KLE Technological University | Creating Value, Leveraging Knowledge

Dr. M. S. Sheshgiri Campus, Belagavi

Department of

Electronics and Communication Engineering

Journal

Of

Embedded Linux

(24EECE447)

Submitted By:

NAME	USN
Pratik P Patil	02FE21BEC063

Semester: VII, 2024-2025

KLE Technological University,
Dr. M. S. Sheshgiri College of Engineering and Technology
BELAGAVI-590 008
2024-2025

Basic Steps To Configure the Codes

1. Create a file using terminal with the command
   "gedit experiment_name.sh"

2. Write the code in the following script saved.

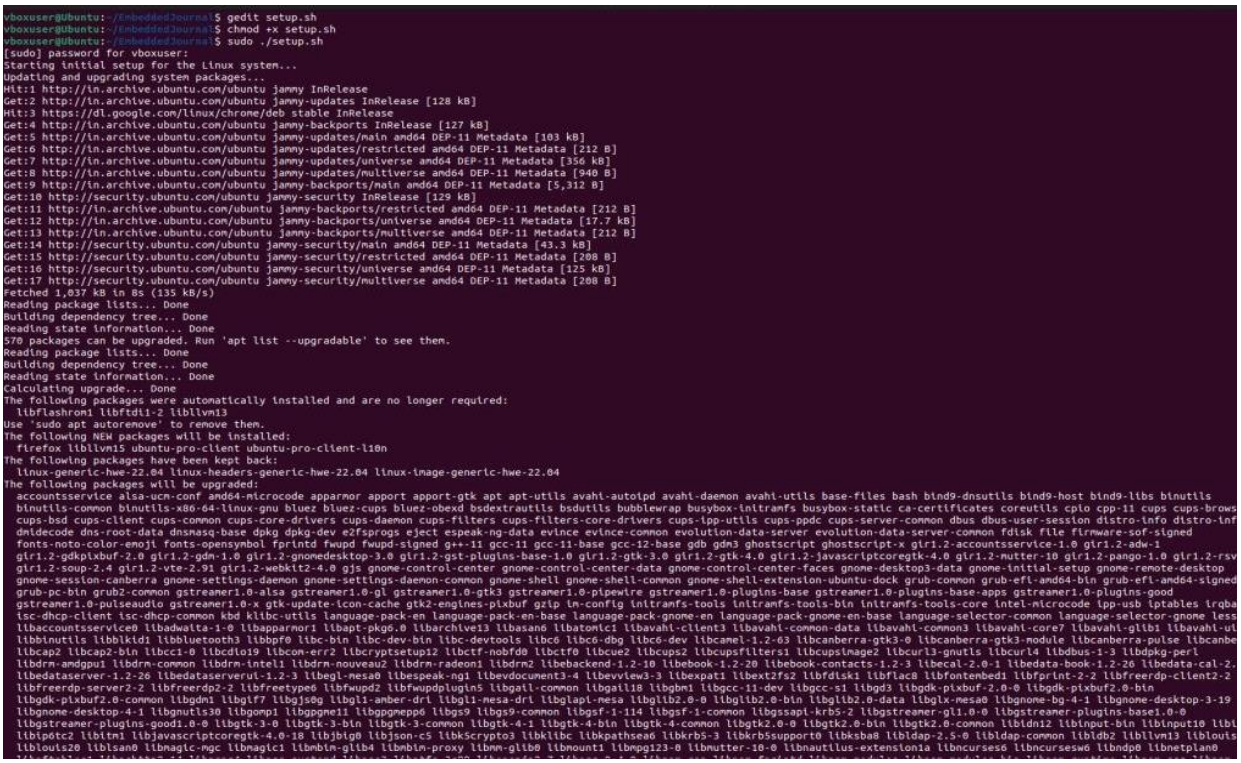
3. Make the script executable with the following command
   "chmod +x experiment_name.sh"

4. Execute the script with command
   "./ experiment_name.sh"

Experiment No. 1: Initial Setup
Code:

```
#!/bin/bash
# Ensure the script is run as root
if [ "$EUID" -ne 0 ]; then
echo "Please run as root"
exit 1
fi
echo "Starting initial setup for the Linux system..."
# Step 1: Update and Upgrade System Packages
echo "Updating and upgrading system packages..."
apt update -y && apt upgrade -y
# Step 2: Install Essential Development Tools
echo "Installing essential development tools..."
apt install -y build-essential git vim curl wget python3 python3-pip
# Step 3: Set up Firewall (optional, adjust rules as needed)
echo "Setting up a basic firewall..."
ufw allow OpenSSH
ufw enable
ufw status
# Step 4: Create a New User (optional, replace 'username' with desired name)
NEW_USER="username"
echo "Creating a new user '$NEW_USER'..."
adduser --gecos "" $NEW_USER
usermod -aG sudo $NEW_USER # Step 5: Set up SSH Key Authentication (for the new user)
echo "Setting up SSH key authentication for $NEW_USER..."
sudo -u $NEW_USER mkdir -p /home/$NEW_USER/.ssh
sudo -u $NEW_USER chmod 700 /home/$NEW_USER/.ssh
echo "Paste your public SSH key below (Ctrl+D to save):"
cat >> /home/$NEW_USER/.ssh/authorized_keys
chmod 600 /home/$NEW_USER/.ssh/authorized_keys
chown -R $NEW_USER:$NEW_USER /home/$NEW_USER/.ssh
# Step 6: Clean Up Unused Packages
echo "Cleaning up unused packages..."
apt autoremove -y
apt clean
# Step 7: Reboot the System
echo "Rebooting the system to apply changes..."
reboot
echo "Initial setup complete!"
```

Output:

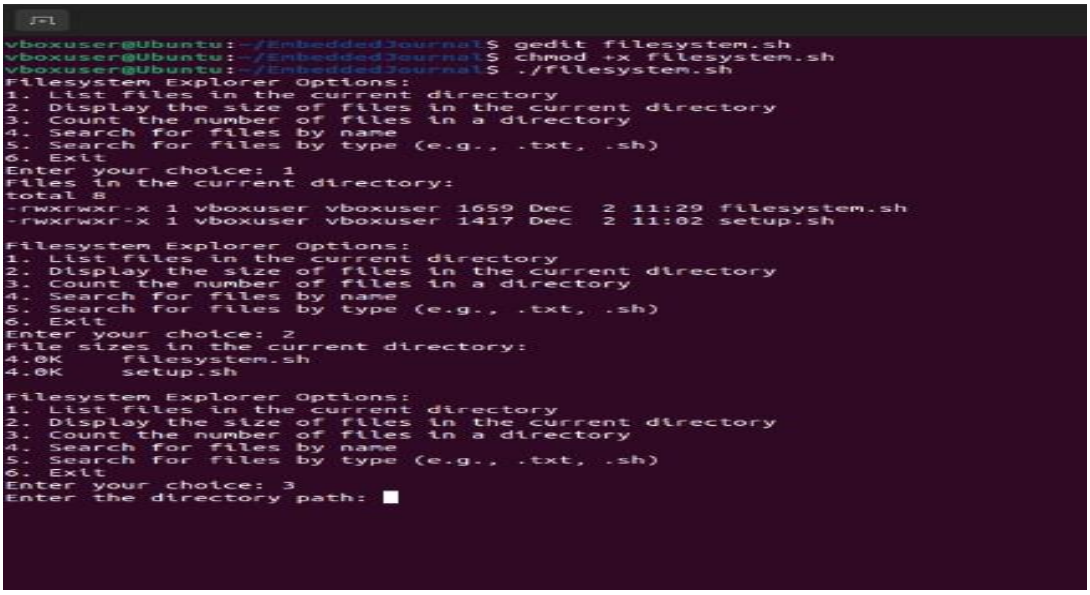

vboxuser@Ubuntu:~/EmbeddedJournal$ gedit setup.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ chmod +x setup.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ sudo ./setup.sh
[sudo] password for vboxuser:
Starting initial setup for the Linux system...
Updating and upgrading system packages...
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 https://dl.google.com/linux/chrome/deb stable InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [103 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 DEP-11 Metadata [212 B]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [356 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:9 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [5,312 B]
Get:10 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-11 Metadata [212 B]
Get:12 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [17.7 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-11 Metadata [212 B]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.3 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Fetched 1,037 kB in 8s (135 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
570 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
libflashrom1 libftdi1-2 libllvm13
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
firefox libllvm15 ubuntu-pro-client ubuntu-pro-client-l10n
The following packages have been kept back:
linux-generic-hwe-22.04 linux-headers-generic-hwe-22.04 linux-image-generic-hwe-22.04
The following packages will be upgraded:
accountsservice alsa-ucm-conf amd64-microcode apparmor apport apport-gtk apt apt-utils avahi-autoipd avahi-daemon avahi-utils base-files bash bind9-dnsutils bind9-host bind9-libs binutils
binutils-common binutils-x86-64-linux-gnu bluez bluez-cups bluez-obexd bsdextrautils bsdutils bubblewrap busybox-initramfs busybox-static ca-certificates coreutils cpio cpp-11 cups cups-brows
cups-bsd cups-client cups-common cups-core-drivers cups-daemon cups-filters cups-filters-core-drivers cups-ipp-utils cups-ppdc cups-server-common dbus dbus-user-session distro-info distro-inf
dmidecode dns-root-data dnsmasq-base dpkg dpkg-dev e2fsprogs eject espeak-ng-data evince evince-common evolution-data-server evolution-data-server-common fdisk file firmware-sof-signed
fonts-noto-color-emoji fonts-opensymbol fprintd fwupd fwupd-signed g++-11 gcc-11 gcc-11-base gcc-12-base gdb gdm3 ghostscript ghostscript-x gir1.2-accountsservice-1.0 gir1.2-adw-1
gir1.2-gdkpixbuf-2.0 gir1.2-gdm-1.0 gir1.2-gnomedesktop-3.0 gir1.2-gst-plugins-base-1.0 gir1.2-gtk-3.0 gir1.2-gtk-4.0 gir1.2-javascriptcoregtk-4.0 gir1.2-mutter-10 gir1.2-pango-1.0 gir1.2-rsv
gir1.2-soup-2.4 gir1.2-vte-2.91 gir1.2-webkit2-4.0 gjs gnome-control-center gnome-control-center-data gnome-control-center-faces gnome-desktop3-data gnome-initial-setup gnome-remote-desktop
gnome-session-canberra gnome-settings-daemon gnome-settings-daemon-common gnome-shell gnome-shell-common gnome-shell-extension-ubuntu-dock grub-common grub-efi-amd64-bin grub-efi-amd64-signed
grub-pc-bin grub2-common gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-pipewire gstreamer1.0-plugins-base gstreamer1.0-plugins-base-apps gstreamer1.0-plugins-good
gstreamer1.0-pulseaudio gstreamer1.0-x gtk-update-icon-cache gtk2-engines-pixbuf gzip lm-config initramfs-tools initramfs-tools-bin initramfs-tools-core intel-microcode ipp-usb iptables irqba
isc-dhcp-client isc-dhcp-common kbd klibc-utils language-pack-en language-pack-en-base language-pack-gnome-en language-pack-gnome-en-base language-selector-common language-selector-gnome less
libaccountsservice0 libadwaita-1-0 libapparmor1 libapt-pkg6.0 libarchive13 libasan6 libatomic1 libavahi-client3 libavahi-common-data libavahi-common3 libavahi-core7 libavahi-glib1 libavahi-ui
libbinutils libblkid1 libbluetooth3 libbpf0 libc-bin libc-dev-bin libc-devtools libc6 libc6-dbg libc6-dev libcamel-1.2-63 libcanberra-gtk3-0 libcanberra-gtk3-module libcanberra-pulse libcanbe
libcap2 libcap2-bin libcc1-0 libcdio19 libcon-err2 libcryptsetup12 libctf-nobfd0 libctf0 libcue2 libcups2 libcupsfilters1 libcupsimage2 libcurl3-gnutls libcurl4 libdbus-1-3 libdpkg-perl
libdrm-amdgpu1 libdrm-common libdrm-intel1 libdrm-nouveau2 libdrm-radeon1 libdrm2 libebackend-1.2-10 libebook-1.2-20 libebook-contacts-1.2-3 libecal-2.0-1 libedata-book-1.2-26 libedata-cal-2.
libedataserver-1.2-26 libedataserverui-1.2-3 libegl-mesa0 libespeak-ng1 libevdocument3-4 libevview3-3 libexpat1 libext2fs2 libfdisk1 libflac8 libfontenbed1 libfprint-2-2 libfreerdp-client2-2
libfreerdp-server2-2 libfreerdp2-2 libfreetype6 libfwupd2 libfwupdplugin5 libgail-common libgail18 libgbm1 libgcc-11-dev libgcc-s1 libgd3 libgdk-pixbuf-2.0-0 libgdk-pixbuf2.0-bin
libgdk-pixbuf2.0-common libgdm1 libgif7 libgjs0g libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglib2.0-0 libglib2.0-bin libglib2.0-data libglx-mesa0 libgnome-bg-4-1 libgnome-desktop-3-19
libgnome-desktop-4-1 libgnutls30 libgomp1 libgpgme11 libgpgmepp6 libgs9 libgs9-common libgsf-1-114 libgsf-1-common libgssapi-krb5-2 libgstreamer-gl1.0-0 libgstreamer-plugins-base1.0-0
libgstreamer-plugins-good1.0-0 libgtk-3-0 libgtk-3-bin libgtk-3-common libgtk-4-1 libgtk-4-bin libgtk-4-common libgtk2.0-0 libgtk2.0-bin libgtk2.0-common libidn12 libinput-bin libinput10 libi
libipotc2 libitm1 libjavascriptcoregtk-4.0-18 libjbig0 libjson-c5 libkScrypto3 libklibc libkpathsea6 libkrb5-3 libkrb5support0 libksba8 liblday-2.5-0 libldap-common libldb2 libllvm13 liblouis
liblouis20 liblsan0 libmagic-mgc libmagic1 libmbim-glib4 libmbim-proxy libmm-glib0 libmount1 libmpg123-0 libmutter-10-0 libnautilus-extension1a libncurses6 libncursesw6 libnd0 libnetplan0

Experiment No. 2: File System Exploration
Code:

```
#!/bin/bash
# Ensure the script is run as a regular user
if [ "$EUID" -eq 0 ]; then
echo "Please do not run as root."
exit 1
fi
# Function to display menu options
function show_menu() {
echo "Filesystem Explorer Options:"
echo "1. List files in the current directory"
echo "2. Display the size of files in the current directory"
echo "3. Count the number of files in a directory"
echo "4. Search for files by name"
echo "5. Search for files by type (e.g., .txt, .sh)"
echo "6. Exit"
echo -n "Enter your choice: "
}
# Function to list files
function list_files() {
echo "Files in the current directory:"
ls -l
}
# Function to display file sizes 8

function display_file_sizes() {
echo "File sizes in the current directory:"
du -h *
}
# Function to count files in a directory
function count_files() {
echo -n "Enter the directory path: "
read dir
if [ -d "$dir" ]; then
echo "Number of files in '$dir': $(find "$dir" -type f | wc -l)"
else
echo "Directory does not exist."
fi
}
# Function to search files by name
function search_by_name() {
echo -n "Enter the file name or pattern to search for: "
read pattern
echo "Searching for '$pattern'..."
```

```
find . -name "$pattern"
}
# Function to search files by type
function search_by_type() {
echo -n "Enter the file extension to search for (e.g., .txt, .sh): "
read ext
echo "Searching for files with extension '$ext'..."
find . -type f -name "*$ext"
}
# Main script loop
while true; do
show_menu
read choice
case $choice in
1) list_files ;;
2) display_file_sizes ;;
3) count_files ;;
4) search_by_name ;;
5) search_by_type ;;
6) echo "Exiting..."; break ;;
*) echo "Invalid choice. Please try again." ;;
esac
echo
done
```

Output:

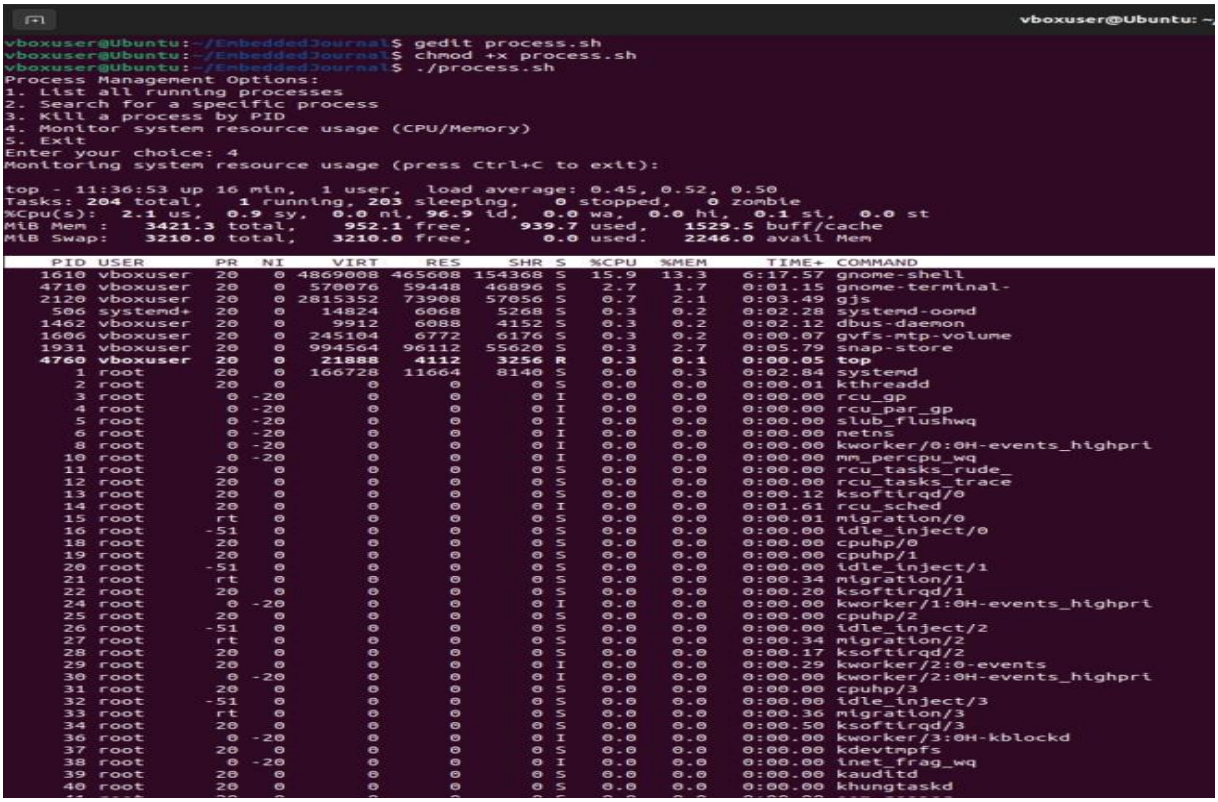


Experiment No. 3: Process Management
Code:

```
#!/bin/bash
# Ensure the script is run as a regular user
if [ "$EUID" -eq 0 ]; then
echo "Please do not run as root."
exit 1
fi
# Function to display menu options
function show_menu() {
echo "Process Management Options:"
echo "1. List all running processes"
echo "2. Search for a specific process"
echo "3. Kill a process by PID"
echo "4. Monitor system resource usage (CPU/Memory)"
echo "5. Exit"
echo -n "Enter your choice: "
}
# Function to list all running processes
function list_processes() {
echo "Listing all running processes:"
ps aux | less
}
# Function to search for a specific process
function search_process() {

echo -n "Enter the process name or keyword to search: "
read process_name
echo "Searching for processes matching '$process_name':"
ps aux | grep -i "$process_name" | grep -v "grep"
}
# Function to kill a process by PID
function kill_process() {
echo -n "Enter the PID of the process to kill: "
read pid
if kill -9 "$pid" 2>/dev/null; then
echo "Process with PID $pid has been terminated."
else
echo "Failed to terminate the process. Please check the PID."
fi
}
# Function to monitor system resource usage
function monitor_resources() {
echo "Monitoring system resource usage (press Ctrl+C to exit):"
top
```

```
}
# Main script loop
while true; do
show_menu
read choice

case $choice in
1) list_processes ;;
2) search_process ;;
3) kill_process ;;
4) monitor_resources ;;
5) echo "Exiting..."; break ;;
*) echo "Invalid choice. Please try again." ;;
esac
echo
done
```

Output:



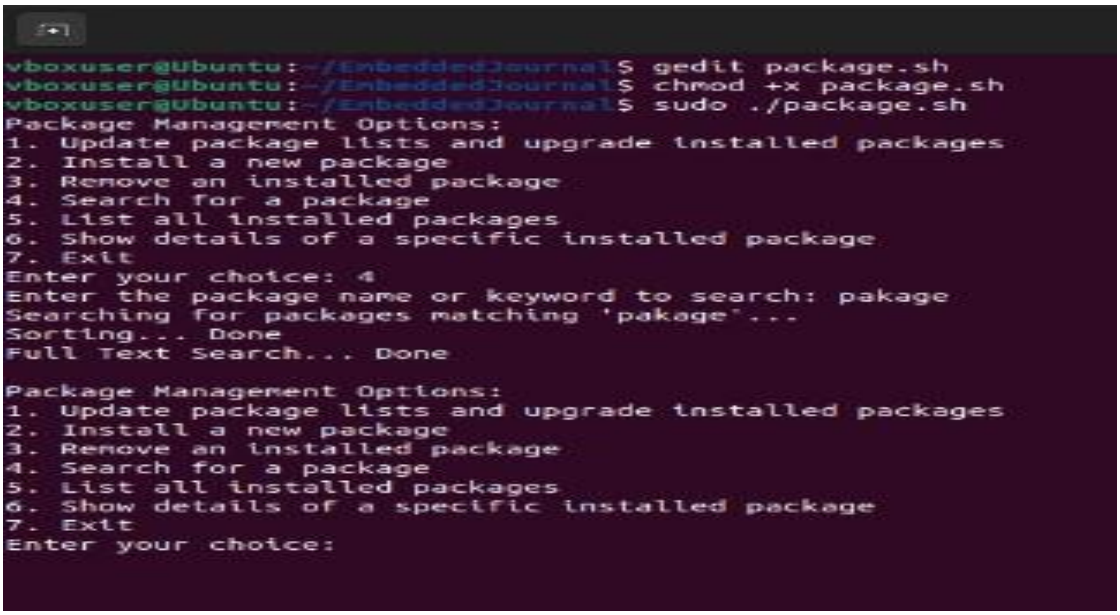
Experiment No. 4: Package Management
Code:

```
#!/bin/bash

# Ensure the script is run as root
if [ "$EUID" -ne 0 ]; then
    echo "Please run as root."
    exit 1
fi

# Function to display menu options
function show_menu() {
    echo "Package Management Options:"
    echo "1. Update package lists and upgrade installed packages"
    echo "2. Install a new package"
    echo "3. Remove an installed package"
    echo "4. Search for a package"
    echo "5. List all installed packages"
    echo "6. Show details of a specific installed package"
    echo "7. Exit"
    echo -n "Enter your choice: "
}

# Function to update and upgrade packages
function update_and_upgrade() {
    echo "Updating package lists and upgrading installed packages..."
    apt update && apt upgrade -y
    echo "Update and upgrade completed."
}
```

```
# Function to install a package
function install_package() {
    echo -n "Enter the package name to install: "
    read package_name
    if apt install -y "$package_name"; then
        echo "Package '$package_name' installed successfully."
    else
        echo "Failed to install package '$package_name'."
    fi
}

# Function to remove a package
function remove_package() {
    echo -n "Enter the package name to remove: "
    read package_name
    if apt remove -y "$package_name"; then
        echo "Package '$package_name' removed successfully."
    else
        echo "Failed to remove package '$package_name'."
    fi
}

# Function to search for a package
function search_package() {
    echo -n "Enter the package name or keyword to search: "
    read search_term
    echo "Searching for packages matching '$search_term'..."
    apt search "$search_term"
}
```

```
# Function to list all installed packages
function list_installed_packages() {
    echo "Listing all installed packages..."
    dpkg --get-selections | less
}
# Function to show details of a specific installed package
function package_details() {
    echo -n "Enter the package name to view details: "
    read package_name
    echo "Details for package '$package_name':"
    apt show "$package_name"
}

# Main script loop
while true; do
    show_menu
    read choice
    case $choice in
        1) update_and_upgrade ;;
        2) install_package ;;
        3) remove_package ;;
        4) search_package ;;
        5) list_installed_packages ;;
        6) package_details ;;
        7) echo "Exiting..."; break ;;
        *) echo "Invalid choice. Please try again." ;;
    esac
    echo
done
```

Output:

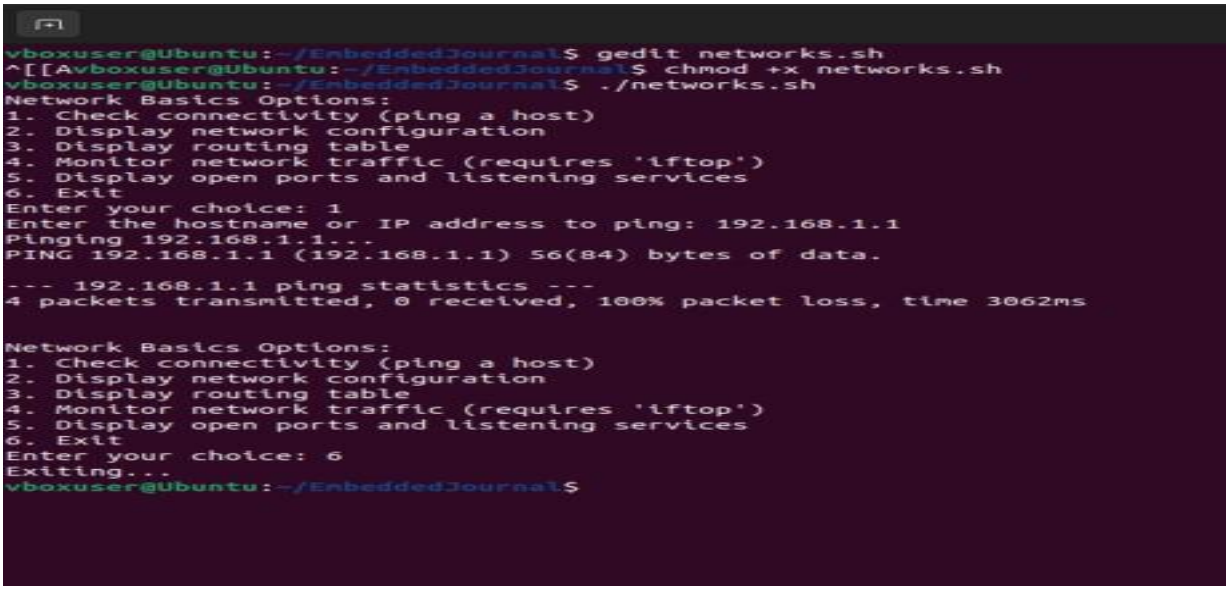

vboxuser@Ubuntu:~/EmbeddedJournal$ gedit package.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ chmod +x package.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ sudo ./package.sh
Package Management Options:
1. Update package lists and upgrade installed packages
2. Install a new package
3. Remove an installed package
4. Search for a package
5. List all installed packages
6. Show details of a specific installed package
7. Exit
Enter your choice: 4
Enter the package name or keyword to search: pakage
Searching for packages matching 'pakage'...
Sorting... Done
Full Text Search... Done
Package Management Options:
1. Update package lists and upgrade installed packages
2. Install a new package
3. Remove an installed package
4. Search for a package
5. List all installed packages
6. Show details of a specific installed package
7. Exit
Enter your choice:

Experiment No. 5: Network Basics
Code:

```
#!/bin/bash

# Ensure the script is run as a regular user
if [ "$EUID" -eq 0 ]; then
   echo "Please do not run as root."
   exit 1
fi

# Function to display menu options
function show_menu() {
   echo "Network Basics Options:"
   echo "1. Check connectivity (ping a host)"
   echo "2. Display network configuration"
   echo "3. Display routing table"
   echo "4. Monitor network traffic (requires 'iftop')"
   echo "5. Display open ports and listening services"
   echo "6. Exit"
   echo -n "Enter your choice: "
}

# Function to check connectivity
function check_connectivity() {
   echo -n "Enter the hostname or IP address to ping: "
   read host
   echo "Pinging $host..."
   ping -c 4 "$host"
}

# Function to display network configuration
function display_network_config() {
   echo "Network configuration:"
   ifconfig
}

# Function to display the routing table
function display_routing_table() {
   echo "Routing table:"
   netstat -rn
}

# Function to monitor network traffic
function monitor_traffic() {
   if ! command -v iftop &>/dev/null; then
```

```
        echo "'iftop' is not installed. Please install it using:"
        echo "sudo apt install iftop"
        return
    fi
    echo "Monitoring network traffic (press 'q' to exit):"
    sudo iftop
}

# Function to display open ports and listening services
function display_open_ports() {
    echo "Open ports and listening services:"
    sudo netstat -tuln
}
# Main script loop
while true; do
    show_menu
    read choice
    case $choice in
        1) check_connectivity ;;
        2) display_network_config ;;
        3) display_routing_table ;;
        4) monitor_traffic ;;
        5) display_open_ports ;;
        6) echo "Exiting..."; break ;;
        *) echo "Invalid choice. Please try again." ;;
    esac
    echo
done
```

Output:

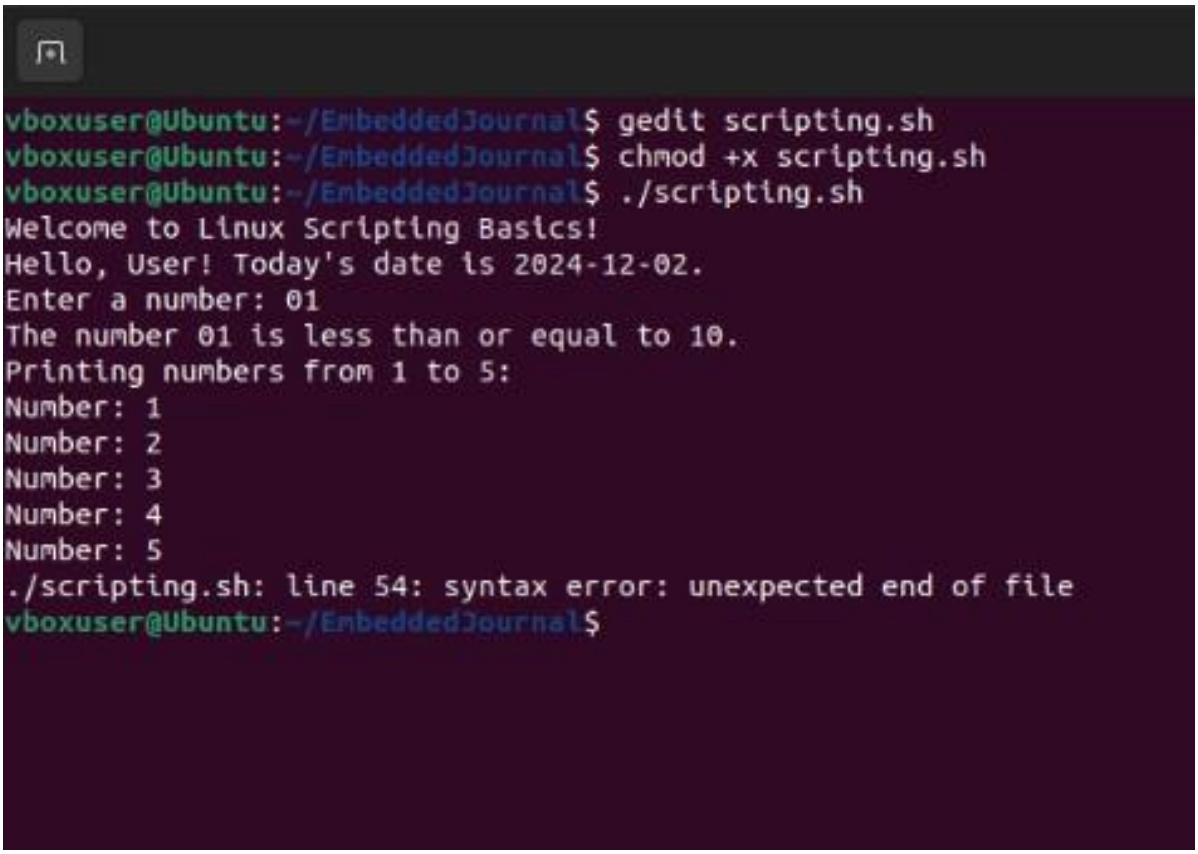


Experiment No. 6: Basic Scripting
Code:

```
#!/bin/bash
# Welcome message
echo "Welcome to Linux Scripting Basics!"
# Variables
name="User"
date=$(date "+%Y-%m-%d")
echo "Hello, $name! Today's date is $date."
# Conditional Statements
echo -n "Enter a number: "
read number
if [ $number -gt 10 ]; then
echo "The number $number is greater than 10."
else
echo "The number $number is less than or equal to 10."
fi
# Loops
echo "Printing numbers from 1 to 5:"
for i in {1..5}; do
echo "Number: $i"
done
# Functions
function greet_user() {
echo "Welcome, $1! Enjoy exploring Linux scripting."
```

```
echo -n "Enter your name: "
read user_name
greet_user "$user_name"
# File Handling
echo "Creating a sample file..."
echo "This is a sample file created by $user_name on $date." > sample.txt
echo "File 'sample.txt' created successfully."
# Display File Content
echo "Displaying the contents of 'sample.txt':"
cat sample.txt
# Basic Menu Example
while true; do
echo "Menu Options:"
echo "1. Show current directory"
echo "2. List files"
echo "3. Show disk usage"
echo "4. Exit"
echo -n "Enter your choice: "
read choice
case $choice in
1) echo "Current directory: $(pwd)" ;;
2) echo "Files in directory:"; ls -lh ;;
3) echo "Disk usage:"; df -h ;; 25
4) echo "Exiting..."; break ;;
*) echo "Invalid choice. Try again." ;;
esac
done
# Goodbye message
echo "Thank you for trying Linux scripting. Goodbye, $user_name!"
```

Output:

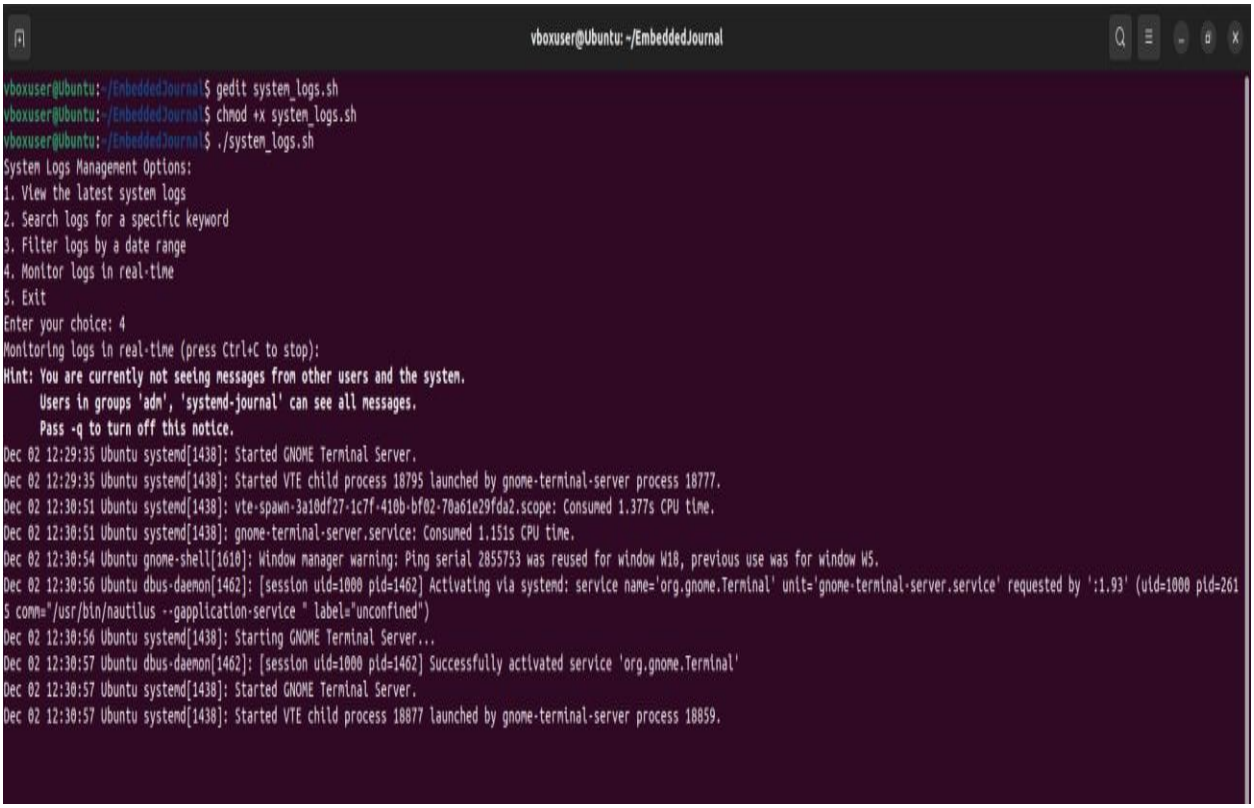

vboxuser@Ubuntu:~/EmbeddedJournal$ gedit scripting.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ chmod +x scripting.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ ./scripting.sh
Welcome to Linux Scripting Basics!
Hello, User! Today's date is 2024-12-02.
Enter a number: 01
The number 01 is less than or equal to 10.
Printing numbers from 1 to 5:
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
./scripting.sh: line 54: syntax error: unexpected end of file
vboxuser@Ubuntu:~/EmbeddedJournal$

Experiment No. 7: System Logs
Code:

```
#!/bin/bash
# Ensure the script is run as a regular user
if [ "$EUID" -eq 0 ]; then
echo "Please do not run as root for safety."
exit 1
fi
# Function to display menu options
function show_menu() {
echo "System Logs Management Options:"
echo "1. View the latest system logs"
echo "2. Search logs for a specific keyword"
echo "3. Filter logs by a date range"
echo "4. Monitor logs in real-time"
echo "5. Exit"
echo -n "Enter your choice: "
}
# Function to view the latest logs
function view_logs() {
echo "Displaying the latest 50 lines of the system log:"
journalctl -n 50
}
# Function to search logs for a keyword
function search_logs() { 27

echo -n "Enter the keyword to search for: "
read keyword
echo "Searching logs for '$keyword':"
journalctl | grep -i "$keyword"
```

```
}
# Function to filter logs by date range
function filter_logs_by_date() {
echo -n "Enter the start date (YYYY-MM-DD): "
read start_date
echo -n "Enter the end date (YYYY-MM-DD): "
read end_date
echo "Filtering logs from $start_date to $end_date:"
journalctl --since "$start_date" --until "$end_date"
}
# Function to monitor logs in real-time
function monitor_logs() {
echo "Monitoring logs in real-time (press Ctrl+C to stop):"
journalctl -f
}
# Main script loop
while true; do
show_menu
read choice
case $choice in 28

1) view_logs ;;
2) search_logs ;;
3) filter_logs_by_date ;;
4) monitor_logs ;;
5) echo "Exiting..."; break ;;
*) echo "Invalid choice. Please try again." ;;
esac
echo
done
```

Output:


vboxuser@Ubuntu: ~/EmbeddedJournal
vboxuser@Ubuntu:~/EmbeddedJournal$ gedit system_logs.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ chmod +x system_logs.sh
vboxuser@Ubuntu:~/EmbeddedJournal$ ./system_logs.sh
System Logs Management Options:
1. View the latest system logs
2. Search logs for a specific keyword
3. Filter logs by a date range
4. Monitor logs in real-time
5. Exit
Enter your choice: 4
Monitoring logs in real-time (press Ctrl+C to stop):
Hint: You are currently not seeing messages from other users and the system.
Users in groups 'adm', 'systemd-journal' can see all messages.
Pass -q to turn off this notice.
Dec 02 12:29:35 Ubuntu systemd[1438]: Started GNOME Terminal Server.
Dec 02 12:29:35 Ubuntu systemd[1438]: Started VTE child process 18795 launched by gnome-terminal-server process 18777.
Dec 02 12:30:51 Ubuntu systemd[1438]: vte-spawn-3a10df27-1c7f-410b-bf02-70a61e29fda2.scope: Consumed 1.377s CPU time.
Dec 02 12:30:51 Ubuntu systemd[1438]: gnome-terminal-server.service: Consumed 1.151s CPU time.
Dec 02 12:30:54 Ubuntu gnome-shell[1610]: Window manager warning: Ping serial 2855753 was reused for window W18, previous use was for window W5.
Dec 02 12:30:56 Ubuntu dbus-daemon[1462]: [session uid=1000 pid=1462] Activating via systemd: service name='org.gnome.Terminal' unit='gnome-terminal-server.service' requested by ':1.93' (uid=1000 pid=261
5 comm="/usr/bin/nautilus --gapplication-service " label="unconfined")
Dec 02 12:30:56 Ubuntu systemd[1438]: Starting GNOME Terminal Server...
Dec 02 12:30:57 Ubuntu dbus-daemon[1462]: [session uid=1000 pid=1462] Successfully activated service 'org.gnome.Terminal'
Dec 02 12:30:57 Ubuntu systemd[1438]: Started GNOME Terminal Server.
Dec 02 12:30:57 Ubuntu systemd[1438]: Started VTE child process 18877 launched by gnome-terminal-server process 18859.