Step 1

Create the Script File:

Open a terminal in your Linux environment.
Use a text editor (like nano or vim) to create a new file. For example, you can type nano internsctl to open nano editor.
Initial Script Content:

Add the following lines to the file. This is a basic structure that allows the script to handle different **commands**:

```
#!/bin/bash

# Function to display help
help() {
   echo "Usage: internsctl [command] [options]"
   # You will expand this with more information later
}

# Parsing command line arguments
case "$1" in
   --help)
      help
      ;;
   --version)
      echo "internsctl version v0.1.0"
      ;;
   *)
      echo "Invalid command. Use --help for usage."
      ;;
esac
```

Save and Exit:

If you are using nano, you can save and exit by pressing Ctrl+X, then Y to confirm, and Enter to save.
Make the Script Executable:

In the terminal, run the command chmod +x internsctl to make the script executable.
Test the Script:

You can test the script by running ./internsctl --help and ./internsctl --version

Implementing CPU Information Command

Command: internsctl cpu getinfo
Expected Behavior: Display CPU information, similar to lscpu.
Add the following code to your internsctl script:

```
case "$1" in
   cpu)
      if [[ "$2" == "getinfo" ]]; then
         lscpu
      else
         echo "Invalid command for cpu. Use --help for usage."
      fi
      ;;
   # ... other cases ...
esac
```

Implementing Memory Information Command

Command: internsctl memory getinfo
Expected Behavior: Display memory information, similar to free.
Add this code to your internsctl script:

```
case "$1" in
   memory)
      if [[ "$2" == "getinfo" ]]; then
         free
      else
         echo "Invalid command for memory. Use --help for usage."
      fi
      ;;
   # ... other cases ...
esac
```

Testing

After adding these cases, test the functionality by running:

./internsctl cpu getinfo
./internsctl memory getinfo

Step 2

Implementing User Management Commands

Create User Command: **internsctl** user create <username>
List Users Command: **internsctl** user list
List Users with Sudo Command: **internsctl user list --sudo-only**
1. Create User Command
This command will create a new user in the Linux system. Add the following to your **internsctl** script:

```
case "$1" in
   user)
      case "$2" in
         create)
            if [ -z "$3" ]; then
               echo "Please specify a username. Usage: internsctl user create <username>"
            else
               sudo useradd -m "$3" && echo "User $3 created successfully."
            fi
            ;;
         # ... other user sub-commands ...
      esac
      ;;
   # ... other main commands ...
esac
```

2. List Users Command

This command lists all regular users on the system. You can use the awk command to filter out system users. Add this to your script:

```
case "$2" in
   list)
      if [ "$3" == "--sudo-only" ]; then
         # Implementation for sudo-only will be added next
         :
      else
         awk -F':' '$3 >= 1000 && $3 != 65534 {print $1}' /etc/passwd
      fi
      ;;
   # ... other user sub-commands ...
esac
```

3. List Users with Sudo Command

This command lists users with sudo permissions. Append this to the user case:

```
case "$3" in
   --sudo-only)
      awk -F':' '/sudo/ {print $4}' /etc/group | tr ',' '\n'
      ;;
   # ... other options for user list command ...
esac
```

Testing

After adding these functionalities, test each one:

./**internsctl** user create testuser (to create a new user named 'testuser').
./**internsctl** user list (to list all regular users).
./**internsctl** user list --sudo-only (to list users with sudo permissions).

STEP 3

Implementing File Information Command

Command: **internsctl** file getinfo <file-name>
Options: --size (-s), --permissions (-p), --owner (-o), --last-modified (-m)
Here's how to implement it:

```
case "$1" in
  file)
    if [ "$2" == "getinfo" ]; then
      if [ -z "$3" ]; then
        echo "Please specify a file name. Usage: internsctl file getinfo <file-name>"
        exit 1
      fi
      case "$3" in
        --size|-s)
          stat -c %s "$4"
          ;;
        --permissions|-p)
          stat -c %A "$4"
          ;;
        --owner|-o)
          stat -c %U "$4"
          ;;
        --last-modified|-m)
          stat -c %y "$4"
          ;;
        *)
          echo "File: $3"
          echo "Access: $(stat -c %A $3)"
          echo "Size(B): $(stat -c %s $3)"
          echo "Owner: $(stat -c %U $3)"
          echo "Modify: $(stat -c %y $3)"
          ;;
      esac
    fi
    ;;
  # ... other main commands ...
esac
```

./internsctl file getinfo <file-name> to display all information.
./internsctl file getinfo --size <file-name> to display file size.
./internsctl file getinfo --permissions <file-name> for permissions.

./internsctl file getinfo --owner <file-name> for the file owner.
./internsctl file getinfo --last-modified <file-name> for the last modified time.