XENON STACK

REPORT 1 : LINUX

SUBMITTED BY :

Puneet Katiyar(1221284)

puneetkatiyar400@gmail.com

SECTION A

Question 1 :

Step 1: Create the Manual Page

Use a text editor to create a manual page for your command. Let's assume you are using `nano` for simplicity:

```
sudo nano /usr/share/man/man1/internsctl.1
```

By using groff

```
.TH INTERNSCTL 1 "December 2023" "v0.1.0" "internsctl manual"

.SH NAME

internsctl \- Custom command for operations

.SH SYNOPSIS

.B internsctl

[\fIOPTIONS\fR] [\fIARGUMENTS\fR]


.SH DESCRIPTION

This command is a custom tool designed for specific operations. It provides functionality for XYZ.


.SH OPTIONS

.TP
```

```
.B \-h, \-\-help

Display help message and exit.


.TP

.B \-v, \-\-version

Display version information and exit.


.SH EXAMPLES

Display help:

.PP

.B internsctl \-h


.SH AUTHOR

Written by Your Name.
```

Step 2: Format the Manual Page

If you are using `groff` to format the manual page:

```
sudo groff -man -Tascii /usr/share/man/man1/internsctl.1 > /tmp/internsctl.man
```

Step 3: Update the Manual Page Index

Update the manual page index using `mandb`:

```
sudo mandb
```

Step 4: View the Manual Page

Now, users can view the manual page using the `man` command:

```
man internsctl
```

Question 2 :

WE will write vim and file name and inside this file we will write this code

```
#!/bin/bash

if [ "$1" == "--help" ]; then
    echo "Usage: internsctl COMMAND [OPTIONS]"
    echo
    echo "Commands:"
    echo "  cpu getinfo   Display CPU information"
    echo
    echo "Options:"
    echo "  --help        Display this help message"
    exit 0
fi
if [ "$1" == "cpu" ] && [ "$2" == "getinfo" ]; then
    lscpu
else
    echo "Invalid command. Run 'internsctl --help' for usage information."
    exit 1
fi
```

by exiting the vim , Now on the terminal we will write the command

```
chmod +x internsctl
sudo mv internsctl /usr/local/bin/
```

Question 3 :

WE will write vim and file name and inside this file we will write this code

```
#!/bin/bash

VERSION="v0.1.0"
```

```
if [ "$1" == "--version" ]; then
   echo "internsctl $VERSION"
   exit 0
fi

if [ "$1" == "--help" ]; then
   echo "Usage: internsctl COMMAND [OPTIONS]"
   echo
   echo "Commands:"
   echo "  cpu getinfo   Display CPU information"
   echo
   echo "Options:"
   echo "  --help       Display this help message"
   echo "  --version    Display version information"
   exit 0
fi

if [ "$1" == "cpu" ] && [ "$2" == "getinfo" ]; then
   lscpu
else
   echo "Invalid command. Run 'internsctl --help' for usage information."
   exit 1
fi
```

by exiting the vim , Now on the terminal we will write the command

```
chmod +x internsctl
sudo mv internsctl /usr/local/bin/
```

SECTION B :

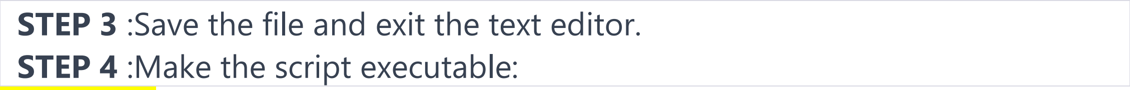
Part1 | Level Easy I want to get cpu information of my server through the following command

STEP 1 : Open a text editor and create a new file. You can use `nano` or `vim`:

```
nano internsctl
```

STEP 2 : Add the following content to the file:

```
#!/bin/bash
if [ "$1" == "cpu" ] && [ "$2" == "getinfo" ]; then
    lscpu
else
    echo "Invalid command. Usage: internsctl cpu getinfo"
    exit 1
fi
```



STEP 3 :Save the file and exit the text editor.
STEP 4 :Make the script executable:

```
chmod +x internsctl
sudo mv internsctl /usr/local/bin/
internsctl cpu getinfo
```

Part2 | Level Intermediate

```
#!/bin/bash
VERSION="v0.1.0"
if [ "$1" == "--version" ]; then
    echo "internsctl $VERSION"
    exit 0
fi
if [ "$1" == "--help" ]; then
    echo "Usage: internsctl COMMAND [OPTIONS]"
    echo
    echo "Commands:"
    echo "  cpu getinfo      Display CPU information"
    echo "  memory getinfo   Display memory information"
    echo "  user create      Create a new user"
    echo "  user list        List all regular users"
    echo "  user list --sudo-only   List users with sudo permissions"
    echo
    echo "Options:"
    echo "  --help          Display this help message"
    echo "  --version       Display version information"
    exit 0
fi

if [ "$1" == "cpu" ] && [ "$2" == "getinfo" ]; then
    lscpu
elif [ "$1" == "memory" ] && [ "$2" == "getinfo" ]; then
    free
elif [ "$1" == "user" ] && [ "$2" == "create" ]; then
    if [ -z "$3" ]; then
        echo "Usage: internsctl user create <username>"
        exit 1
    fi
    sudo useradd -m -s /bin/bash "$3"
    echo "User '$3' created successfully."
elif [ "$1" == "user" ] && [ "$2" == "list" ]; then
    if [ "$3" == "--sudo-only" ]; then
        getent passwd {1000..60000} | cut -d: -f1,3,4 | awk -F: '$2 >= 1000 { print $1 }' | xargs groups | awk -F: '{if ($2 ~ /sudo/) print $1}'
    else
        getent passwd | cut -d: -f1
    fi
else
    echo "Invalid command. Run 'internsctl --help' for usage information."
    exit 1
fi
```

With this script, users can execute the following commands:

IN BASH WE WILL EXECUTE THE FOLLOWING COMMANDS:

```
# For CPU information
internsctl cpu getinfo

# For Memory information
internsctl memory getinfo

# To create a new user
internsctl user create <username>

# To list all regular users
internsctl user list

# To list users with sudo permissions
internsctl user list --sudo-only
```

Part3 | Advanced Level

```
#!/bin/bash

VERSION="v0.1.0"

if [ "$1" == "--version" ]; then
  echo "internsctl $VERSION"
  exit 0
fi

if [ "$1" == "--help" ]; then
```

```
    echo "Usage: internsctl COMMAND [OPTIONS]"
    echo
    echo "Commands:"
    echo "  cpu getinfo      Display CPU information"
    echo "  memory getinfo   Display memory information"
    echo "  user create      Create a new user"
    echo "  user list        List all regular users"
    echo "  user list --sudo-only   List users with sudo permissions"
    echo "  file getinfo     Display file information"
    echo
    echo "Options:"
    echo "  --size, -s       Print file size"
    echo "  --permissions, -p Print file permissions"
    echo "  --owner, -o      Print file owner"
    echo "  --last-modified, -m Print last modified time"
    echo
    echo "  --help           Display this help message"
    echo "  --version        Display version information"
    exit 0
fi

if [ "$1" == "cpu" ] && [ "$2" == "getinfo" ]; then
    lscpu
elif [ "$1" == "memory" ] && [ "$2" == "getinfo" ]; then
    free
elif [ "$1" == "user" ] && [ "$2" == "create" ]; then
    if [ -z "$3" ]; then
        echo "Usage: internsctl user create <username>"
        exit 1
    fi
    sudo useradd -m -s /bin/bash "$3"
    echo "User '$3' created successfully."
elif [ "$1" == "user" ] && [ "$2" == "list" ]; then
    if [ "$3" == "--sudo-only" ]; then
        getent passwd {1000..60000} | cut -d: -f1,3,4 | awk -F: '$2 >= 1000 { print $1 }' | xargs groups | awk -F: '{if ($2 ~ /sudo/) print $1}'
    else
        getent passwd | cut -d: -f1
    fi
elif [ "$1" == "file" ] && [ "$2" == "getinfo" ]; then
    if [ -z "$3" ]; then
        echo "Usage: internsctl file getinfo [options] <file-name>"
        exit 1
    fi

    file_name="$3"
    shift 3

    size=false
    permissions=false
    owner=false
    last_modified=false

    while [ "$#" -gt 0 ]; do
        case "$1" in
            --size|-s)
                size=true
                ;;
            --permissions|-p)
                permissions=true
                ;;
            --owner|-o)
                owner=true
                ;;
            --last-modified|-m)
                last_modified=true
                ;;
            *)
                echo "Invalid option: $1"
                exit 1
                ;;
        esac
        shift
    done

    if [ ! -e "$file_name" ]; then
        echo "File '$file_name' not found."
        exit 1
    fi

    if [ "$size" == true ]; then
        stat -c "%s" "$file_name"
    fi

    if [ "$permissions" == true ]; then
        stat -c "%A" "$file_name"
    fi
```

```
    if [ "$owner" == true ]; then
        stat -c "%U" "$file_name"
    fi

    if [ "$last_modified" == true ]; then
        stat -c "%y" "$file_name"
    fi

    if [ "$size" == false ] && [ "$permissions" == false ] && [ "$owner" == false ] && [ "$last_modified" == false ]; then
        echo "No options provided. Use options like --size, --permissions, --owner, --last-modified."
        exit 1
    fi
else
    echo "Invalid command. Run 'internsctl --help' for usage information."
    exit 1
fi
```

With this script, users can execute the following commands:

IN BASH WE WILL EXECUTE THE FOLLOWING COMMANDS:

```
# For CPU information
internsctl cpu getinfo

# For Memory information
internsctl memory getinfo

# To create a new user
internsctl user create <username>

# To list all regular users
```

```
internsctl user list


# To list users with sudo permissions

internsctl user list --sudo-only


# For file information with options

internsctl file getinfo <file-name>

# For specific information

internsctl file getinfo --size/--permissions/--owner/--last-modified <file-name>
```

THANK YOU

I LOOK FORWARD TO HEARING FROM YOU VERY SOON