the system as said in *Man Command in Linux with Examples* (2021). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/man-command-in-linux-with-examples/#:~:text=man%20command%20in%20Linux%20is%20used%20to%20display,Every%20manual%20is%20divided%20into%20the%20following%20sections%3A> [Accessed April 19, 2021]. In this case, man ls returns the user manual for the list directory. Again, you have to click q to quit in order to terminate this command.



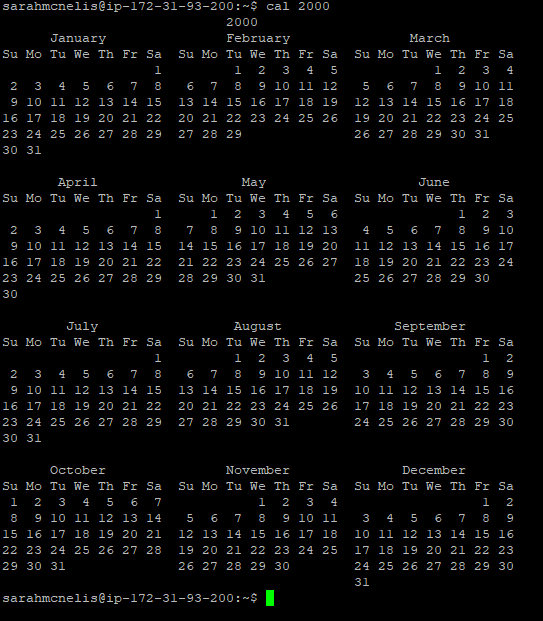
1. Similarly, the man who command gives the manual on who is logged on. According to Kumar, V. (2020) *How to use man command in Linux, important options for beginner.* Cyberpratibha. Available at: <https://www.cyberpratibha.com/how-to-use-man-command-in-linux/#:~:text=How%20to%20use%20man%20command%20in%20Linux%2C%20important,option%20of%20man%20command.%20...%209%20Conclusion.%20> [Accessed April 19, 2021], this manual would include instructions on possible ways and options to use that particular command. This can be seen in the image below.



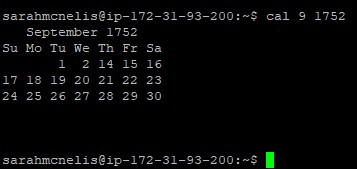
1. The clear command is used to clear the terminal screen as described in *Clear Command in Linux with Examples* (2020). Geekforgeeks. Available at: <https://www.geeksforgeeks.org/clear-command-in-linux-with-examples/> [Accessed April 19, 2021]. This webpage also explains that this command will ignore any command-line parameters that may be present. As you can see from the screenshot below, it cleared my entire screen.



1. When I entered in cal 2000 into the command line it outputs the full calendar for the year 2000. Prakash, A. (2019), *5 Examples of Cal Command in Linux.* Linuxhandbook. Available at: <https://linuxhandbook.com/cal-command/> [Accessed April 19, 2021] said that cal is a handy command with extra options to view a particular date, month or year. If I were to just enter cal it would show me the current month highlighting today’s date.

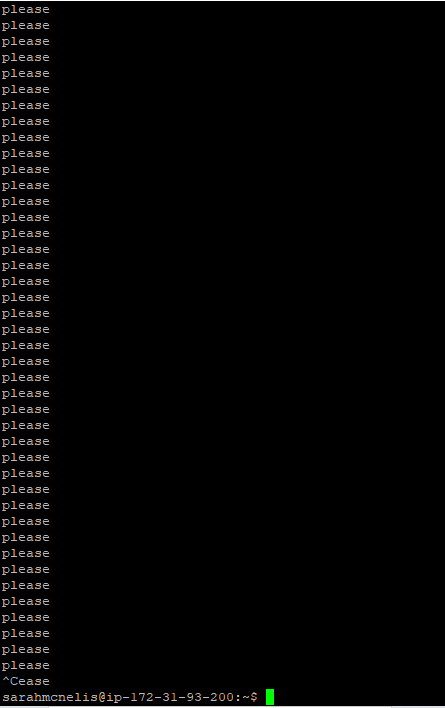


1. Similar to the previous example by entering cal 9 1752 the output will show me the calendar for month 9 which is September from the year 1752. Prakash (2019) also explains that this can be written in three ways; firstly, as I have entered it into the terminal, secondly it can be written as cal sept 1752 and finally it can also be written as cal -d 1752-9. All three options give the same result. However, as you can see from the screenshot below the full month is not printed. The reason for this as explained by *The Missing Eleven Days!* (2013)Projectbritain. Available at: <https://www.projectbritain.com/calendar/September/calendar.html#:~:text=In%201752%20Britain%20decided%20to%20abandon%20the%20Julian,Many%20people%20believed%20their%20lives%20would%20be%20shortened> [Accessed April 19, 2021], is that in September 1752, The British Empire replaced the Julian Calendar with the Gregorian Calendar and as a result lost eleven days. The Julian Calendar consisted of 365.25days which was too long and over time fell out of sync with the seasons. This is why the Gregorian Calendar was introduced as it has a leap year every four years keeping it in line with the seasons.

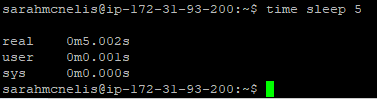


1. McKay, D. (2019) *How to Use the Yes Command on Linux*. Howtogeek. Available at: <https://www.howtogeek.com/415535/how-to-use-the-yes-command-on-linux/> [Accessed April 19, 2021], explains that the yes command behaves as if you were typing yes over and over again and will only cease if you press Ctrl+C. He also states that the system will repeat any word following yes until forced to stop. As you can see from the screenshots below which only show a snip of the pleases’ that appeared on the terminal screen, this is a very effective command.





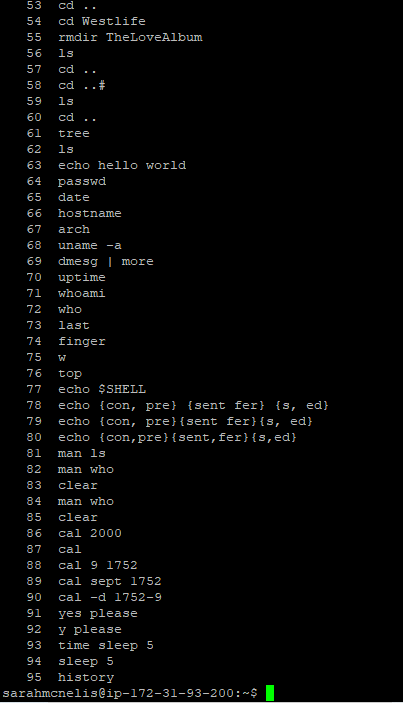
1. The time sleep command is described as a simple command which delays the output. In this case I have used time sleep 5 where it will delay output for 5 seconds. The default is seconds; however, it is possible to specify other time units in the format sleep [number][unit] as explained on *How to use the Linux Sleep Command with Examples* (2021). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-sleep> [Accessed April 19, 2021]. It is important to note here that by starting the command with time it will return a report on how long it takes for the command to execute according to  *Linux sleep command* (2019). Computerhope. Available at: <https://www.computerhope.com/unix/usleep.htm> [Accessed April 19, 2021]. If you just entered sleep 5 it would sleep for 5 seconds and not give a report. This comparison can be seen in the screenshots below.





1. The history command is one of the most useful commands yet. It gives a full report of all previous commands entered into the terminal as stated by McKay, D. (2020) *How to Use the History Command on Linux.* Howtogeek. Available at: <https://www.howtogeek.com/465243/how-to-use-the-history-command-on-linux/#:~:text=%20How%20to%20Use%20the%20history%20Command%20on,the%20history%20list%2C%20type%20an%20exclamation...%20More%20> [Accessed April 19, 2021]. McKay (2020) also explains that this command allows the user to review and repeat commands and is quite useful especially when working with more complex and lengthier commands. As you can see from the following screenshots, I have entered 95 commands into this terminal since creating my account.





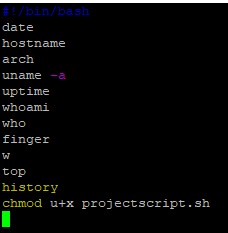
### Q2.2

This question involved carrying out a lot of research on the Vim text editor. First, I discovered how to create a file in Vim. *Unix/linux – The vi Editor Tutorial*. Tutorialspoint. Available at: <https://www.tutorialspoint.com/unix/unix-vi-editor.htm> [Accessed April 26, 2021], explains that in order to create a new file you type “vi filename” into the linux command line.



This then bring you to the file where you can compose the shell script. Zonker (2009), *Vim 101: A Beginner’s Guide to Vim*. Linux. Available at: <https://www.linux.com/training-tutorials/vim-101-beginners-guide-vim/> [Accessed April 26, 2021] gives some basic tips on how to use Vim for example, i for insert and :q for quit.

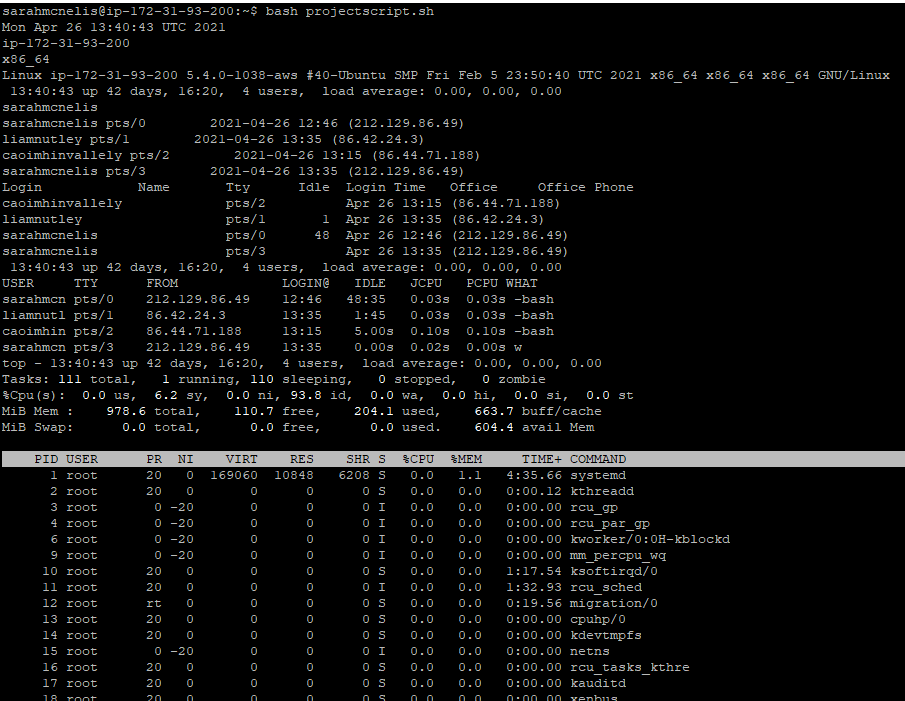
The next step was writing the actual shell script. I learned from *how to automate some linux commands* (2013). Stackoverflow. Available at: <https://stackoverflow.com/questions/14882292/how-to-automate-some-linux-commands> [Accessed April 26, 2021] that in order to create a script the first line of the script should be: #!/bin/bash. This webpage continues to explain that following this first line you should type the commands you wish to include in the script as if you were typing them into the linux command line. Furthermore, once the commands are written in the script the final line needs to allow execute permissions for the current user. This can be done with the simple line of code: chmod u+x filename. U stands for user and x gives execute privileges. The full script can be seen in the image below.



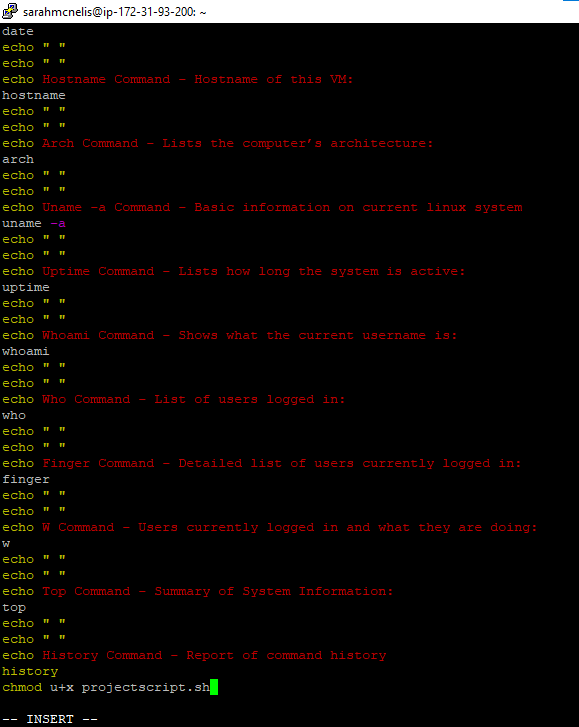
The next step involved running the script. Gite, V. (2021) *How to Run the .sh File Shell Script in Linux.UNIX.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/run-execute-sh-shell-script/#:~:text=%20The%20procedure%20to%20run%20the%20.sh%20file,5%20To%20run%20your%20script%20%3A%0A.%2Fscript-name-here.sh%0AAnother...%20More%20> [Accessed April 26, 2021] describes a number of ways to complete this: 1)./filename.sh 2) sh filename.sh or 3)Bash filename.sh. Kiarie, J. (2021) *How to Create Simple Shell Scripts in Linux.* Tecmint. Available at: <https://www.tecmint.com/create-shell-scripts-in-linux/> [Accessed April 26, 2021] also confirms that all three of these commands will complete the same task. I chose to use option three as I felt it was easier to remember.



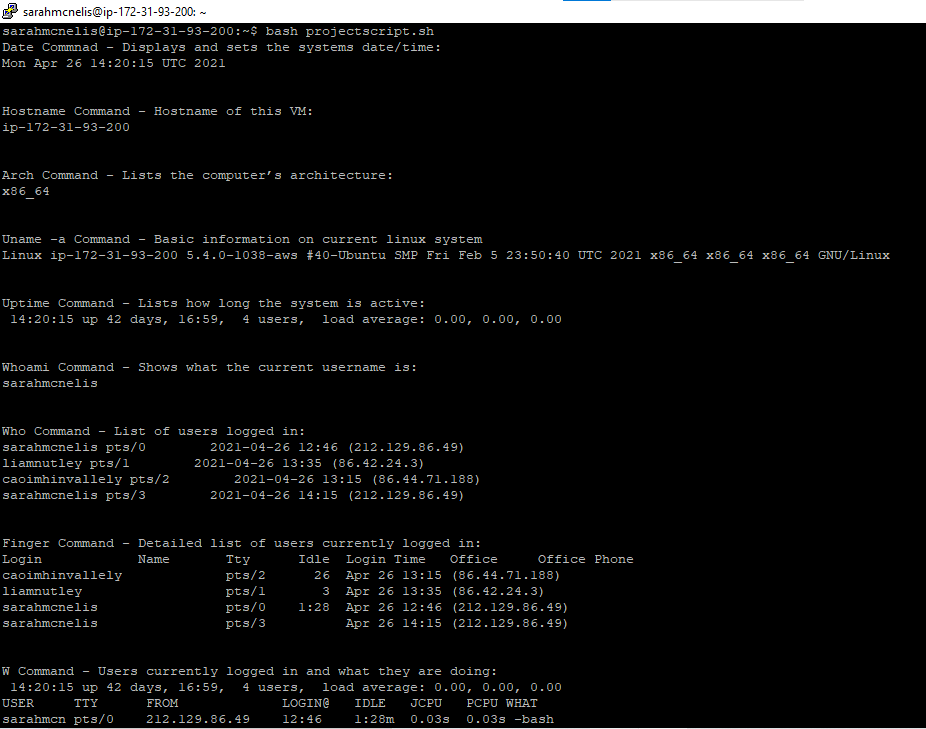
When I entered this into the command line it automatically executed the shell script as seen in the following image.

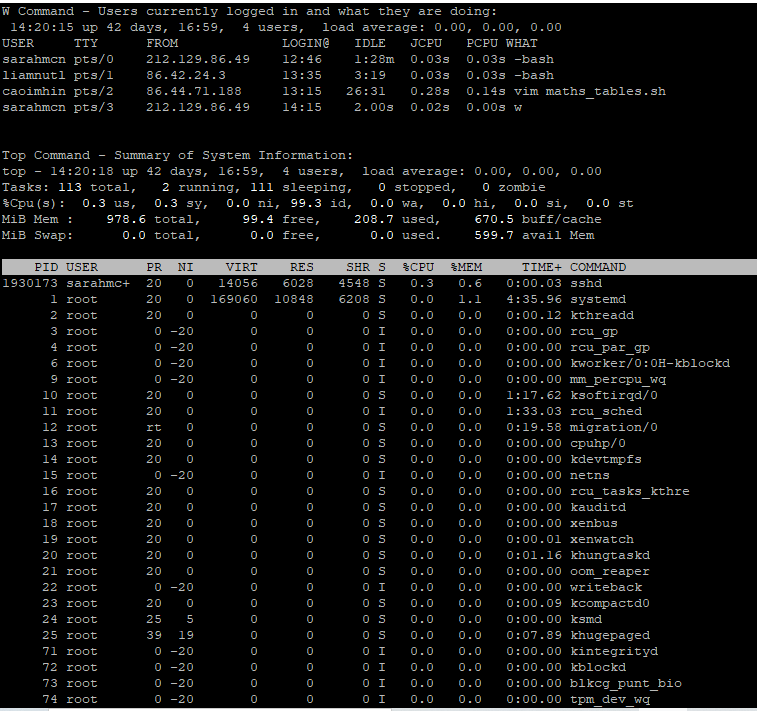


I have now verified that this script does work. However, I am not happy with the output. As you can see from the image above it is quite untidy and difficult to read. Therefore, I have adjusted my shell script to include the echo command in order to give headings to each output and I have added extra spacing to make it easier to read.



This shell script has worked wonders for the output as can be seen in the two screenshots below. It is neater, easier to read and distinguishes what the command is, what it does and what the output of that command is.



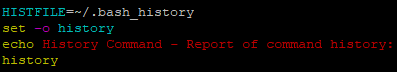


Finally, the last step involves redirecting this output to a text file. For the purpose of this project, I have named the text file: sarahMcnelis.txt. Gite, V. (2009), *Save Bash Shell Script Output to a File.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/bash-script-output/> [Accessed April 26, 2021] explains that this can be done using the redirection operator: > as I have done in the screenshot below.

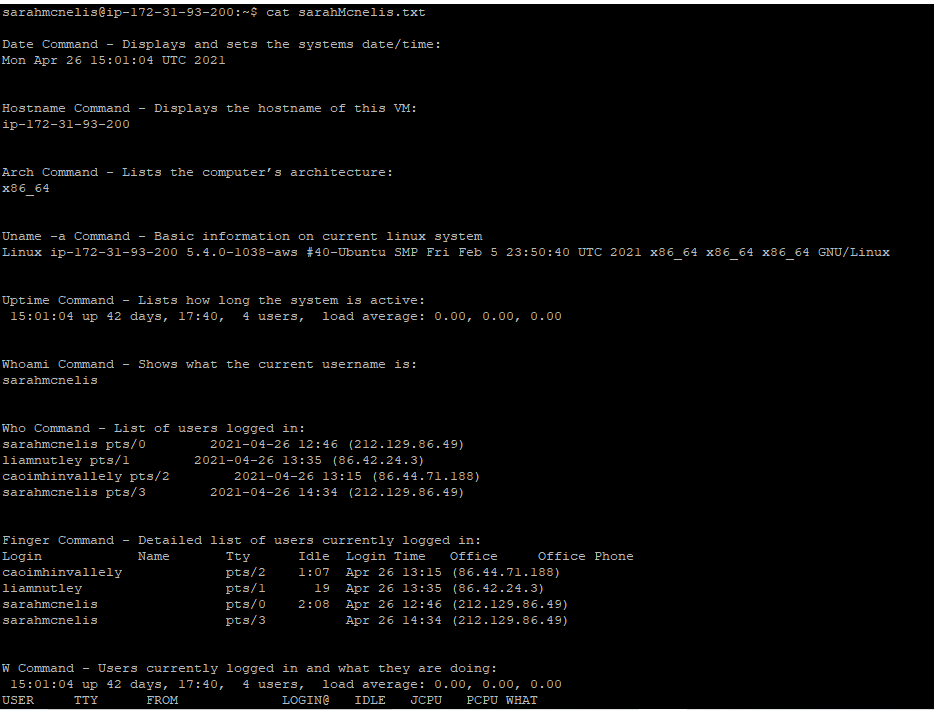


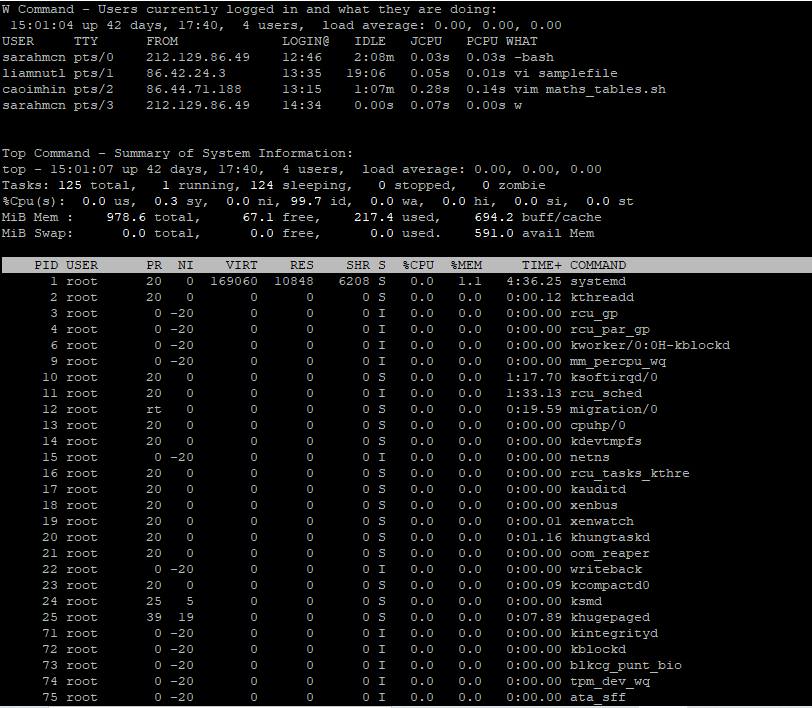
After waiting for a few minutes, I realised I needed to press Ctrl+C in order to stop some of the commands running. Next I decided I wanted to access and view this text file to check if my command had redirected the shell script to the file as I had wanted. Gite, V. (2020), *Linux and Unix Command to View File.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/unix-linux-command-to-view-file/#:~:text=Linux%20And%20Unix%20Command%20To%20View%20File%201,as%20if%20you%20had%20double-clicked%20the%20file’s%20icon> [Accessed April 26, 2021] describes that this can be done from the linux terminal by entering cat filename into the command line.

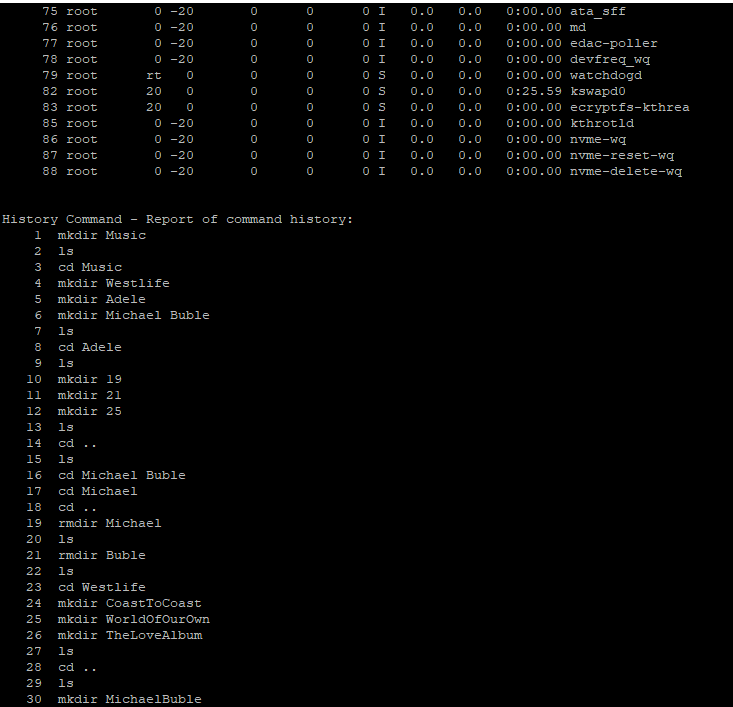
However, after viewing the text file I noticed my history command had not output at all. I double checked my script. And when I confirmed I had included the history command in my shell script, and I double checked the output of the script I realised it had not output this command in the terminal either. After some research, I discovered that bash actually disables history in noninteractive shells by default. In *History command inside bash scripts* (2001). Stackexchange. Available at: <https://unix.stackexchange.com/questions/5684/history-command-inside-bash-script> [Accessed April 26, 2021] I learned that a simple piece of code can change this default setting to allow my script to output the command history. I have added a screenshot below with this extra two lines which I have added to my shell script.

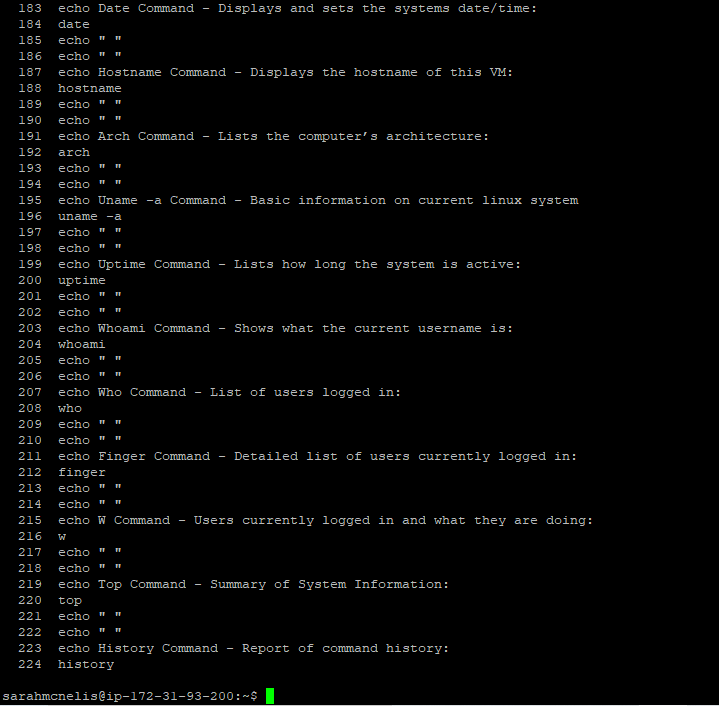


I redirected my shell script to the text file again. I viewed the file again using the cat command and I can now see that the full output of the shell script has been successfully redirected to the text file. This can be seen in the images below.









#### Contents of my text file:

Date Command - Displays and sets the systems date/time:

Wed Apr 28 19:42:54 UTC 2021

Hostname Command - Hostname of this VM:

ip-172-31-93-200

Arch Command - Lists the computer’s architecture:

x86\_64

Uname -a Command - Basic information on current linux system

Linux ip-172-31-93-200 5.4.0-1038-aws #40-Ubuntu SMP Fri Feb 5 23:50:40 UTC 2021 x86\_64 x86\_64 x86\_64 GNU/Linux

Uptime Command - Lists how long the system is active:

19:42:54 up 44 days, 22:22, 3 users, load average: 0.00, 0.00, 0.00

Whoami Command - Shows what the current username is:

sarahmcnelis

Who Command - List of users logged in:

katemcgrath pts/0 2021-04-28 16:27 (91.142.105.64)

sarahmcnelis pts/1 2021-04-28 19:31 (212.129.80.46)

pauljackbyrne pts/2 2021-04-28 18:39 (86.41.2.235)

Finger Command - Detailed list of users currently logged in:

Login Name Tty Idle Login Time Office Office Phone

katemcgrath pts/0 Apr 28 16:27 (91.142.105.64)

pauljackbyrne pts/2 14 Apr 28 18:39 (86.41.2.235)

sarahmcnelis pts/1 Apr 28 19:31 (212.129.80.46)

W Command - Users currently logged in and what they are doing:

19:42:54 up 44 days, 22:22, 3 users, load average: 0.00, 0.00, 0.00

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

katemcgr pts/0 91.142.105.64 16:27 4.00s 0.61s 0.48s vim testInput.sh

sarahmcn pts/1 212.129.80.46 19:31 6.00s 0.03s 0.00s w

pauljack pts/2 86.41.2.235 18:39 14:22 0.05s 0.05s -bash

Top Command - Summary of System Information:

top - 19:42:58 up 44 days, 22:22, 3 users, load average: 0.00, 0.00, 0.00

Tasks: 112 total, 1 running, 111 sleeping, 0 stopped, 0 zombie

%Cpu(s): 1.0 us, 0.0 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st

MiB Mem : 978.6 total, 103.0 free, 211.3 used, 664.4 buff/cache

MiB Swap: 0.0 total, 0.0 free, 0.0 used. 597.0 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

2035429 katemcg+ 20 0 24100 10348 6376 S 0.3 1.0 0:00.49 vim

1 root 20 0 168932 10952 6316 S 0.0 1.1 4:50.59 systemd

2 root 20 0 0 0 0 S 0.0 0.0 0:00.13 kthreadd

3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_gp

4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu\_par\_gp

6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd

9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm\_percpu\_wq

10 root 20 0 0 0 0 S 0.0 0.0 1:22.40 ksoftirqd/0

11 root 20 0 0 0 0 I 0.0 0.0 1:38.67 rcu\_sched

12 root rt 0 0 0 0 S 0.0 0.0 0:20.58 migration/0

13 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0

14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kdevtmpfs

15 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 netns

16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu\_tasks\_kthre

17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kauditd

18 root 20 0 0 0 0 S 0.0 0.0 0:00.00 xenbus

19 root 20 0 0 0 0 S 0.0 0.0 0:00.01 xenwatch

20 root 20 0 0 0 0 S 0.0 0.0 0:01.22 khungtaskd

21 root 20 0 0 0 0 S 0.0 0.0 0:00.00 oom\_reaper

22 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 writeback

23 root 20 0 0 0 0 S 0.0 0.0 0:00.09 kcompactd0

24 root 25 5 0 0 0 S 0.0 0.0 0:00.00 ksmd

25 root 39 19 0 0 0 S 0.0 0.0 0:08.38 khugepaged

71 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kintegrityd

72 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kblockd

73 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 blkcg\_punt\_bio

74 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 tpm\_dev\_wq

75 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 ata\_sff

76 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 md

77 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 edac-poller

78 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 devfreq\_wq

79 root rt 0 0 0 0 S 0.0 0.0 0:00.00 watchdogd

82 root 20 0 0 0 0 S 0.0 0.0 0:26.21 kswapd0

83 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ecryptfs-kthrea

85 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kthrotld

86 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 nvme-wq

87 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 nvme-reset-wq

History Command - Report of command history

1 mkdir Music

2 ls

3 cd Music

4 mkdir Westlife

5 mkdir Adele

6 mkdir Michael Buble

7 ls

8 cd Adele

9 ls

10 mkdir 19

11 mkdir 21

12 mkdir 25

13 ls

14 cd ..

15 ls

16 cd Michael Buble

17 cd Michael

18 cd ..

19 rmdir Michael

20 ls

21 rmdir Buble

22 ls

23 cd Westlife

24 mkdir CoastToCoast

25 mkdir WorldOfOurOwn

26 mkdir TheLoveAlbum

27 ls

28 cd ..

29 ls

30 mkdir MichaelBuble

31 cd MichaelBuble/

32 mkdir CrazyLove

33 mkdir Christmas

34 mkdir Love

35 ls

36 cd..

37 cd ..

38 ls

39 cd ..

40 tree

41 cd Music

42 cd Adele

43 rm 19

44 ls

45 cd 19

46 cd ..

47 rmdir 19

48 ls

49 cd ..

50 cd MichaelBuble

51 rmdir CrazyLove

52 ls

53 cd ..

54 cd Westlife

55 rmdir TheLoveAlbum

56 ls

57 cd ..

58 cd ..#

59 ls

60 cd ..

61 tree

62 ls

63 echo hello world

64 passwd

65 date

66 hostname

67 arch

68 uname -a

69 dmesg | more

70 uptime

71 whoami

72 who

73 last

74 finger

75 w

76 top

77 echo $SHELL

78 echo {con, pre} {sent fer} {s, ed}

79 echo {con, pre}{sent fer}{s, ed}

80 echo {con,pre}{sent,fer}{s,ed}

81 man ls

82 man who

83 clear

84 man who

85 clear

86 cal 2000

87 cal

88 cal 9 1752

89 cal sept 1752

90 cal -d 1752-9

91 yes please

92 y please

93 time sleep 5

94 sleep 5

95 history

96 ls -l

97 rm ip\_location

98 ls

99 clear

100 linux www.google.com

101 lynx www.google.com

102 clear

103 lynx www.google.com

104 ls

105 vi filename

106 ls

107 mv filename shellscript

108 ls

109 vi shellscript

110 ls

111 shellscript

112 vi shellscript

113 mv shellscript projectscript.sh

114 ls

115 projectscript.sh

116 ls

117 vi projectscript.sh

118 ls

119 bash projectscript.sh

120 vi projectscript.sh

121 ls

122 echo "\n\n" hello

123 echo \n hello \n

124 echo $"\n\n" hello $"\n"

125 echo " "

126 vi projectscript.sh

127 clear

128 bash projectscript.sh

129 vi projectscript.sh

130 clear

131 bash projectscript.sh

132 vi projectscript.sh

133 ls

134 vi projectscript.sh

135 clear

136 bash projectscript.sh

137 echo " "

138 echo Date Command - Displays and sets the systems date/time:

139 date

140 echo " "

141 echo " "

142 echo Hostname Command - Displays the hostname of this VM:

143 hostname

144 echo " "

145 echo " "

146 echo Arch Command - Lists the computer’s architecture:

147 arch

148 echo " "

149 echo " "

150 echo Uname -a Command - Basic information on current linux system

151 uname -a

152 echo " "

153 echo " "

154 echo Uptime Command - Lists how long the system is active:

155 uptime

156 echo " "

157 echo " "

158 echo Whoami Command - Shows what the current username is:

159 whoami

160 echo " "

161 echo " "

162 echo Who Command - List of users logged in:

163 who

164 echo " "

165 echo " "

166 echo Finger Command - Detailed list of users currently logged in:

167 finger

168 echo " "

169 echo " "

170 echo W Command - Users currently logged in and what they are doing:

171 w

172 echo " "

173 echo " "

174 echo Top Command - Summary of System Information:

175 top

176 echo " "

177 echo " "

178 echo History Command - Report of command history:

179 history

180 echo " "

181 chmod u+x projectscript.sh

182 echo " "

183 echo Date Command - Displays and sets the systems date/time:

184 date

185 echo " "

186 echo " "

187 echo Hostname Command - Displays the hostname of this VM:

188 hostname

189 echo " "

190 echo " "

191 echo Arch Command - Lists the computer’s architecture:

192 arch

193 echo " "

194 echo " "

195 echo Uname -a Command - Basic information on current linux system

196 uname -a

197 echo " "

198 echo " "

199 echo Uptime Command - Lists how long the system is active:

200 uptime

201 echo " "

202 echo " "

203 echo Whoami Command - Shows what the current username is:

204 whoami

205 echo " "

206 echo " "

207 echo Who Command - List of users logged in:

208 who

209 echo " "

210 echo " "

211 echo Finger Command - Detailed list of users currently logged in:

212 finger

213 echo " "

214 echo " "

215 echo W Command - Users currently logged in and what they are doing:

216 w

217 echo " "

218 echo " "

219 echo Top Command - Summary of System Information:

220 top

221 echo " "

222 echo " "

223 echo History Command - Report of command history:

224 history

225 echo " "

226 chmod u+x projectscript.sh

227 echo History Command - Report of command history

228 history

229 chmod u+x projectscript.sh

230 echo History Command - Report of command history

231 history

232 chmod u+x projectscript.sh

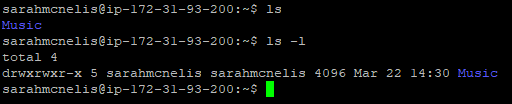
233 echo History Command - Report of command history

234 history

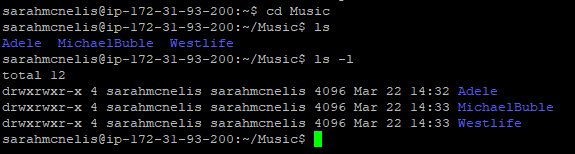
### Q2.3

#### Q2.3.1

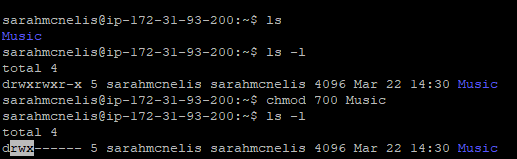
In order to answer this question, there are a number of details which we must be aware of. Firstly, the ls command will populate a list of all directories as explained by Saive, R (2018) *15 Basic ‘ls’ Command Examples in Linux.* Techmint. Available at: <https://www.tecmint.com/15-basic-ls-command-examples-in-linux/> [Accessed April 20, 2021]. Saive (2018) also states that the ls -l command shows the file or directory name, size, owner, modified date and time and also it is permissions.



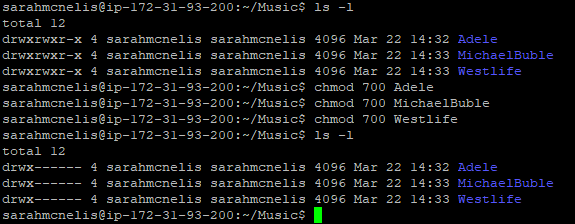
As you can see from this screenshot using the ls command I can see a list of all my directories. I currently only have one called Music. Once I entered ls -l I was able to see a detailed list of my directories. From this screenshot we can see the permissions of this directory; drwxrwxr-x. McKay, D. (2021) *How to Use the chmod Command on Linux.* Howtogeek. Available at: <https://www.howtogeek.com/437958/how-to-use-the-chmod-command-on-linux/#:~:text=How%20to%20Use%20the%20chmod%20Command%20on%20Linux,6%20Numerical%20Shorthand.%20...%207%20Advanced%20Options.%20> [Accessed April 20, 2021] explains that the d stands for directory. If this were a file there would be a dash symbol instead of the d. McKay (2021) continues by disclosing the meaning behind the other letters; r= read permissions, w= write permissions and x= execute permissions. He elaborates further telling us that a dash symbol in the permission section would mean there are no permissions at all. He also explains that the first set are permissions of the owner or u, the second set are permissions for the group or g and the third set are the permissions for other users or o (McKay, 2021). He also states that the letter ‘a’ can be used for all three permissions.

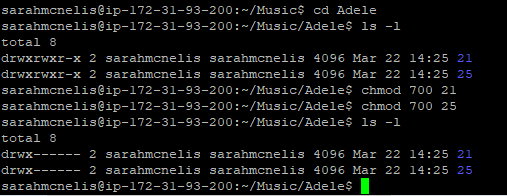


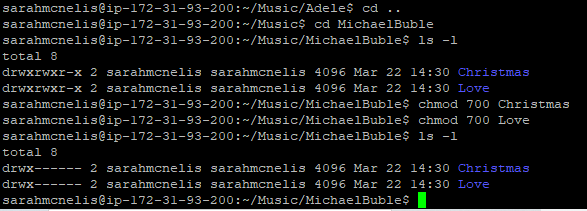
As you can see from this screenshot I have moved into the Music directory using the cd command and have used both the ls and ls -l commands to list the contents of these directories. We can see again that there are three more directories. If I were to move into each of these I would also be able to list the contents of those directories. In any case I will first change the permission of my directory Music so that only the owner has read, write and execute privileges. McKay (2021) states that this can be done using the chmod command followed by the letters and characters that we want to set permissions for. However, the *Linux File Permission Tutorial: How to Check and Change Permissions* (2019). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-file-permissions> [Accessed April 20, 2021] describes another way of changing the permissions. It explains the use of the chmod command followed by three digits and the filename. These three digits refer to the level of permissions granted at each level. The first digit represents the owner, the second is for the group and the third is for other users. Phoenixnap (2019) explains what each number represents; 0= no access, 1= execute, 2= write, 3= execute and write, 4= read, 5= execute and read, 6= read and write and 7=execute, read and write. As per the below screenshot I have entered chmod 700 Music. The number seven gives me/the owner full access, and the two zeros gives both the group and other users no access to this file. We can see this represented when I entered ls -l into the command line which shows the list of the permissions for the directory. We can see it shows drwx------. As you can see from the image below I have highlighted where my access is still in read, write and execute mode and the rest of the permissions are represented with the dash symbol showing they have been removed.

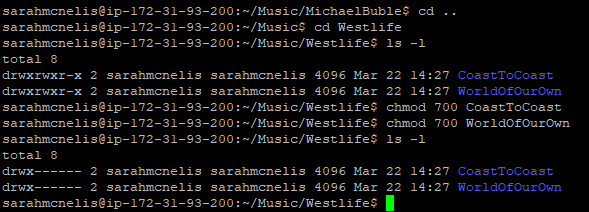


An important note here is that once I moved into the Music directory and entered the command ls -l, this showed me that the action I just performed back in the main directory did not change the access settings of the files within the Music directory. Therefore, I have enclosed screenshots below showing before and after the permissions were changed for each.



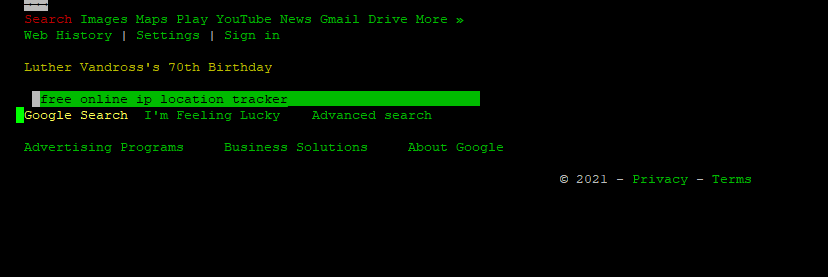




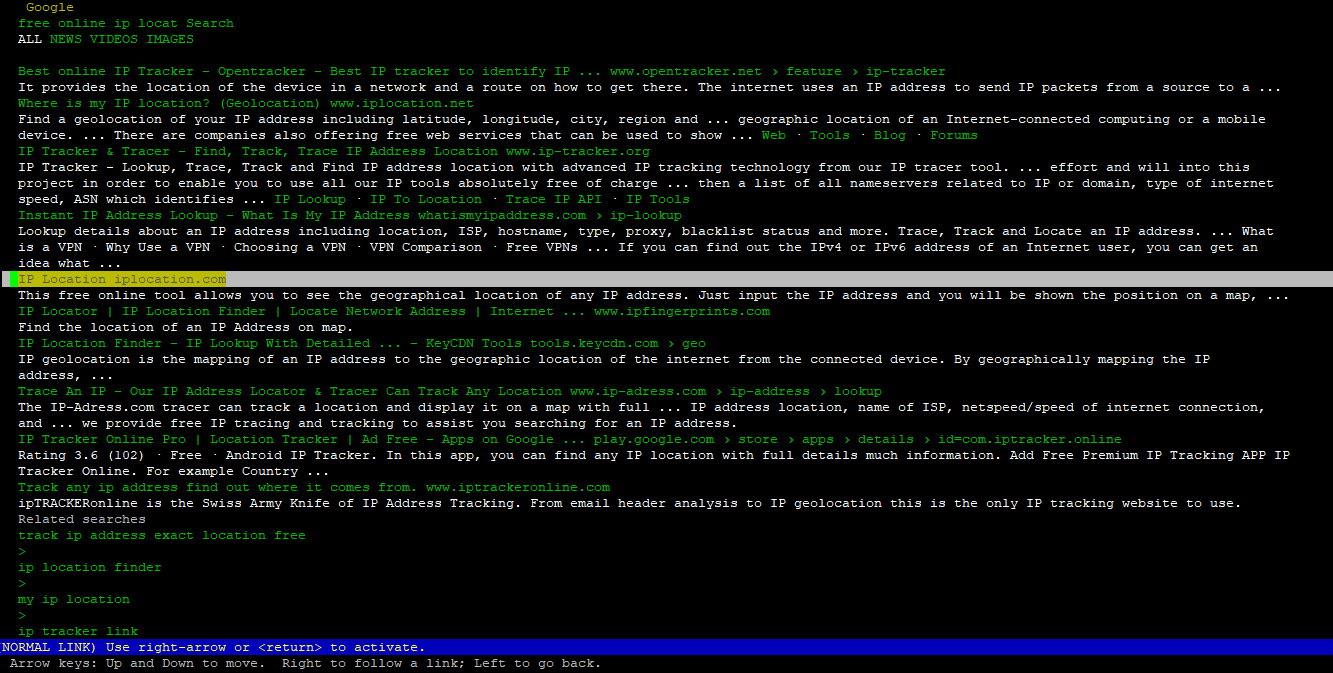


#### Q2.3.2:

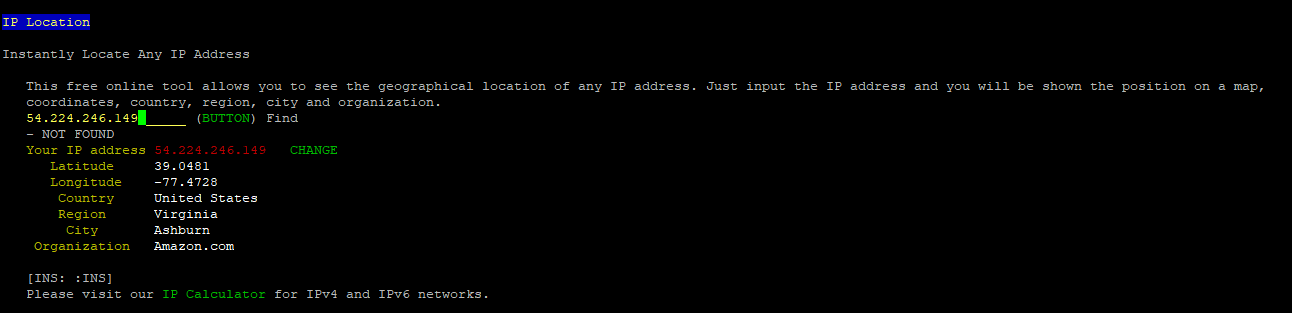
For this exercise I entered: lynx [www.google.com](http://www.google.com) into the command line of the terminal. This brought me to google in the linux terminal as seen in the below image. I entered my search into google as can be seen in highlighted area in the screenshot below.



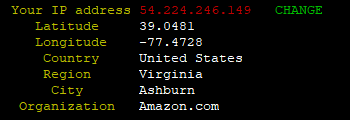
Next, a page with results populated. After looking through each webpage on the list I discovered that iplocation.com ( the one which I have highlighted in the below image) was the best site. I found it was simple, clear and easy to use.



After clicking into this iplocation webpage it auto populated my IP address into the search bar.

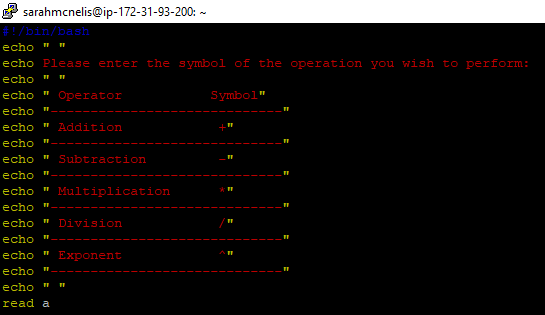


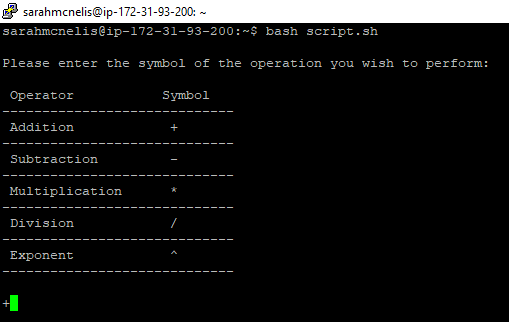
As you can see from the image below, the results included the latitude, longitude, country, region, city and organization related to this IP address. Therefore, I have determined that the VM in which I am using is located in Ashburn, Virginia in USA.



### Q2.4:

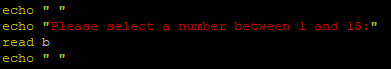
For this task I have used my research from task 2.2 in regard to shell scripting. Following the same procedure, I created a shell script which first asks the user to input the symbol of the arithmetic operation they wish to perform. Similarly, to task 2.2 I have made use of the echo command in order for the script to print out what I want to ask the user and for spacing to make it easy to read. As explained in *Simple Calculator in Bash,* (2018). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/simple-calculator-bash/> [Accessed April 26, 2021] the echo command is used to print a line of text in standard output. This can be seen in the images below. The first image is the shell script. The second is what the output looks like in the linux terminal.

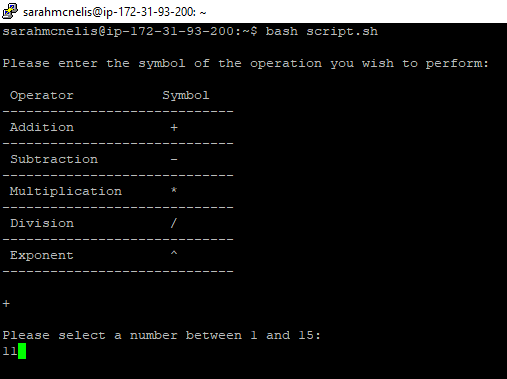




As you can also see in the above script image I have made use of the read command. Yesmin, F. (2018), *30 Bash Script Examples.* Linuxhint. Available at: <https://linuxhint.com/30_bash_script_examples/#t5> [Accessed 27 April, 2021] explains that the read command takes in the user’s input in bash. For this script I have labelled the first input as ‘a’.

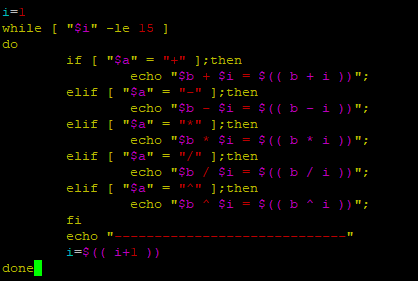
The next step of this shell script is to prompt the user to enter a number between 1 and 15. I have labelled this input for the user, ‘b’. I have also made use of the echo command to allow for blank spaces which as you can see below makes them much easier to read in the terminal. The first screenshot is the shell script. The second is what the output looks like in the linux terminal.





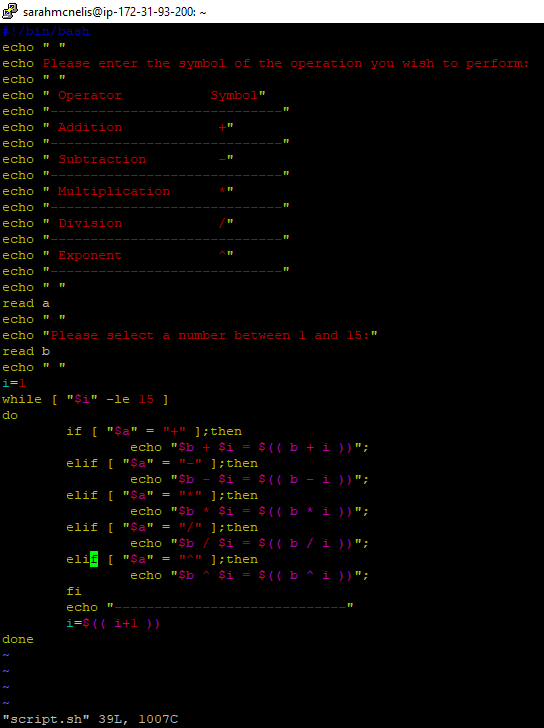
The next step involved a lot of research to complete this task. I needed to create a loop which iterates the user’s input ‘b’ from 1 to 15. In other words, I needed to calculate the operation first selected on the input number and iterate from 1 to 15. I decided I needed to use a while loop to begin with. Yesmin (2018) explains that a while loop will increment by 1 in each step until it reaches the selected variable. In my script I have first set ‘i’ equal to 1. In other words, the calculation will start with 1. Next I have stated while ‘i‘ is less than 15, which is the range which I want to calculate to from 1, do the following statements. A while block will only begin with the command ‘do’ and end with the command ‘done’ (Yesmin, 2018).

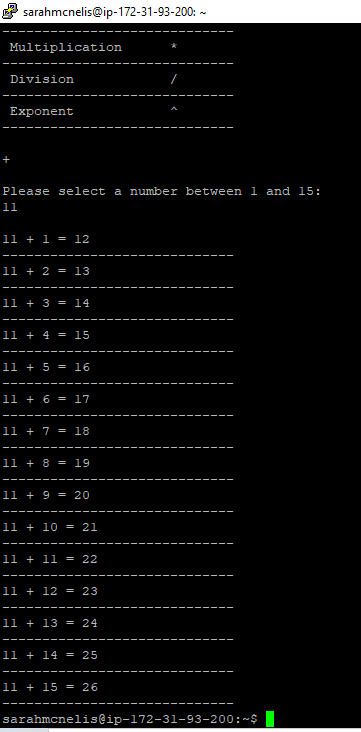
At this point I have created the while loop for iteration; however, I now have to consider the maths table which contains 5 different operators. This is when the if statements come into effect. Yesmin (2018) states that an if statement should be used for single or multiple conditions. In other words, when an if statement is true it will run that line of code. When that if statement is false then it will move onto the next if statement and go through the same process.



Therefore, in terms of this task, I have set the first if statement for the first operator, addition. To break it down I am saying if ‘a’ is equal to addition then print the following statement. I have done this using the echo command again which says b plus i is equal to the calculation b plus i. I have used $((…)) notation which is known as Arithmetic Expansion according to *Math Arithmetic: How to do Calculation in Bash.* Shell-tips. Available at: <https://www.shell-tips.com/bash/math-arithmetic-calculation/> [Accessed April 25, 2021]. In other words , in order to perform calculations, you need to use the dollar symbol with either brackets or square brackets. This will perform the actual calculation in my script and produce the answer in the terminal. The same structure is applied to each if/elif statement, the only change is the operator.

Yesmin (2018) also explains that the block starts with ‘if’ and ends with ‘fi’. Therefore, the code will run down through the if/elif statements and will terminate at fi. Remembering that the if block is within the while loop, I have now set i equal to i plus 1. This will run the selected if statement in a loop from 1 to 15. Finally, the last step of this script is ending the while block with the done command. The first image below is the shell script and the second is the output in the linux terminal. As you can see I have successfully created a shell script which takes two inputs of the user, performs calculations using a while loop and if statements.





#### **Actual shell script for Q2.4:**

#!/bin/bash

echo " "

echo Please enter the symbol of the operation you wish to perform:

echo " "

echo " Operator Symbol"

echo "-----------------------------"

echo " Addition +"

echo "-----------------------------"

echo " Subtraction -"

echo "-----------------------------"

echo " Multiplication \*"

echo "-----------------------------"

echo " Division /"

echo "-----------------------------"

echo " Exponent ^"

echo "-----------------------------"

echo " "

read a

echo " "

echo "Please select a number between 1 and 15:"

read b

echo " "

i=1

while [ "$i" -le 15 ]

do

if [ "$a" = "+" ];then

echo "$b + $i = $(( b + i ))";

elif [ "$a" = "-" ];then

echo "$b - $i = $(( b - i ))";

elif [ "$a" = "\*" ];then

echo "$b \* $i = $(( b \* i ))";

elif [ "$a" = "/" ];then

echo "$b / $i = $(( b / i ))";

elif [ "$a" = "^" ];then

echo "$b ^ $i = $(( b ^ i ))";

fi

echo "-----------------------------"

i=$(( i+1 ))

done

## **References**

* *Arch Command in Linux with Examples* (2019). Geekforgeeks. Available at: <https://www.geeksforgeeks.org/arch-command-in-linux-with-examples/#:~:text=arch%20command%20is%20used%20to%20print%20the%20computer,to%20Install%20Tilix%20Terminal%20Emulator%20in%20Ubuntu%2FFedora%2FArch%20Linux%3F> [Accessed April 19, 2021]
* Beck, K. (2020) *How to Convert Negative Numbers to Binary.*  Sciencing. Available at: <https://sciencing.com/convert-negative-numbers-binary-5124016.html> Accessed [April 12, 2021]
* *Clear Command in Linux with Examples* (2020). Geekforgeeks. Available at: <https://www.geeksforgeeks.org/clear-command-in-linux-with-examples/> [Accessed April 19, 2021]
* *Date Command in Linux: How to Set, Change, Format and Display Date* (2020). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-date-command> [Accessed April 19, 2021]
* *Difference between “echo $SHELL” and “which bash”* (2017). Unix.stackexchange. Available at: <https://unix.stackexchange.com/questions/43499/difference-between-echo-shell-and-which-bash#:~:text=echo%20%24SHELL%20shows%20the%20value%20of%20the%20SHELL,alone%20unless%20it%20is%20unset%20when%20they%20start> [Accessed April 19, 2021]
* *Echo Command in Linux with Examples* (2019). Linuxize. Available at: <https://linuxize.com/post/echo-command-in-linux-with-examples/> [Accessed April 19, 2021]
* Emmet (2020) *Using the Hostname Command.* Pimylifeup. Available at: <https://pimylifeup.com/hostname-command/#:~:text=The%20hostname%20command%20can%20be%20used%20to%20retrieve,and%20connect%20to%20machines%20within%20a%20network%20quickly> [Accessed April 19, 2021]
* *Finger Command in Linux with Examples* (2020). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/finger-command-in-linux-with-examples/> [Accessed April 19, 2021]
* Gite, V. (2021) *How to Run the .sh File Shell Script in Linux.UNIX.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/run-execute-sh-shell-script/#:~:text=%20The%20procedure%20to%20run%20the%20.sh%20file,5%20To%20run%20your%20script%20%3A%0A.%2Fscript-name-here.sh%0AAnother...%20More%20> [Accessed April 26, 2021]
* Gite, V. (2020), *Linux and Unix Command to View File.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/unix-linux-command-to-view-file/#:~:text=Linux%20And%20Unix%20Command%20To%20View%20File%201,as%20if%20you%20had%20double-clicked%20the%20file’s%20icon> [Accessed April 26, 2021]
* Gite, V. (2009), *Save Bash Shell Script Output to a File.* Cyberciti. Available at: <https://www.cyberciti.biz/faq/bash-script-output/> [Accessed April 26, 2021]
* *History command inside bash scripts* (2001). Stackexchange. Available at: <https://unix.stackexchange.com/questions/5684/history-command-inside-bash-script> [Accessed April 26, 2021]
* *How to Add Binary Numbers* (2021). Wikihow. Available at: <https://www.wikihow.com/Add-Binary-Numbers#:~:text=%20Adding%20Binary%20Numbers%20Using%20Place%20Value%20,or%203%20%28if%20you%20carried%20from...%20More%20> [Accessed April 12, 2021]
* *How to automate some linux commands* (2013). Stackoverflow. Available at: <https://stackoverflow.com/questions/14882292/how-to-automate-some-linux-commands> [Accessed April 26, 2021]
* *How to Check and Change Permissions* (2019). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-file-permissions> [Accessed April 20, 2021]
* *How to Convert Decimal Number into Binary* (2020). Electricalbaba. Available at: <https://electricalbaba.com/how-to-convert-decimal-number-into-binary/#:~:text=An%20easy%20method%20of%20converting%20a%20decimal%20number,remainder%20after%20each%20division%20in%20the%20reverse%20order> [Accessed April 12, 2021]
* *How to use the Linux Sleep Command with Examples* (2021). Phoenixnap. Available at: <https://phoenixnap.com/kb/linux-sleep> [Accessed April 19, 2021].
* Kiarie, J. (2021) *How to Create Simple Shell Scripts in Linux.* Tecmint. Available at: <https://www.tecmint.com/create-shell-scripts-in-linux/> [Accessed April 26, 2021]
* Kumar, V. (2020) *How to use man command in Linux, important options for beginner.* Cyberpratibha. Available at: <https://www.cyberpratibha.com/how-to-use-man-command-in-linux/#:~:text=How%20to%20use%20man%20command%20in%20Linux%2C%20important,option%20of%20man%20command.%20...%209%20Conclusion.%20> [Accessed April 19, 2021]
* *Last Command in Linux with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/last-command-in-linux-with-examples/#:~:text=The%20last%20command%20in%20Linux%20is%20used%20to,login%20in%20%28and%20out%29%20time%20and%20their%20host-name> [Accessed April 19, 2021]
* *Linux sleep command* (2019). Computerhope. Available at: <https://www.computerhope.com/unix/usleep.htm> [Accessed April 19, 2021]
* *Linux Uptime Command with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/linux-uptime-command-with-examples/> [Accessed April 19, 2021]
* *Linux who command* (2019). Computer hope. Available at: <https://www.computerhope.com/unix/uwho.htm> [Accessed April 19, 2021]
* *Logic Gate Simulator.* Academo. Available at: <https://academo.org/demos/logic-gate-simulator/> [Accessed on April 12, 2021]
* *Logic Gates,* Tutorialspoint. Available at: <https://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm> [Accessed April 12, 2021].
* *Man Command in Linux with Examples* (2021). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/man-command-in-linux-with-examples/#:~:text=man%20command%20in%20Linux%20is%20used%20to%20display,Every%20manual%20is%20divided%20into%20the%20following%20sections%3A> [Accessed April 19, 2021].
* *Math Arithmetic: How to do Calculation in Bash?* Shell-tips. Available at: <https://www.shell-tips.com/bash/math-arithmetic-calculation/> [Accessed 25 April, 2021]
* McKay, D. (2021) *How to Use the chmod Command on Linux.* Howtogeek. Available at: <https://www.howtogeek.com/437958/how-to-use-the-chmod-command-on-linux/#:~:text=How%20to%20Use%20the%20chmod%20Command%20on%20Linux,6%20Numerical%20Shorthand.%20...%207%20Advanced%20Options.%20> [Accessed April 20, 2021]
* McKay, D. (2020) *How to Use the History Command on Linux.* Howtogeek. Available at: <https://www.howtogeek.com/465243/how-to-use-the-history-command-on-linux/#:~:text=%20How%20to%20Use%20the%20history%20Command%20on,the%20history%20list%2C%20type%20an%20exclamation...%20More%20> [Accessed April 19, 2021].
* McKay, D. (2019) *How to Use the Yes Command on Linux*. Howtogeek. Available at: <https://www.howtogeek.com/415535/how-to-use-the-yes-command-on-linux/> [Accessed April 19, 2021]
* Mishra, S. (2018) (*#!/Bin/bash) What exactly is this?*. Medium. Available at: <https://medium.com/@codingmaths/bin-bash-what-exactly-is-this-95fc8db817bf#:~:text=%2Fbin%2Fbash%20is%20the%20most%20common%20shell%20used%20as,developed%20and%20better%20syntax.%20WHAT%20IS%20IT%20%3F> [Accessed April 19, 2021]
* Murray, C. (2020) *How to Handle Account Passwords in Linux with Passwd Command*. Linuxhandbook. Available at: <https://linuxhandbook.com/passwd-command/> [Accessed April 19, 2021]
* *Numbers and binary addition.* BBC. Available at: <https://www.bbc.co.uk/bitesize/guides/zjfgjxs/revision/5#:~:text=The%20smallest%20positive%20number%20is%20the%20smallest%20binary,use%20two%27s%20complement%20to%20create%20negative%20binary%20numbers> [Accessed April 12, 2021]
* *Overflow.* BBC. Available at: <https://www.bbc.co.uk/bitesize/guides/zjfgjxs/revision/3> [Accessed April 12, 2021].
* Prakash, A. (2019), *5 Examples of Cal Command in Linux.* Linuxhandbook. Available at: <https://linuxhandbook.com/cal-command/> [Accessed April 19, 2021]
* *RapidTables.* Available at: <https://www.rapidtables.com/convert/number/how-decimal-to-binary.html> [Accessed April 12, 2021
* Saive, R (2018) *15 Basic ‘ls’ Command Examples in Linux.* Techmint. Available at: <https://www.tecmint.com/15-basic-ls-command-examples-in-linux/> [Accessed April 20, 2021].
* *Simple calculator in Bash* (2018). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/simple-calculator-bash/> [Accessed 26 April, 2021]
* Shrestha, N. (2015) *7 ‘dmesg’ Commands for Troubleshooting and Collecting Information of Linus Systems.* Tecmint. Available at: <https://www.tecmint.com/dmesg-commands/> [Accessed April 19, 2021]
* Teja, R. (2021) *Binary Adder and Subtractor*. Electronicshub. Available at: <https://www.electronicshub.org/binary-adder-and-subtractor/> [Accessed April 19, 2021]
* *The Missing Eleven Days!* (2013)Projectbritain. Available at: <https://www.projectbritain.com/calendar/September/calendar.html#:~:text=In%201752%20Britain%20decided%20to%20abandon%20the%20Julian,Many%20people%20believed%20their%20lives%20would%20be%20shortened> [Accessed April 19, 2021]
* *Top command in Linux with Examples* (2019). Geeksforgeeks. Available at: <https://www.geeksforgeeks.org/top-command-in-linux-with-examples/> [Accessed April 19,2021]
* *Uname Command – Print Unix System Information.* Unixtutorial. Available at: <https://www.unixtutorial.org/commands/uname#:~:text=uname%20command%20–%20print%20Unix%20system%20information%201,uname%20-v.%20...%203%20Common%20uname%20usage.%20> [Accessed April 19, 2021]
* *Unix/linux – The vi Editor Tutorial*. Tutorialspoint. Available at: <https://www.tutorialspoint.com/unix/unix-vi-editor.htm> [Accessed April 26, 2021]
* *W Command in Linux* (2019). Linuxize. Available at: <https://linuxize.com/post/w-command-in-linux/> [Accessed April 19, 2021]
* *Whoami Command in Linux* (2020). Linuxize. Available at: <https://linuxize.com/post/whoami-command-in-linux/> [Accessed April 19, 2021
* Yesmin, F. (2018), *30 Bash Script Examples.* Linuxhint. Available at: <https://linuxhint.com/30_bash_script_examples/#t5> [Accessed 27 April, 2021
* Zonker (2009), *Vim 101: A Beginner’s Guide to Vim*. Linux. Available at: <https://www.linux.com/training-tutorials/vim-101-beginners-guide-vim/> [Accessed April 26, 2021]