

FINAL

by Corey Harrison

Submission date: 26-Jul-2023 01:56PM (UTC+0100)

Submission ID: 210704651

File name: 44013_Corey_Harrison_FINAL_503754_378324405.docx (4.57M)

Word count: 2202

Character count: 12830

COMPUTER ARCHITECTURE AND OPERATING SYSTEMS.

Corey Harrison

2231450

1

TABLE OF CONTENTS.

- Introduction
- Operating Systems and their key functions
- Use of CLI commands in a Windows
- Use of CLI commands in a Linux OS
- Use of Batch scripts in a Windows
- Use of Bash scripts in a Linux OS
- Conclusion

Introduction.

1

An operating system (OS) is a fundamental software that manages computer hardware and software resources, acting as an intermediary between users and the underlying hardware. In Windows, the Command-Line Interface (CLI) is accessible through Command Prompt, allowing users to interact with the OS using text-based commands. Linux embraces the Command-Line Interface (CLI) as an integral part of its design. The Linux terminal provides a powerful environment for executing CLI commands, enabling users to manage files, control processes, configure networks, and perform system administration tasks. Batch scripts in Windows, often with .bat or .cmd extensions, are sequences of text-based commands that automate tasks. These scripts allow users to create reusable and automated solutions for tasks like file management, program execution, system configuration, and repetitive operations. Bash scripts in Linux utilize the Bash shell (Bourne Again SHell) and provide powerful scripting capabilities. These text-based scripts automate tasks, manage system configurations, and execute programs. Bash scripts are essential for Linux system administration, software installations, and automation processes.

Operating Systems and their key functions.

In the modern world of computers and technology, operating systems operating system serves as the foundation of modern computing and play a huge fundamental role in enabling the efficient interaction between hardware, software, and users. An operating system (OS) is a crucial software

component that serves as the bridge between the hardware of a computer system and the various applications and programs running on it. Different types of operating systems exist, each designed to fulfil specific requirements and cater to different computing devices and scenarios. Common examples include Microsoft Windows, macOS, Linux, iOS, and Android, which power desktop and laptop computers, servers, smartphones, tablets, and other embedded systems. Windows OS was first created it was called DOS 1.0 (Stallings, W.), which was released in August 1981 and had several updates like DOS 2.0 in 1983(Stallings, W.) and so on up till the point of technology where we are now which is much more advanced with all the other major updates (Stallings, W.). Not too further in the future Mac Operating systems was created, Mac operating system was developed by the apple in 1984 (Mudiraj, A.R) which also continued to have several updates as technology advanced. Linux operating system was first created in 1991 (Rajagopal, R). FEOGLI QODIROV, ZAQ MANSUROVA state that Android operating system was first introduced in 2007 and continued to update and improve the Android OS from there on (QODIROV).

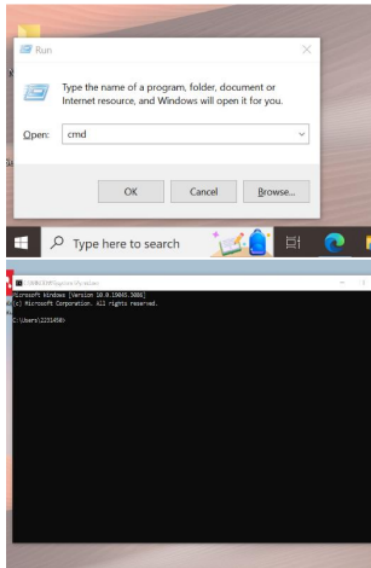
At its core, an operating system acts as a facilitator, providing an interface through which users can interact with the computer's hardware and software resources. It manages and coordinates the system's hardware components, such as the central processing unit (CPU), memory, storage devices, input/output devices, and network connections, allowing them to work together harmoniously (Brett, P). One of the primary benefits of the operating system managing the CPU is efficient process scheduling. It allocates CPU time to various running programs, ensuring fair distribution and maximizing overall system performance. By managing memory, the operating system optimizes RAM usage, allowing multiple applications to coexist without running out of memory, which enhances multitasking capabilities. Furthermore, the operating system's management of storage devices ensures seamless and organized data storage and retrieval. It handles file management, creating, accessing, and deleting files and directories, making data management hassle-free for users. It also implements features like caching and virtual memory to improve performance and support larger data sets. In the realm of input/output devices, the operating system acts as a mediator, enabling smooth communication between software applications and hardware devices. This allows users to interact with printers, scanners, keyboards, and other peripherals effortlessly (Fischer, T). Moreover, the OS manages device drivers, ensuring that compatible drivers are available for different hardware components, facilitating plug-and-play functionality.

The primary goal of an operating system is to provide a user-friendly and efficient environment for executing programs and utilizing system resources effectively. It acts as a mediator, allocating and managing the computer's resources, such as memory and processing power, among the different software applications and processes running concurrently. An operating system offers a range of services and functionalities that enhance the overall user experience. It provides a platform for installing and executing applications, organizing files and folders, controlling user access and permissions, managing network connections, handling security measures, and facilitating communication between different software components (Dinneen, J.D). All these key features and more improves the speed of use of applications and so on, make it easier and more accessible for the user and therefore causes the sales on the products from Android OS, Mac OS, Windows OS, iOS, ETC to skyrocket and over time the operating systems have improved and been updated repeatedly to meet the desired and unique requirements of their respective domains.

An operating system is the essential software layer that enables users to interact with computers and devices while efficiently managing and coordinating hardware resources. It plays a vital role in ensuring smooth and reliable operation, improving user productivity, and providing a foundation for running a wide range of software applications. Without operating systems, the digital world as we know it would not be possible (Waldspurger, C.A).

Use of CLI commands in a Windows

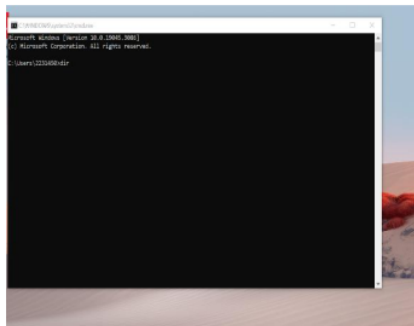
Example 1.



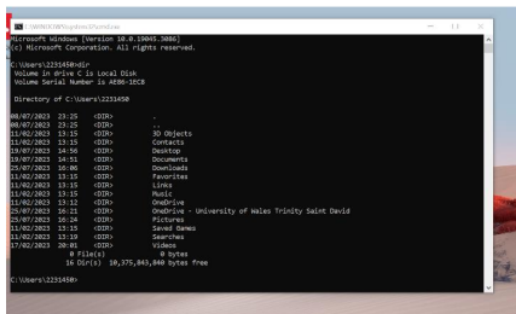
Firstly, you want to use a shortcut for example 'Win+r' which will bring you here.

Once here type in 'cmd' and click ok or enter.

Once done if done correctly it will bring you here.

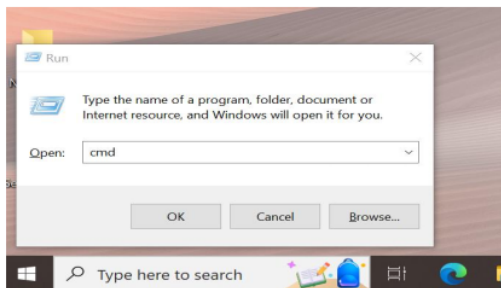


Once here then type in 'dir'.

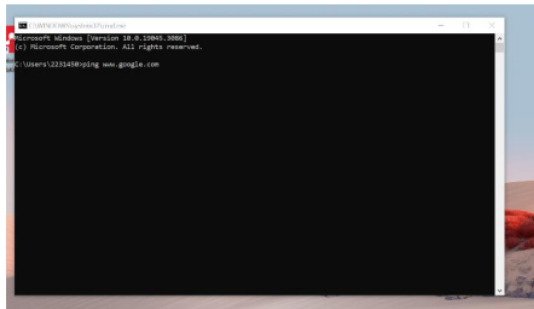


Once you have clicked enter the output will display a list of files and directories in the current directory, along with their sizes and with timestamps.

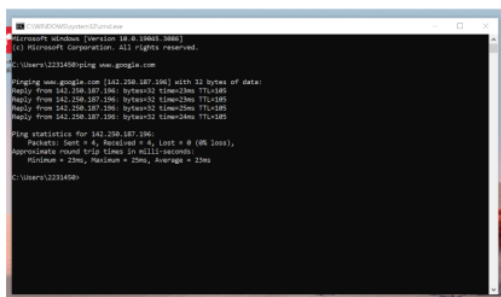
Example 2.



Again, you can start with 'Win+r', once loaded type 'cmd' and click enter or press ok.



Once loaded type in 'ping' following a space and followed by the IP address or hostname of the device you want to test the connection to. The example I used was 'ping www.google.com'.



Once you have clicked enter the command will send a series of ICMP (Internet Control Message Protocol) echo requests to the specified device and wait for responses. It will display the results, including the time it takes for each echo request to reach the device and get a response. The results will also show the success rate of the pings.

Command-Line Interface (CLI) commands in Windows provide a powerful way to interact with the operating system and execute various tasks without the need for a graphical user interface.

Command Prompt (cmd.exe): The traditional Command Prompt, also known as CMD, is a basic and uncomplicated CLI tool. To conduct activities like exploring the file system, creating, or deleting directories, copying files, and launching executable programmes, users can put commands in the command line and then execute them. The automatic command interpretation capability of Microsoft Windows CMD is available (Yamin, M.M).

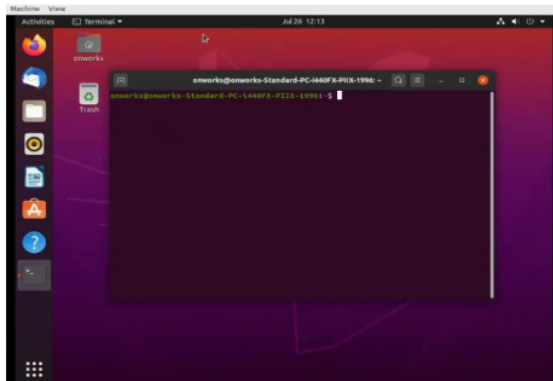
PowerShell is a more sophisticated CLI that provides a broad range of commands and scripting features. It is adaptable for system administration duties thanks to its use of a scripting language and access to the .NET framework. A variety of features are available using PowerShell cmdlets, such as managing services, registry settings, Active Directory, and more (Holmes, L).

Windows CLI commands provide an efficient and flexible way to interact with the operating system, automate tasks, troubleshoot issues, and manage system resources effectively. Whether it's basic file

operations or complex system administration, mastering these commands can significantly enhance productivity in the Windows environment (Mueller, J.P).

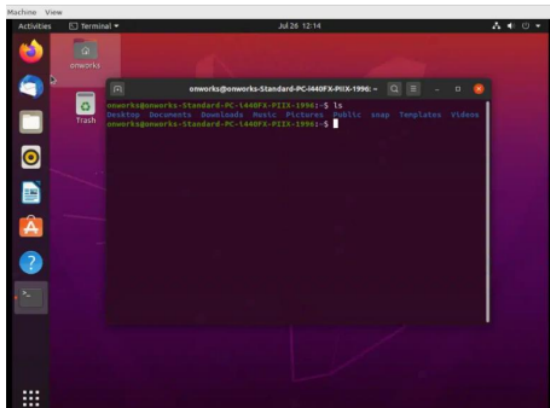
Use of CLI commands in a Linux OS.

Example 1.

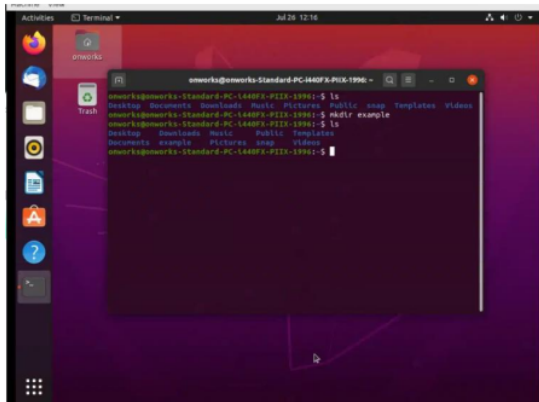


Firstly, load up Terminal with-in Linux OS

Once loaded like on the left type in 'ls' which lists files and directories in the current location.

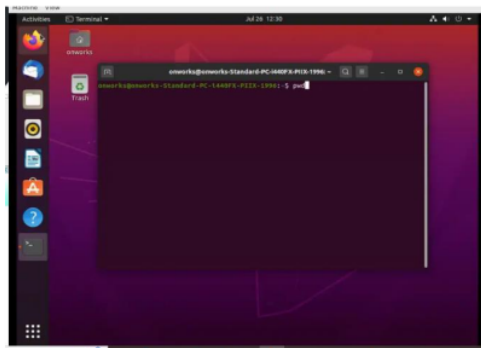


Now you'll be here, once you are you want to create an example by typing and entering 'mkdir example'.

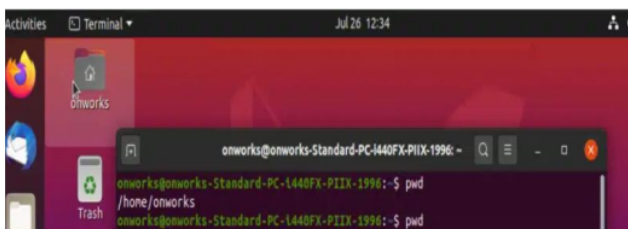


Then finally, after doing that you then type 'ls' again which would have created new directory which you can see.

Example 2.



Once you have loaded up the terminal you can type in 'pwd'



Once entered it will then show you the exact location you are, in this example is the home directory.

Command-Line Interface (CLI) commands in Linux provide users with a powerful and flexible way to interact with the operating system and perform various tasks without relying on a graphical user interface. The Linux terminal allows users to execute commands, automate tasks, and manage the system efficiently (Dandamudi, S.P). Unlike graphical user interfaces (GUIs), where users interact with visual elements, the CLI relies on text-based commands and responses.

Here are couple more examples below.

ps: Shows a list of currently running processes.

kill: Sends a signal to terminate a process. For example, kill PID terminates the process with the specified Process ID.

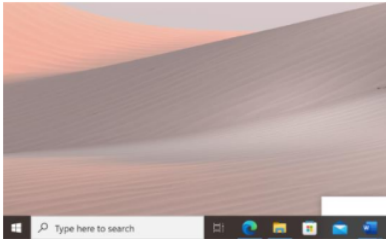
sudo: Short for "superuser do," it allows users with appropriate permissions to execute commands with administrative privileges.

These are just a few examples of the numerous CLI commands available on Linux. Learning and utilizing CLI commands can significantly enhance productivity and control over the system, making it an essential skill for Linux users. Since linux was created to function similarly to a previous operating system called unix, it may be claimed that it is a type of unix descendant and that most of the older shells and other programmes work well on it (Adekotujo, A).

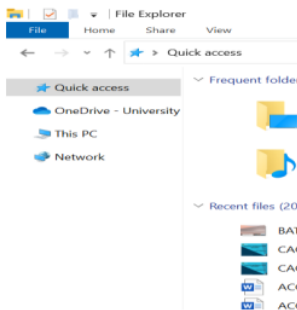
To open a terminal, you need to access 'Terminal launcher' or you can use a short cut – CTRL+ALT+T (Kirkbride, P.).

BATCH SCRIPT WINDOWS.

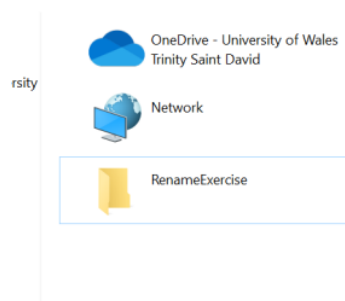
EXAMPLE 1 (RenameExercise).



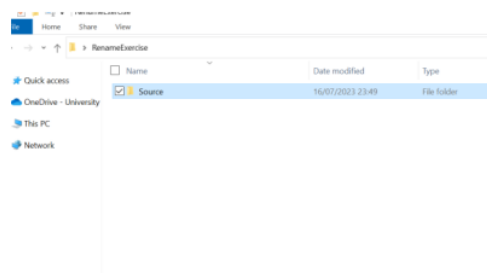
Firstly, click and load up files.



Then create a new folder.

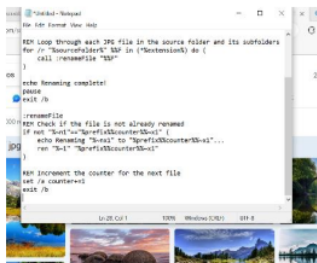


Name it "RenameExercise".



Then create a subfolder within that folder and name it "Source".

Place some JPG image files within this "Source" sub folder.



Load up a text editor like Notepad for example and create a new text file. Put the following batch script code within it - @echo off

```
set "sourceFolder=%USERPROFILE%\Desktop\RenameExercise\Source"
```

```
set "extension=.jpg"
```

```
set "prefix=mypic_"
```

```
set "counter=1"
```

```
echo Renaming all "%extension%" files in %sourceFolder%...
```

```
REM Loop through each JPG file in the source folder and its subfolders
```

```
for /r "%sourceFolder%" %%F in (*%extension%) do (
```

```
    call :renameFile "%%F"
```

```
)
```

```
echo Renaming complete!
```

```
pause
```

```
exit /b
```

```
:renameFile
```

```
REM Check if the file is not already renamed
```

```
if not "%~n1"=="%prefix%%counter%%~x1" (
```

```
    echo Renaming "%~nx1" to "%prefix%%counter%%~x1" ...
```

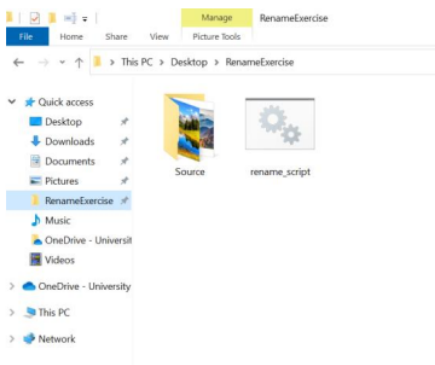
```
    ren "%~n1" "%prefix%%counter%%~x1"
```

```
)
```

```
REM Increment the counter for the next file
```

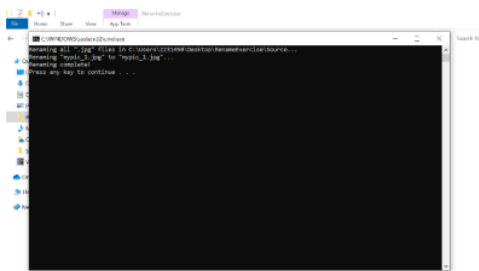
```
set /a counter+=1
```

```
exit /b
```

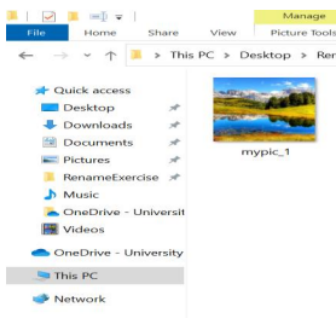


Save the Notepad file that contains the batch script with the extension .bat in the "RenameExercise" folder. Name it "rename_script.bat".

Now you can double click the "rename_script.bat" to execute the script.



This would have come up on your screen click any button as it says, it will then continue to run the script.



The batch script would have then renamed all .jpg files in the "source" folder. The files should be renamed to "mypic_1", "mypic_2" and so on.

BATCH SCRIPT WINDOWS

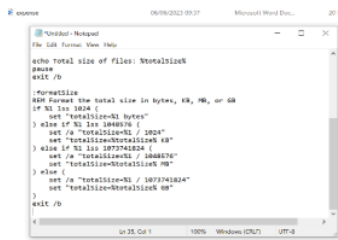
Example 2 (SizeCalculatorExercise).

RenameExercise	19/07/2023 14:07	File folder
SearchExercise	19/07/2023 14:51	File folder
SizeCalculatorExercise	19/07/2023 15:04	File folder

Firstly again, create a new folder name this one "SizeCalculatorExercise".

Name	Date modified	Type	Size
1	20/02/2023 15:46	Microsoft Word Doc...	13 KB
Drawing	13/06/2023 12:20	Microsoft Word Doc...	32 KB
expense	06/06/2023 09:37	Microsoft Word Doc...	20 KB

Then you want to add different files in the "SizeCalculatorExercise" folder all of varying sizes, these are the files that will be used to get the calculation of the total size.



```

@echo off
echo Total size of files: %totalSize%
pause
exit /b

:formatSize
REM Format the total size in bytes, KB, MB, or GB
if %totalSize% == 0 {
    set "totalSize=0 bytes"
} else if %totalSize% < 1024 {
    set "totalSize=%totalSize% bytes"
} else if %totalSize% < 1048576 {
    set "totalSize=%totalSize% KB"
} else if %totalSize% < 1073741824 {
    set "totalSize=%totalSize% MB"
} else {
    set "totalSize=%totalSize% GB"
}
exit /b

```

Save the notepad file as
"size_script.bat"

Open up a text editor like Notepad and create a file text use the batch script here – @echo off

```

set
"sourceFolder=%USERPROFILE%\Desktop\SizeCalculatorExercise"

set "totalSize=0"

echo Calculating total size of files
in %sourceFolder%...

REM Loop through each file in the
source folder and its subfolders

for /r "%sourceFolder%" %%F in (*)
do (

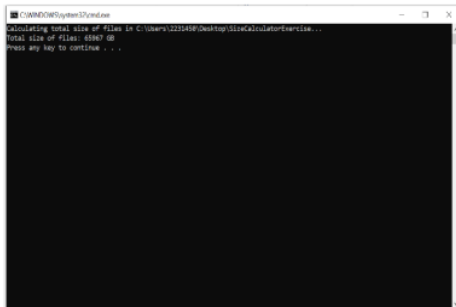
    REM Calculate the size of each file
using "%%~zF" and add it to the
total

    set /a totalSize+=%%~zF

)

REM Format the total size in bytes,
KB, MB, or GB

```



```

C:\Windows\System32\cmd.exe
Calculating total size of files in C:\Users\123456\Desktop\SizeCalculatorExercise...
Total size of files: 65967 GB
Press any key to continue . . .

```

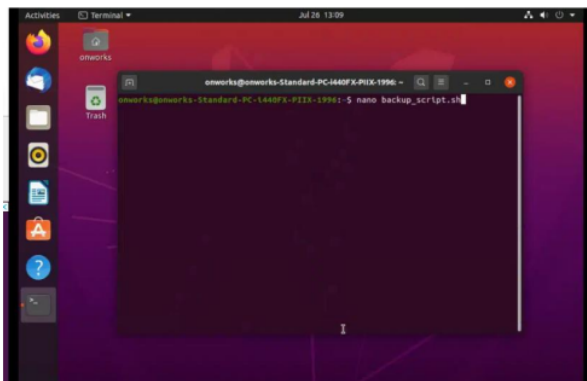
Again, like with the first batch script on this batch script double click the file saved as "size_script.bat" it will execute the batch script and give you the result of the total size of the files altogether, like my example here on the side where it's calculated the size of them files together to be 65967GB.

Windows batch scripts are a type of scripting language used to automate tasks on Windows operating systems. They are also known as batch files or .bat files. Windows batch scripts contain a series of commands written in plain text, which are executed one after another to perform specific tasks. These scripts help users automate repetitive tasks, manage system configurations, run programs, and handle various administrative tasks (Chaganti, R). Batch scripts have the file extension ".bat" or ".cmd," depending on the version of Windows and the commands used. When a user runs a batch script, the Windows Command Prompt or PowerShell reads and interprets each line of the script sequentially. It executes the commands in the same order they appear in the script (Ahmad, S.R). Common Uses of Batch Scripts: Automating backup and file management tasks,

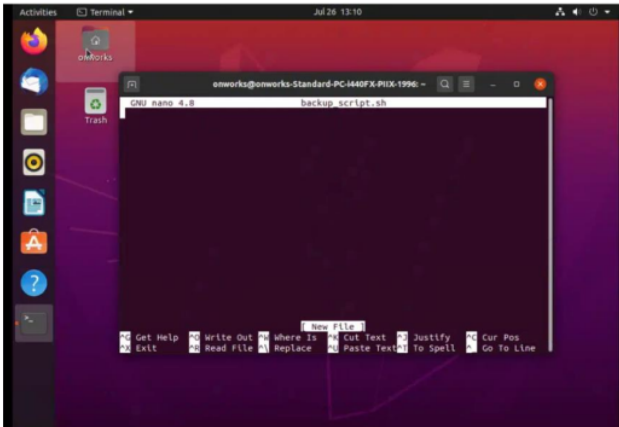
Installing or uninstalling software silently, managing services and scheduled tasks, configuring network settings, automating repetitive administrative tasks, creating simple interactive tools or menus. In general, batch scripts offer a simple and effective method for automating operations on Windows computers. Batch scripting is a useful tool for both seasoned system administrators and regular users looking to boost productivity and streamline their workflows because of its simplicity and the variety of commands and capabilities that are readily available (McFedries, P).

BATCH SCRIPTS IN LINUX OS.

Example 1.



Firstly, create a new script file: nano script.sh



You'll then be brought here.

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996: ~  
GNU nano 4.8 script.sh Modified  
chmod +x script.sh ./script.sh  
echo "hello!world"  
echo "enter the first number"  
read num1  
echo "enter the second number"  
read num2  
sum=$((num1 + num2))  
echo "the sum is: $sum"  
echo "enter a number"  
read number  
if ((number%2==0));then
```

Then type in the following code as shown on the example on the left.

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996: ~  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ nano script.sh  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ chmod +x script.sh  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ scr  
Command 'scr' not found, but there are 18 similar ones.  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ SCR  
SCR: command not found  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ script.sh  
script.sh: command not found  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ bash script.sh  
enter the first number  
23  
enter the second number  
54  
the sum is: 77  
enter a number
```

Once the operation is executed it will then have you enter the numbers you wish to and it will calculate the total amount.

Example 2.

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996: ~  
GNU nano 4.8 script.sh  
echo "enter a number"  
read number  
if ((number%2==0));  
echo "the number is even"  
else  
echo "the number is odd"  
fi
```

Enter the following code to the left to calculate if the number is odd or even.

```
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$ bash script.sh  
enter a number  
45  
the number is odd  
onworks@onworks-Standard-PC-l440FX-PIIX-1996:~$
```

Once executed it will run the operation and calculate the answer as you can see on the example.

Bash scripts in Linux make use of various features, such as loops, conditional statements, variables, and functions, enabling users to create sophisticated automation solutions. They can be used for tasks like file and directory management, software installations, system configurations, data processing, and more (Love, R).

One of the key strengths of Bash scripts lies in their ability to interact with other command-line tools, combining their functionalities and producing efficient and custom workflows. Bash scripts give users the freedom and power they need to customise their automation solutions to meet unique requirements because of the extensive community support and a comprehensive collection of tools and utilities present in the Linux environment. They play a crucial role in Linux system management because of their adaptability and efficiency, which makes computing more streamlined and effective (Cook, H).

- [1] [2] [3] ¹⁶ - Stallings, W., 2005. The Windows Operating System. *Operating Systems: Internals and Design Principles*.
- ⁸ [4] - Mudiraj, A.R., 2013. Windows linux and mac operating system booting process: a comparative study. *International Journal of Research in Computer and Communication Technology*, 2(11), pp.2278-5841.
- [5] - Rajagopal, R., 2020. A Quick Overview Of Linux. *Local Area Network Handbook*, p.335.
- ¹¹ [6] - QODIROV, F.E.O.G.L., MANSUROVA, Z.A.Q. and JURAYEV, S.S.O.G.L., 2019. MODERN MOBILE APPLICATIONS AND THEIR PECULIARITIES AND ALSO ANALYSIS AND GENERAL CHARACTERISTICS BASED ON THE ANDROID OPERATING SYSTEM. *Наука среди нас*, (5), pp.106-111.
- ³ [7] - Li, T., Brett, P., Knauerhase, R., Koufaty, D., Reddy, D. and Hahn, S., 2010, January. Operating system support for overlapping-ISA heterogeneous multi-core architectures. In *HPCA-16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture* (pp. 1-12). IEEE.
- ⁷ [8] - Fischer, T., Sadeghi, A.R. and Winandy, M., 2009, August. A pattern for secure graphical user interface systems. In *2009 20th International Workshop on Database and Expert Systems Application* (pp. 186-190). IEEE.
- ² [9] - Dinneen, J.D. and Frissen, I., 2020, April. Mac users do it differently: The role of operating system and individual differences in file management. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1-8).
- ¹⁰ [10] - Waldspurger, C.A., 2002. Memory resource management in VMware ESX server. *ACM SIGOPS Operating Systems Review*, 36(SI), pp.181-194.
- ¹ [11] - Yamin, M.M. and Katt, B., 2019. Detecting malicious windows commands using natural language processing techniques. In *Innovative Security Solutions for Information Technology and Communications: 11th International Conference, SecITC 2018, Bucharest, Romania, November 8–9, 2018, Revised Selected Papers 11* (pp. 157-169). Springer International Publishing.
- ⁵ [12] - Holmes, L., 2010. *Windows PowerShell Cookbook: The Complete Guide to Scripting Microsoft's New Command Shell*. " O'Reilly Media, Inc."
- ⁹ [13] - Mueller, J.P., 2007. *Windows Administration at the Command Line for Windows Vista, Windows 2003, Windows XP, and Windows 2000*. John Wiley & Sons.
- ¹ [14] - Dandamudi, S.P., 2005. Using Linux. *Guide to Assembly Language Programming in Linux*, pp.115-149.
- ² [15] - Adekotujo, A., Odumabo, A., Adedokun, A. and Aiyeniko, O., 2020. A Comparative Study of Operating Systems: Case of Windows, UNIX, Linux, Mac, Android and iOS. *International Journal of Computer Applications*, 176(39), pp.16-23.
- ¹⁵ [16] - Kirkbride, P. and Kirkbride, P., 2020. History and Shortcuts. *Basic Linux Terminal Tips and Tricks: Learn to Work Quickly on the Command Line*, pp.47-55.
- ⁵ [17] - Chaganti, R. and Chaganti, R., 2014. Beginning windows powershell. *Windows PowerShell Desired State Configuration Revealed*, pp.3-42.
- [18] - Ahmad, S.R., Ahmad, A., Abd Hamid, N.N.A., Rajab, A., Sharif, M., Awang, N.F. and Wook, M., 2011. Development of a single honeypot system interface.
- [19] - McFedries, P., 2004. *Microsoft Windows Vista Unleashed*. Sams Publishing.
- ¹² [20] - Love, R., 2013. *Linux system programming: talking directly to the kernel and C library*. " O'Reilly Media, Inc."

[21] - Cook, H., Moreto, M., Bird, S., Dao, K., Patterson, D.A. and Asanovic, K., 2013. A hardware evaluation of cache partitioning to improve utilization and energy-efficiency while preserving responsiveness. *ACM SIGARCH Computer Architecture News*, 41(3), pp.308-319.

FINAL

ORIGINALITY REPORT

19%

SIMILARITY INDEX

14%

INTERNET SOURCES

10%

PUBLICATIONS

17%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Wales, Lampeter Student Paper	4%
2	Submitted to Kent Institute of Business and Technology Student Paper	3%
3	Submitted to Campbellsville University Student Paper	1%
4	upcommons.upc.edu Internet Source	1%
5	Amal Alahmadi, Norah Alkhraan, Wojdan BinSaeedan. "MPSAutodetect: A Malicious PowerShell Script Detection Model Based on a Stacked Denoising Auto-Encoder", Computers & Security, 2022 Publication	1%
6	Submitted to Washington University of Science and Technology Student Paper	1%
7	Submitted to University of Gloucestershire Student Paper	1%

8	Submitted to Middlesex University Student Paper	1 %
9	Submitted to University of East London Student Paper	1 %
10	Submitted to Glasgow Caledonian University Student Paper	1 %
11	econferences.ru Internet Source	1 %
12	www.springerprofessional.de Internet Source	1 %
13	Submitted to City College Brighton and Hove Student Paper	1 %
14	www.coursehero.com Internet Source	1 %
15	Submitted to Liberty University Student Paper	<1 %
16	Submitted to University of Bedfordshire Student Paper	<1 %
17	www.glencove.com Internet Source	<1 %
18	travisaltman.com Internet Source	<1 %
19	dokumen.pub Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off