**Yash Chaudhary**

**(20BCS8245)**

**\*\* Solutions written in red color, because I am unable to setup Linux Runtime Environment.**

General instructions

1. The approach of solving the Problem solely depends on the Candidate
2. Make sure to have Draw.io diagrams for the workﬂowns and application architecture
3. Every conﬁguration, code written should be pushed on git (Private Repo)
4. Your are not permitted to share the doc with anyone, even with your colleagues

**Scenario** There is a customer who came to you with a problem to have a custom linux command for his operations. Your task is to understand the problem and create a linux command via bash script as per the instructions.

Command name - internsctl

Command version - v0.1.0

# Section A

1. I want a manual page of command so that I can see the full documentation of the command.

For example if you execute the command

man ls

as output we get the doc and usage guidelines. Similarly if I execute man internsctl I want to see the manual of my command.

1. Each linux command has an option --help which helps the end user to understand the use cases via examples. Similarly if I execute internsctl --help it should provide me the necessary help
2. I want to see version of my command by executing

internsctl --version

# Section B

I want to execute the following command for -

# Part1 | Level Easy

I want to get cpu information of my server through the following command:

$ internsctl cpu getinfo

Expected Output -

I want similar output as we get from lscpu command

**COMMAND:** alias internsctl\_cpu\_getinfo=”lscpu”

---

I want to get memory information of my server through the following command:

$ internsctl memory getinfo

Expected Output

I want similar output as we get from free command

**COMMAND:** alias internsctl\_memory\_getinfo=”free”

# Part2 | Level Intermediate

I want to create a new user on my server through the following command:

$ internsctl user create <username>

**COMMAND:**

nano internsctl

if [ "$1" = "cpu" ] && [ "$2" = "getinfo" ]; then

lscpu

elif [ "$1" = "user" ] && [ "$2" = "create" ]; then

if [ -z "$3" ]; then

echo "Error: Please provide a username."

echo "Usage: internsctl user create <username>"

else

sudo adduser "$3"

fi

else

echo "Usage: internsctl cpu getinfo"

echo " internsctl user create <username>"

fi

chmod +x internsctl

**Note** - above command should create user who can login to linux system and access his home directory

---

I want to list all the regular users present on my server through the following command:

$ internsctl user list

**COMMAND:**

nano internsctl

if [ "$1" = "cpu" ] && [ "$2" = "getinfo" ]; then

lscpu

elif [ "$1" = "user" ]; then

if [ "$2" = "create" ]; then

if [ -z "$3" ]; then

echo "Error: Please provide a username."

echo "Usage: internsctl user create <username>"

else

sudo adduser "$3"

fi

elif [ "$2" = "list" ]; then

cut -d: -f1 /etc/passwd

else

echo "Usage: internsctl user create <username>"

echo " internsctl user list"

fi

else

echo "Usage: internsctl cpu getinfo"

echo " internsctl user create <username>"

echo " internsctl user list"

fi

chmod +x internsctl

---

If want to list all the users with sudo permissions on my server through the following command:

$ internsctl user list --sudo-only

**COMMAND:**

nano internsctl

#!/bin/bash

if [ "$1" = "cpu" ] && [ "$2" = "getinfo" ]; then

lscpu

elif [ "$1" = "user" ]; then

if [ "$2" = "create" ]; then

if [ -z "$3" ]; then

echo "Error: Please provide a username."

echo "Usage: internsctl user create <username>"

else

sudo adduser "$3"

fi

elif [ "$2" = "list" ]; then

if [ "$3" = "--sudo-only" ]; then

getent group sudo | cut -d: -f4 | tr ',' '\n'

else

cut -d: -f1 /etc/passwd

fi

else

echo "Usage: internsctl user create <username>"

echo " internsctl user list [--sudo-only]"

fi

else

echo "Usage: internsctl cpu getinfo"

echo " internsctl user create <username>"

echo " internsctl user list [--sudo-only]"

fi

chmod +x internsctl

# Part3 | Advanced Level

By executing below command I want to get some information about a **file**

$ internsctl file getinfo <file-name>

Expected Output [make sure to have the output in following format only]

xenonstack@xsd-034:~$ internsctl file getinfo hello.txt File: hellot.txt

Access: -rw-r--r--

Size(B): 5448

Owner: xenonstack

Modify: 2020-10-07 20:34:44.616123431 +0530

**COMMAND:**

if [ "$1" = "file" ] && [ "$2" = "getinfo" ] && [ -n "$3" ]; then

if [ -e "$3" ]; then

filename="$3"

file\_info=$(stat --format="File: %n%nAccess: %a%nSize(B): %s%nOwner: %U" "$filename")

echo -e "$file\_info"

else

echo "File not found."

fi

else

echo "Usage: internsctl file getinfo <file-name>"

fi

In case I want only specific information then I must have a provision to use **options**

$ internsctl file getinfo [options] <file-name>

--size, -s to print size

--permissions, -p print file permissions

--owner, o print file owner

--last-modified, m

Expected Output with options

If I want to obtain the size of the specified file only, I should be able to use the following command:

xenonstack@xsd-034:~$ internsctl file getinfo --size hello.txt 5448

**COMMAND:**

if [ "$1" = "file" ] && [ "$2" = "getinfo" ]; then

shift 2 # Remove the first two arguments: "file" and "getinfo"

# Default values for options

size=false

permissions=false

owner=false

last\_modified=false

# Parse options

while [[ $# -gt 0 ]]; do

case $1 in

-s|--size)

size=true

;;

-p|--permissions)

permissions=true

;;

-o|--owner)

owner=true

;;

-m|--last-modified)

last\_modified=true

;;

\*)

filename="$1"

;;

esac

shift

done

if [ -z "$filename" ]; then

echo "Error: Please provide a filename."

echo "Usage: internsctl file getinfo [options] <file-name>"

exit 1

fi

if [ ! -e "$filename" ]; then

echo "File not found."

exit 1

fi

# Fetch specific information based on options

if [ "$size" = true ]; then

stat --format="%s" "$filename"

fi

if [ "$permissions" = true ]; then

stat --format="%A" "$filename"

fi

if [ "$owner" = true ]; then

stat --format="%U" "$filename"

fi

if [ "$last\_modified" = true ]; then

stat --format="%y" "$filename"

fi

else

echo "Usage: internsctl file getinfo [options] <file-name>"

echo "Options:"

echo " --size, -s Print size"

echo " --permissions, -p Print file permissions"

echo " --owner, -o Print file owner"

echo " --last-modified, -m Print last modified date"

fi

If I want to obtain the permissions of the specified file only, I should be able to use the following command:

xenonstack@xsd-034:~$ internsctl file getinfo --permissions hello.txt

-rw-r--r—

**COMMAND:**

if [ "$1" = "file" ] && [ "$2" = "getinfo" ]; then

shift 2 # Remove the first two arguments: "file" and "getinfo"

# Default values for options

size=false

permissions=false

owner=false

last\_modified=false

# Parse options

while [[ $# -gt 0 ]]; do

case $1 in

-s|--size)

size=true

;;

-p|--permissions)

permissions=true

;;

-o|--owner)

owner=true

;;

-m|--last-modified)

last\_modified=true

;;

\*)

filename="$1"

;;

esac

shift

done

if [ -z "$filename" ]; then

echo "Error: Please provide a filename."

echo "Usage: internsctl file getinfo [options] <file-name>"

exit 1

fi

if [ ! -e "$filename" ]; then

echo "File not found."

exit 1

fi

# Fetch specific information based on options

if [ "$size" = true ]; then

stat --format="%s" "$filename"

fi

if [ "$permissions" = true ]; then

stat --format="%A" "$filename"

fi

if [ "$owner" = true ]; then

stat --format="%U" "$filename"

fi

if [ "$last\_modified" = true ]; then

stat --format="%y" "$filename"

fi

else

echo "Usage: internsctl file getinfo [options] <file-name>"

echo "Options:"

echo " --size, -s Print size"

echo " --permissions, -p Print file permissions"

echo " --owner, -o Print file owner"

echo " --last-modified, -m Print last modified date"

fi

If I want to obtain the owner of the specified file only, I should be able to use the following command:

xenonstack@xsd-034:~$ internsctl file getinfo --owner hello.txt xenonstack

**COMMAND:**

if [ "$1" = "file" ] && [ "$2" = "getinfo" ]; then

shift 2 # Remove the first two arguments: "file" and "getinfo"

# Default values for options

size=false

permissions=false

owner=false

last\_modified=false

# Parse options

while [[ $# -gt 0 ]]; do

case $1 in

-s|--size)

size=true

;;

-p|--permissions)

permissions=true

;;

-o|--owner)

owner=true

;;

-m|--last-modified)

last\_modified=true

;;

\*)

filename="$1"

;;

esac

shift

done

if [ -z "$filename" ]; then

echo "Error: Please provide a filename."

echo "Usage: internsctl file getinfo [options] <file-name>"

exit 1

fi

if [ ! -e "$filename" ]; then

echo "File not found."

exit 1

fi

# Fetch specific information based on options

if [ "$size" = true ]; then

stat --format="%s" "$filename"

fi

if [ "$permissions" = true ]; then

stat --format="%A" "$filename"

fi

if [ "$owner" = true ]; then

file\_owner=$(stat --format="%U" "$filename")

if [ "$file\_owner" = "$3" ]; then

echo "$file\_owner"

else

echo "Owner does not match specified user."

fi

fi

if [ "$last\_modified" = true ]; then

stat --format="%y" "$filename"

fi

else

echo "Usage: internsctl file getinfo [options] <file-name>"

echo "Options:"

echo " --size, -s Print size"

echo " --permissions, -p Print file permissions"

echo " --owner, -o Print file owner"

echo " --last-modified, -m Print last modified date"

fi

If I want to obtain the last modified time of the specified file only, I should be able to use the following command:

xenonstack@xsd-034:~$ internsctl file getinfo --last-modified hello.txt 2020-10-07 20:34:44.616123431 +0530

**COMMAND:**

if [ "$1" = "file" ] && [ "$2" = "getinfo" ]; then

shift 2 # Remove the first two arguments: "file" and "getinfo"

# Default values for options

size=false

permissions=false

owner=false

last\_modified=false

timestamp=""

# Parse options

while [[ $# -gt 0 ]]; do

case $1 in

-s|--size)

size=true

;;

-p|--permissions)

permissions=true

;;

-o|--owner)

owner=true

;;

-m|--last-modified)

last\_modified=true

;;

\*)

if [ -z "$filename" ]; then

filename="$1"

else

timestamp="$1"

fi

;;

esac

shift

done

if [ -z "$filename" ]; then

echo "Error: Please provide a filename."

echo "Usage: internsctl file getinfo [options] <file-name> [timestamp]"

exit 1

fi

if [ ! -e "$filename" ]; then

echo "File not found."

exit 1

fi

# Fetch specific information based on options

if [ "$size" = true ]; then

stat --format="%s" "$filename"

fi

if [ "$permissions" = true ]; then

stat --format="%A" "$filename"

fi

if [ "$owner" = true ]; then

stat --format="%U" "$filename"

fi

if [ "$last\_modified" = true ]; then

last\_mod=$(stat --format="%y" "$filename" | cut -d ' ' -f1-2)

if [ "$last\_mod" = "$timestamp" ]; then

echo "$last\_mod"

else

echo "Last modified time does not match specified timestamp."

fi

fi

else

echo "Usage: internsctl file getinfo [options] <file-name> [timestamp]"

echo "Options:"

echo " --size, -s Print size"

echo " --permissions, -p Print file permissions"

echo " --owner, -o Print file owner"

echo " --last-modified, -m Print last modified date"

fi