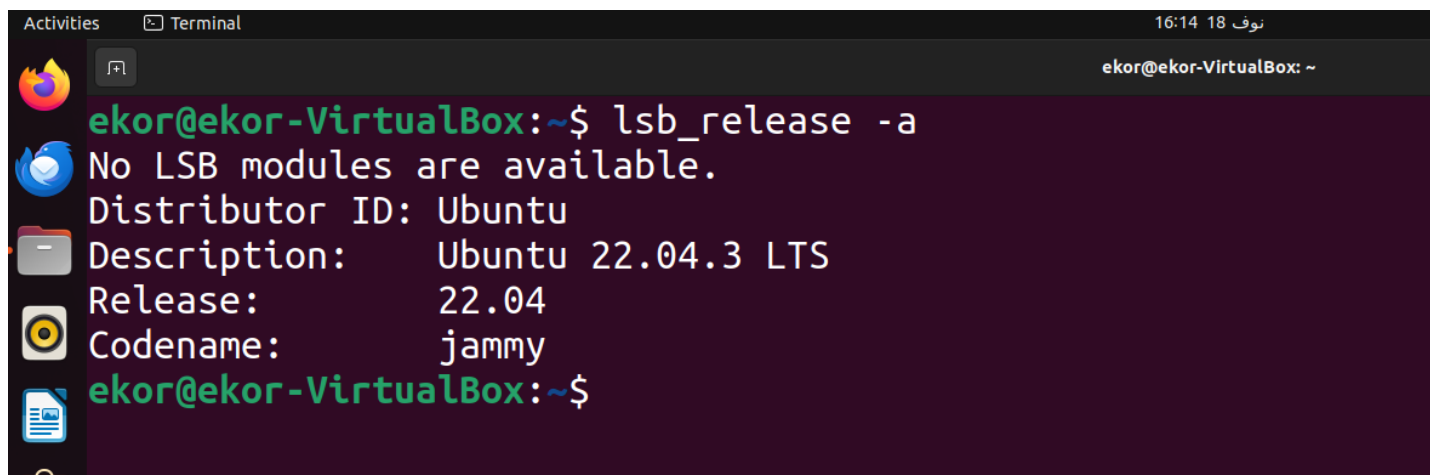


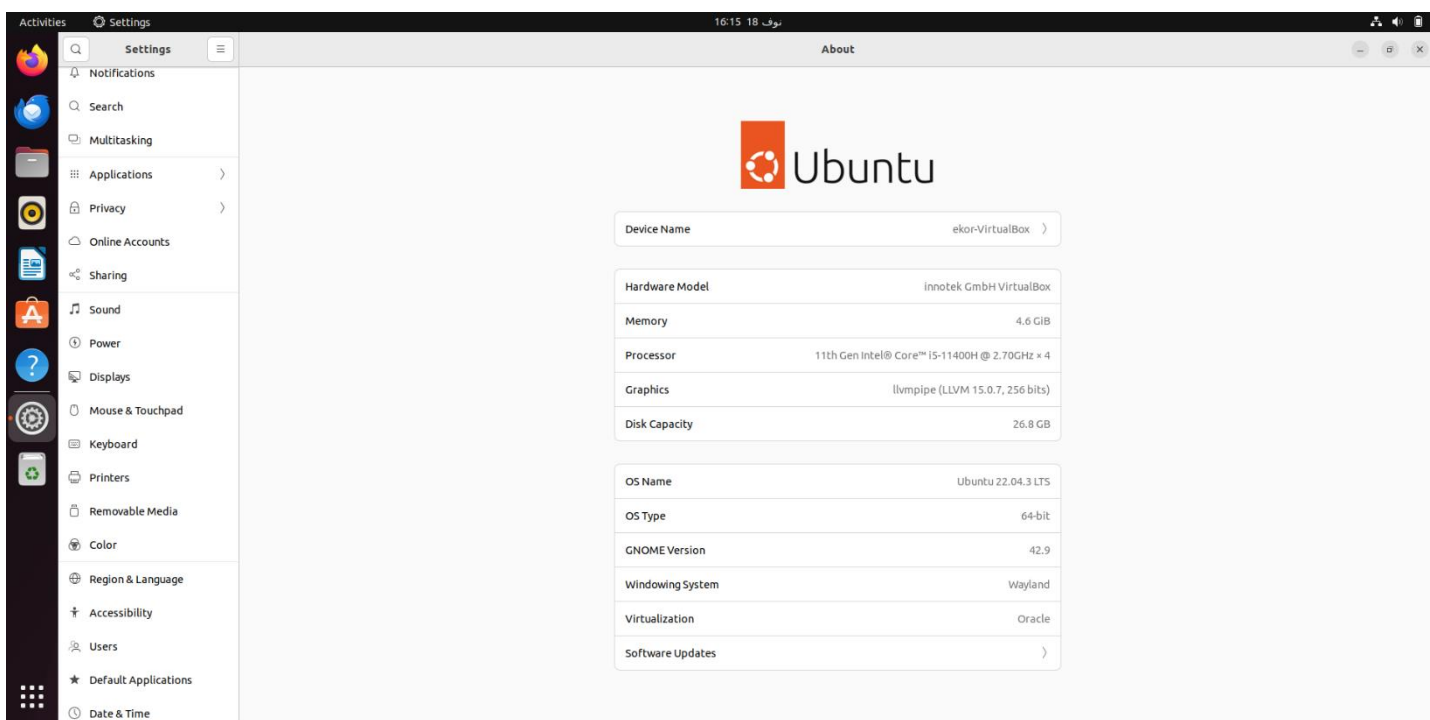
## • Task1: ( In Lab Task )

### 1. Install Ubuntu OS, then identify its version.



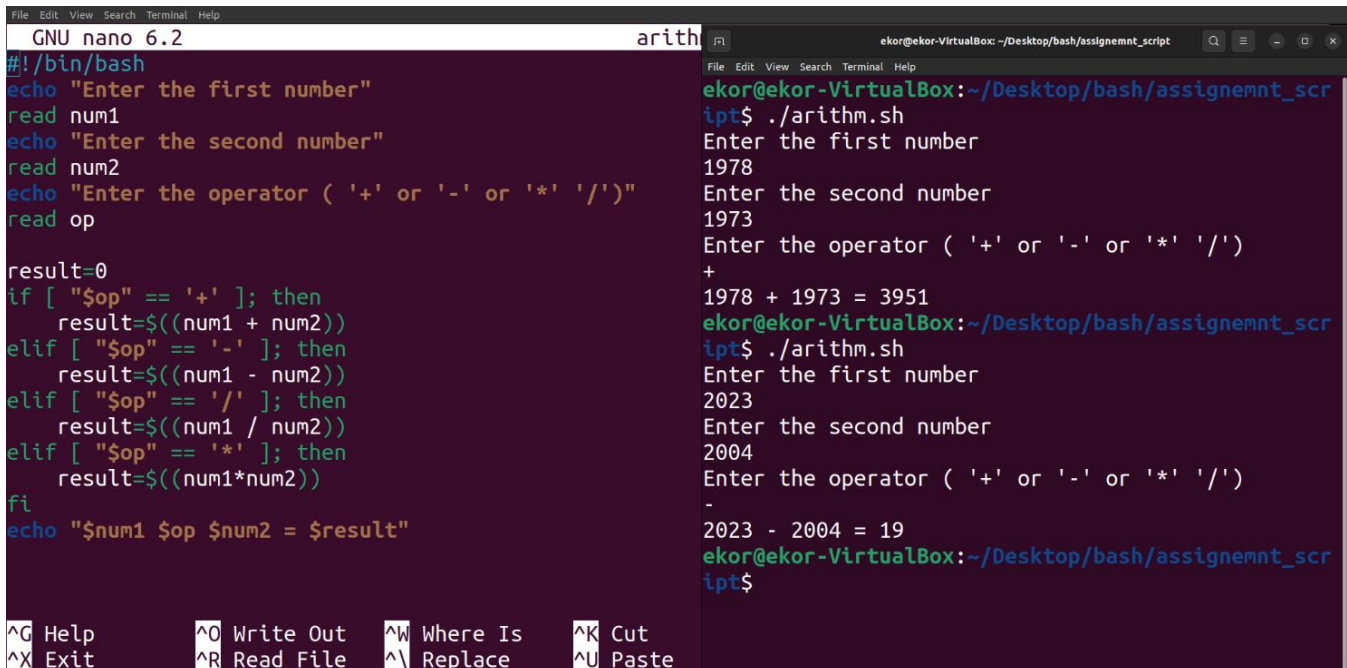
A terminal window titled 'Terminal' with a dark background. The prompt is 'ekor@ekor-VirtualBox: ~\$'. The command 'lsb\_release -a' has been executed, resulting in the following output:

```
ekor@ekor-VirtualBox:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
ekor@ekor-VirtualBox:~$
```



## 2. Carry out the following:

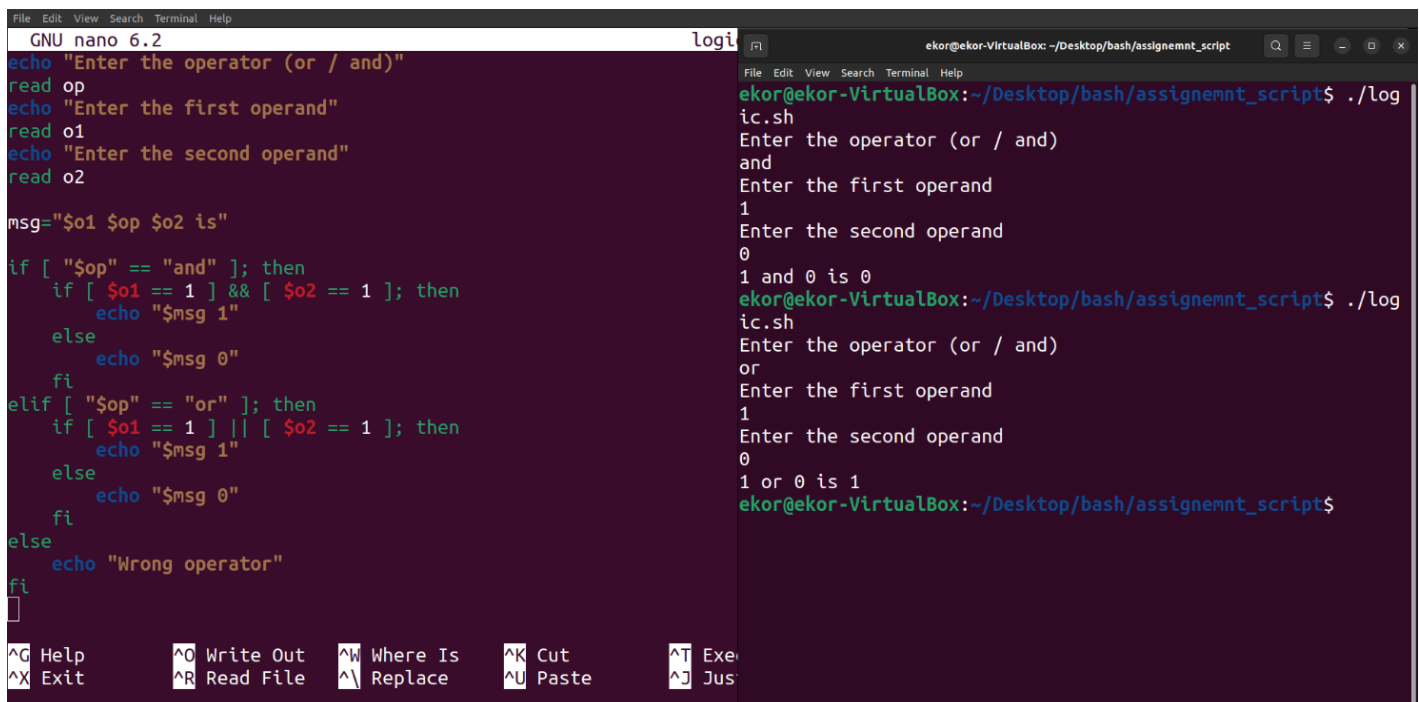
- Use “Echo” Command with different options and arguments to examine some Arithmetic, Logic and Quotes operations form user's input variables in three different examples.



```
GNU nano 6.2 arith
#!/bin/bash
echo "Enter the first number"
read num1
echo "Enter the second number"
read num2
echo "Enter the operator ( '+' or '-' or '*' '/')"
read op

result=0
if [ "$op" == '+' ]; then
    result=$((num1 + num2))
elif [ "$op" == '-' ]; then
    result=$((num1 - num2))
elif [ "$op" == '/' ]; then
    result=$((num1 / num2))
elif [ "$op" == '*' ]; then
    result=$((num1*num2))
fi
echo "$num1 $op $num2 = $result"
```

```
ekor@ekor-VirtualBox: ~/Desktop/bash/assignemnt_script
ipt$ ./arithm.sh
Enter the first number
1978
Enter the second number
1973
Enter the operator ( '+' or '-' or '*' '/')
+
1978 + 1973 = 3951
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./arithm.sh
Enter the first number
2023
Enter the second number
2004
Enter the operator ( '+' or '-' or '*' '/')
-
2023 - 2004 = 19
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$
```

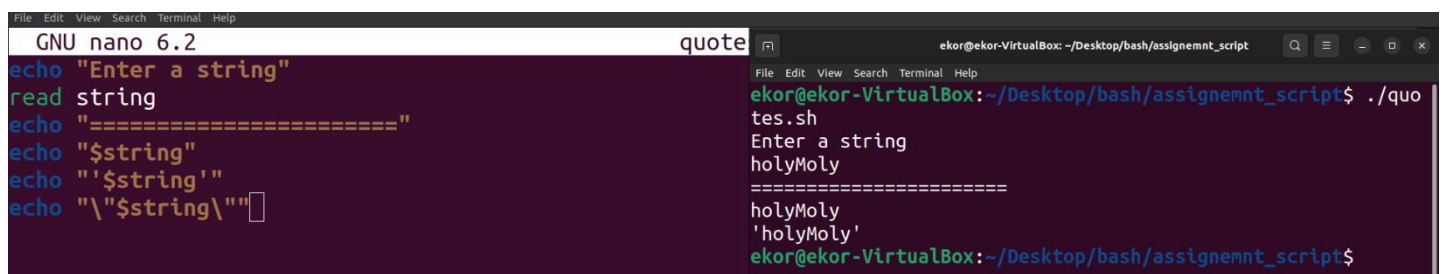


```
GNU nano 6.2 logic
echo "Enter the operator (or / and)"
read op
echo "Enter the first operand"
read o1
echo "Enter the second operand"
read o2

msg="$o1 $op $o2 is"

if [ "$op" == "and" ]; then
    if [ $o1 == 1 ] && [ $o2 == 1 ]; then
        echo "$msg 1"
    else
        echo "$msg 0"
    fi
elif [ "$op" == "or" ]; then
    if [ $o1 == 1 ] || [ $o2 == 1 ]; then
        echo "$msg 1"
    else
        echo "$msg 0"
    fi
else
    echo "Wrong operator"
fi
```

```
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./logic.sh
Enter the operator (or / and)
and
Enter the first operand
1
Enter the second operand
0
1 and 0 is 0
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./logic.sh
Enter the operator (or / and)
or
Enter the first operand
1
Enter the second operand
0
1 or 0 is 1
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$
```



```
GNU nano 6.2 quote
echo "Enter a string"
read string
echo "====="
echo "$string"
echo "'$string'"
echo "\"$string\""
```

```
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./quotes.sh
Enter a string
holyMoly
=====
holyMoly
'holyMoly'
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$
```

- b. Use “Cat” command with different options and arguments in three different - examples to create single or multiple files, view content of a file, and redirect output in terminal or files.

```
File Edit View Search Terminal Help
ekor@ekor-VirtualBox:~/Desktop/bash$ cat > file1.txt
this is file 1 ..... blablabla
ekor@ekor-VirtualBox:~/Desktop/bash$ cat > file2.txt > file3.txt
this is file 3 ..... writing some irrelevant words
ekor@ekor-VirtualBox:~/Desktop/bash$ cat file3.txt > file2.txt
ekor@ekor-VirtualBox:~/Desktop/bash$ cat file1.txt
this is file 1 ..... blablabla
ekor@ekor-VirtualBox:~/Desktop/bash$ cat file2.txt
this is file 3 ..... writing some irrelevant words
ekor@ekor-VirtualBox:~/Desktop/bash$ cat file3.txt
this is file 3 ..... writing some irrelevant words
ekor@ekor-VirtualBox:~/Desktop/bash$
```

- c. In light of that scenario, create a simple Shell script for user's login process as described in the given scenario and as shown in figure (1). Use your Name as a user login name and assume your password.

```
GNU nano 6.2                                llogin.sh *
username="ekor" password="ekor" attempts=3
while [ $attempts -ne 0 ]
do
    echo "Enter your username"
    read user
    echo "Enter you password"
    read pass

    if [ "$user" == "$username" ] && [ "$pass" == "$password" ]
    then
        echo "You are in!"
        exit 0
    else
        ((attempts--))
        if [ $attempts == 0 ]
        then
            echo "=====
            echo "You account is locked!"
            echo "=====
            exit 0
        fi
        echo "====Wrong username or password===="
        echo "====You havs $attempts tries===="
    fi
done

File Edit View Search Terminal Help
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./llogin
.sh
Enter your username
ekor
Enter you password
ekor
You are in!
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ ./llogin
.sh
Enter your username
test
Enter you password
test
====Wrong username or password====
====You havs 2 tries====
Enter your username
rrr
Enter you password
rrr
====Wrong username or password====
====You havs 1 tries====
Enter your username
adfs
Enter you password
afds
=====
You account is locked!
=====
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_sekor@ekor-Virtu
alBox:~/Desktop/bash/assignemnt_sekor@ekor-VirtualBox:~/Desktop
Jekor@ekor-VirtualBox:~/Desktopekor@ekor-VirtualBox:~/Desktop/ba
ekor@ekor-VirtualBox:~/Desktopekor@ekor-VirtualBox:~/Desktop/ba
```

Help Write Out Where Is Cut  
Exit Read File Replace Paste

**d. Form the given scenario, Analyze the usage of different types of loops and decision making conditions in Shell script for the given user's login process.**

First line I used while loop, the condition will keep asking the user for username and password till the attempts be zero.

The **if** condition here is checking if the user entered true username and password or not. If the input is true the loop will terminate.

- **Task 2:**

- a. **Describe kernel types and link each type with one of Linux operating systems.**

- **Monolithic Kernel:** A monolithic kernel is an operating system architecture where **the entire system runs in kernel mode**. In this design, the kernel consists of a single, large executable that includes various services such as memory management, device drivers, file system management, and process management, among others. Consequently, all these services can directly interact with each other, bypassing the need for IPC (Inter-Process Communication) mechanisms, which simplifies communication and reduces overhead.

- Used by Ubuntu.

- **Microkernels:** A microkernel architecture has only core functionality like process and memory management in kernel mode, while other services like device drivers, file systems, and networking are separate user-mode processes. This design leads to a more modular, fault-tolerant, and secure system, as individual components can be updated or replaced without affecting the entire system.

- Used by MINIX and Arch

- **Hybrid Kernels:** Hybrid kernels combine aspects of both monolithic and microkernel architectures. They run some core services in kernel mode and others in user mode, offering a balance between performance and modularity. Hybrid kernels can adapt their design according to specific requirements, incorporating the best aspects of both architectures.

- Used by Fedora

- b. Based on your exploration of Linux operating system, critically explain the advantages and disadvantages of kali Linux as one of the most popular Linux based distribution. Then from your point of view, present briefly some of its features such as desktop environment and kernel type and applications such as network analysis or system monitoring applications.

**Advantages of Kali Linux:**

- **Specialized for Security Testing:** Kali is tailored for penetration testing, offering a comprehensive suite of security tools.
- **Extensive Toolset:** Pre-installed security tools save time for professionals, covering a wide range of testing scenarios.
- **Regular Updates:** Active maintenance ensures access to the latest security tools and features.
- **Customizability:** Based on Debian, Kali allows users to customize the system to their specific security needs.

**Disadvantages of Kali Linux:**

- **Specialized Nature:** Not ideal for general-purpose computing, limiting its everyday use.
- **Tool Overwhelm:** Extensive toolset might be overwhelming for users not engaged in security testing.
- **Potential Instability:** Rapid updates could introduce changes impacting system stability.
- **Customization Complexity:** Customization may require a good understanding of Linux systems, posing a challenge for beginners.

**Features of Kali Linux:** 9. **Desktop Environment:** Primarily uses GNOME, offering a user-friendly interface.

- **Kernel Type:** Utilizes a customized Linux kernel optimized for security tasks.
- **Network Analysis Tools:** Includes Wireshark, Nmap, and Aircrack-ng for in-depth network assessment.
- **System Monitoring:** Provides tools like htop and atop for real-time system resource monitoring.

**C. Compare and contrast the required system specifications for one play Gamming applications if it has been used in basic operating system platform such as Microsoft Windows, Android OS, and Linux distribution: Red Hat or Debian. Then, discuss how it could be matched with each operating systems, and suggest any programs needed to make it works with that application, if found.**

**Stardew Valley System Requirements:**

**1. Linux:**

- OS: Ubuntu 16.04 or greater
- Processor: 2 GHz
- Memory: 2 GB RAM
- Graphics: 256 MB video memory, OpenGL 2 compatible card
- Storage: 500 MB available space

**2. Windows:**

- OS: Windows 7 or greater
- Processor: 2 GHz
- Memory: 2 GB RAM
- Graphics: 256 MB video memory, OpenGL 2 compatible card
- Storage: 500 MB available space

**3. Android:**

- OS: Android 4.4 or later
- Processor: 1.2 GHz processor or higher
- Memory: 2 GB RAM
- Storage: 300 MB available space

**D. Critically evaluate the benefits and drawbacks of using Ubuntu Linux rather than Kali Linux. For example: The desktop environment, application programs, and Web browsers.**

**Ubuntu Linux:**

Benefits:

1. User-Friendly Desktop Environment:
  - Pros: Ubuntu uses a user-friendly desktop environment, such as GNOME or Unity, making it accessible for users familiar with other operating systems like Windows or macOS.
2. Wide Range of Application Programs:
  - Pros: Ubuntu has a vast repository of software available through its package manager, making it easy to install and update various applications. It is a general-purpose Linux distribution suitable for a variety of tasks.
3. Web Browsers:
  - Pros: Ubuntu supports a wide range of web browsers, including popular ones like Firefox, Chrome, and others. Users have the flexibility to choose their preferred browser.
4. Community Support:
  - Pros: Ubuntu has a large and active community, providing extensive support through forums, documentation, and online resources. This makes problem-solving and learning easier for users.

Drawbacks:

1. Security Emphasis:
  - Cons: Ubuntu is not designed with a primary focus on security and penetration testing, making it less suitable for ethical hacking or cybersecurity tasks compared to specialized distributions like Kali Linux.

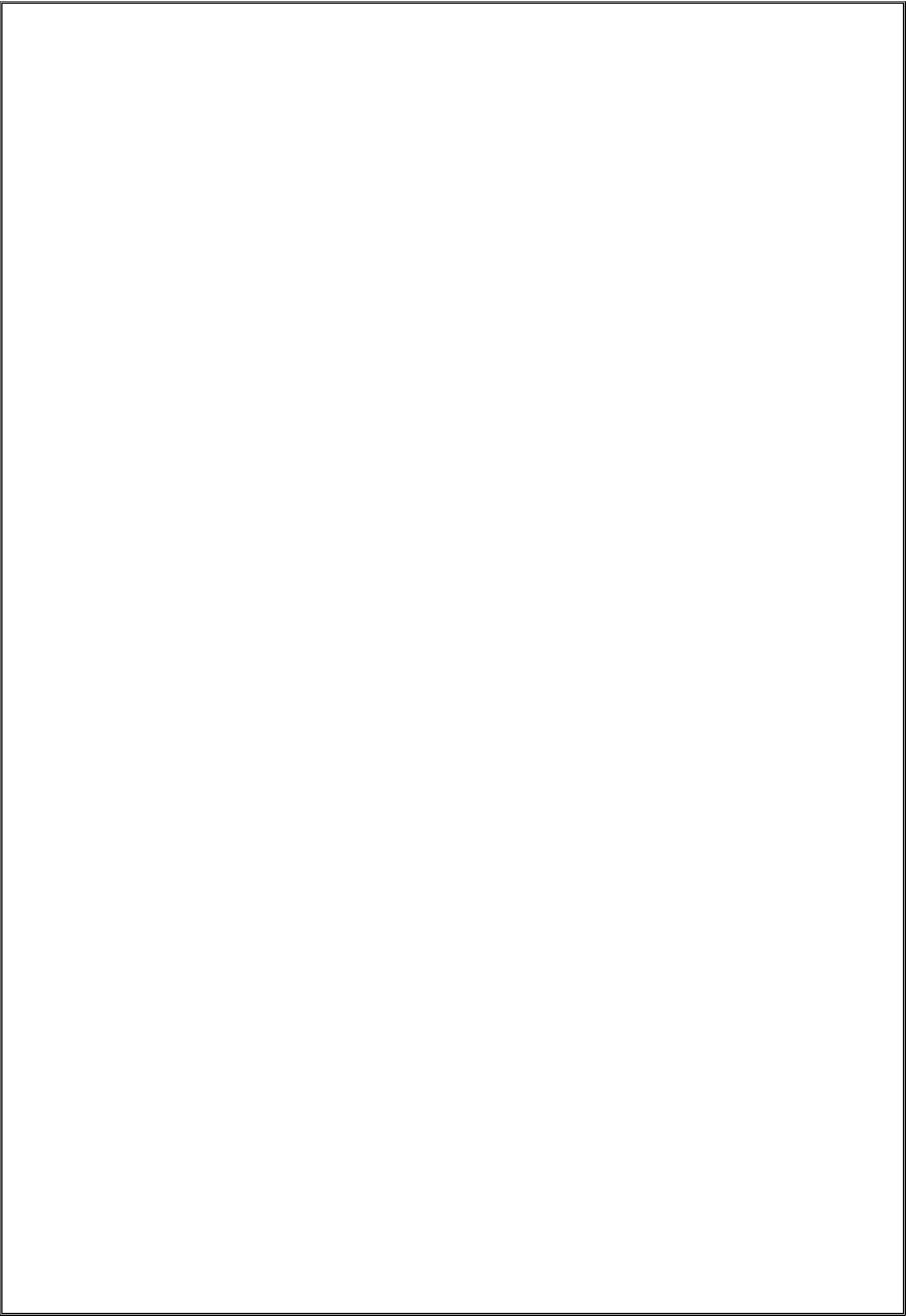
**Kali Linux:**

Benefits:

1. Specialized for Penetration Testing:
  - Pros: Kali Linux is specifically designed for penetration testing, ethical hacking, and security-related tasks. It comes pre-installed with a variety of security tools, making it a go-to choice for cybersecurity professionals.
2. Security-Focused Desktop Environment:
  - Pros: Kali Linux uses a customized desktop environment optimized for security tasks. It provides a focused and efficient environment for penetration testers.
3. Pre-Installed Security Tools:
  - Pros: Kali Linux includes a vast array of pre-installed security tools for network analysis, vulnerability assessment, and penetration testing. This eliminates the need for users to manually install these tools.

Drawbacks:

1. Less User-Friendly for General Use:
  - Cons: Kali Linux may not be as user-friendly for general tasks as Ubuntu. Its primary purpose is security testing, and it might lack the convenience features found in more general-purpose distributions.
2. Limited Application Variety





- **Task 3:**

**A. From your exploration, contrast shell types and extract the relations for Linux Debian distributions**

- **sh (Bourne shell)** is a shell command-line interpreter, for Unix/Unix-like operating systems. It provides some built-in commands.
- **The GNU Bourne-Again Shell:** Bash, the default shell, is an extended version of the Bourne Shell (`sh`). Debian scripts and utilities are often written with Bash compatibility in mind.
- **C Shell Family:** Debian supports shells like Tcsh and C Shell (`csh`). While less common for scripting, they offer different interactive features and syntax.
- **KornShell** is a Unix shell which was developed by David Korn at Bell Labs in the early 1980s and announced at USENIX on July 14, 1983. The initial development was based on Bourne shell source code.
- **The Z shell** is a Unix shell that can be used as an interactive login shell and as a command interpreter for shell scripting. Zsh is an extended Bourne shell with many improvements, including some features of Bash, ksh, and tcsh. Zsh was created by Paul Falstad in 1990 while he was a student at Princeton University.

**Relations for Debian Distributions:**

- **Debian Almquist Shell (dash)** is chosen as the default system shell for Debian and its derivatives to optimize system startup and script execution.
- While dash is the default system shell, users can still use other shells like Bash for their interactive sessions.
- Debian's choice of dash as the default shell aligns with the distribution's focus on stability and efficiency.

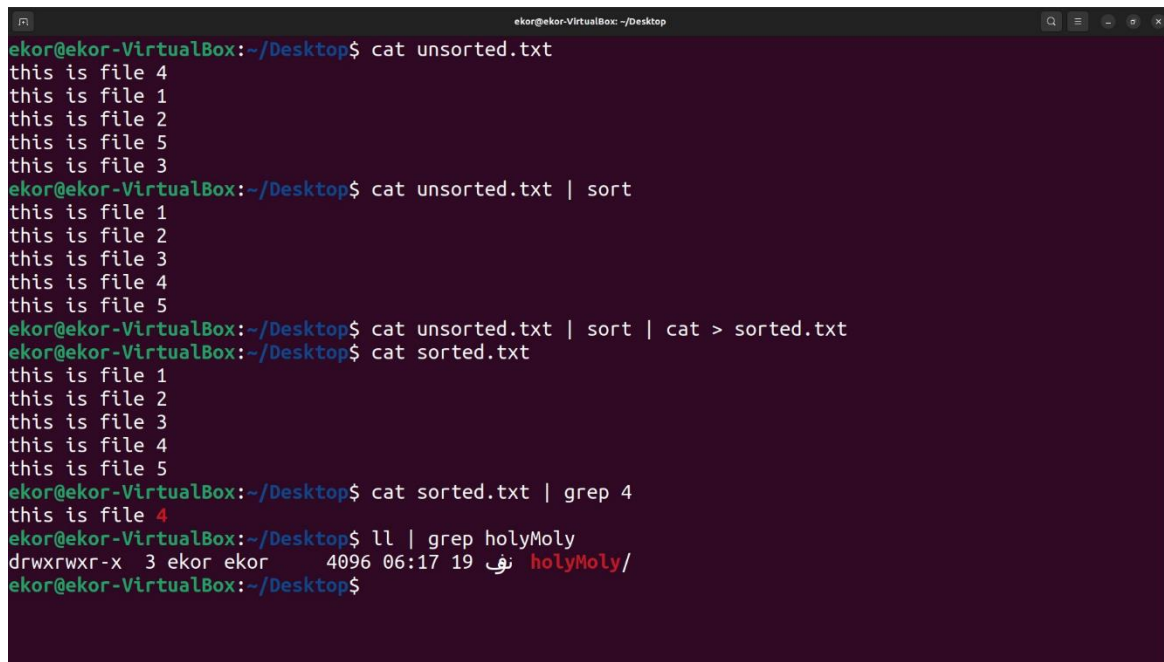
**B. List some of Ubuntu text editors and create text file using one of them.**

- VIM
- GNU Nano
- Gedit
- Kate



```
ekor@ekor-VirtualBox: ~/Desktop/bash/assignemnt_script
This is vim text editor
text text text
nine one one,
who's there?
```

**C. Evaluate the Pipe method for two commands, such that output of one command serves as input to the next, with many options and arguments, and examine that with at least three different examples.**



```
ekor@ekor-VirtualBox: ~/Desktop
ekor@ekor-VirtualBox:~/Desktop$ cat unsorted.txt
this is file 4
this is file 1
this is file 2
this is file 5
this is file 3
ekor@ekor-VirtualBox:~/Desktop$ cat unsorted.txt | sort
this is file 1
this is file 2
this is file 3
this is file 4
this is file 5
ekor@ekor-VirtualBox:~/Desktop$ cat unsorted.txt | sort | cat > sorted.txt
ekor@ekor-VirtualBox:~/Desktop$ cat sorted.txt
this is file 1
this is file 2
this is file 3
this is file 4
this is file 5
ekor@ekor-VirtualBox:~/Desktop$ cat sorted.txt | grep 4
this is file 4
ekor@ekor-VirtualBox:~/Desktop$ ll | grep holyMoly
drwxrwxr-x 3 ekor ekor 4096 06:17 19 نف holyMoly/
ekor@ekor-VirtualBox:~/Desktop$
```

## **D Critically evaluate the usage of Redirected method using "tail" and "head" commands for text files with example**

### **Tail Command**

#### **Advantages:**

- Real-time monitoring
- Customizable output
- Combining with other commands

#### **Disadvantages:**

- Limited to end of file
- Random access not efficient

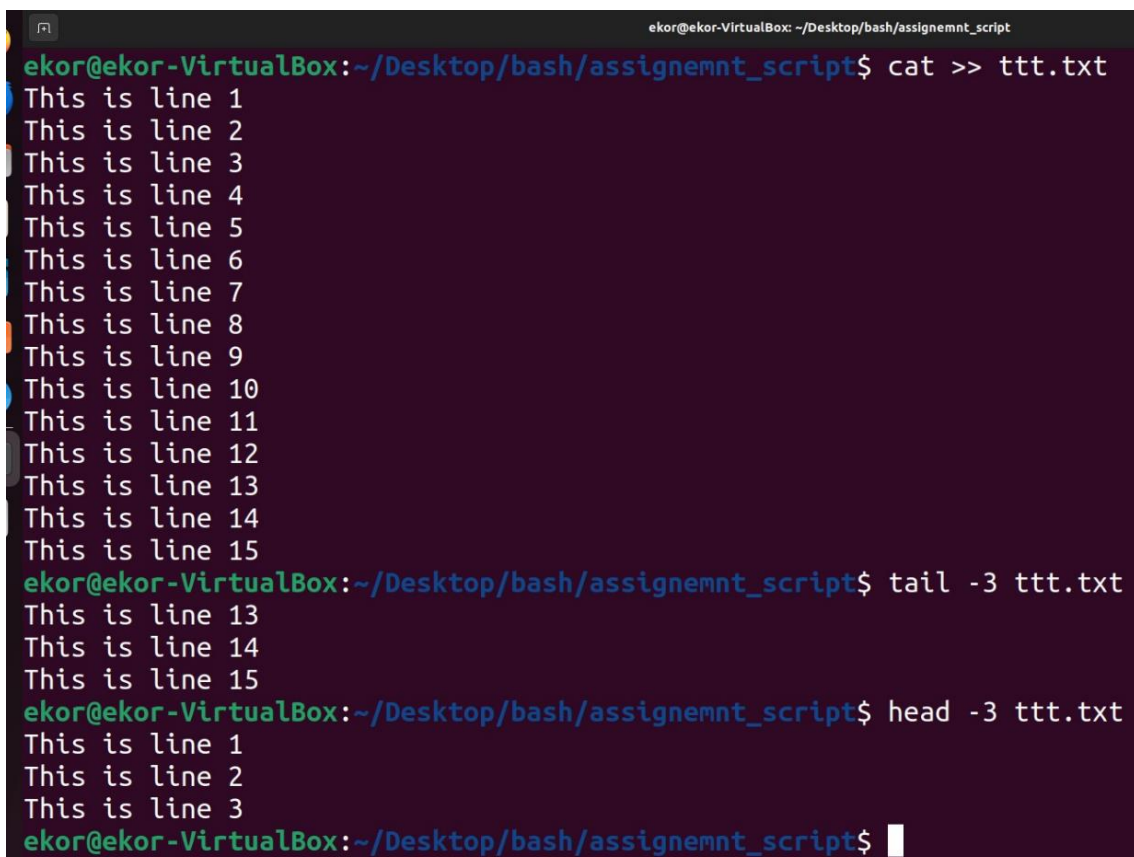
### **Head Command**

#### **Advantages:**

- Quick Preview
- Efficient Output
- Readability

#### **Disadvantages:**

- Limited to the beginning
- Not real time



```
ekor@ekor-VirtualBox: ~/Desktop/bash/assignemnt_script
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ cat >> ttt.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
This is line 11
This is line 12
This is line 13
This is line 14
This is line 15
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ tail -3 ttt.txt
This is line 13
This is line 14
This is line 15
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$ head -3 ttt.txt
This is line 1
This is line 2
This is line 3
ekor@ekor-VirtualBox:~/Desktop/bash/assignemnt_script$
```