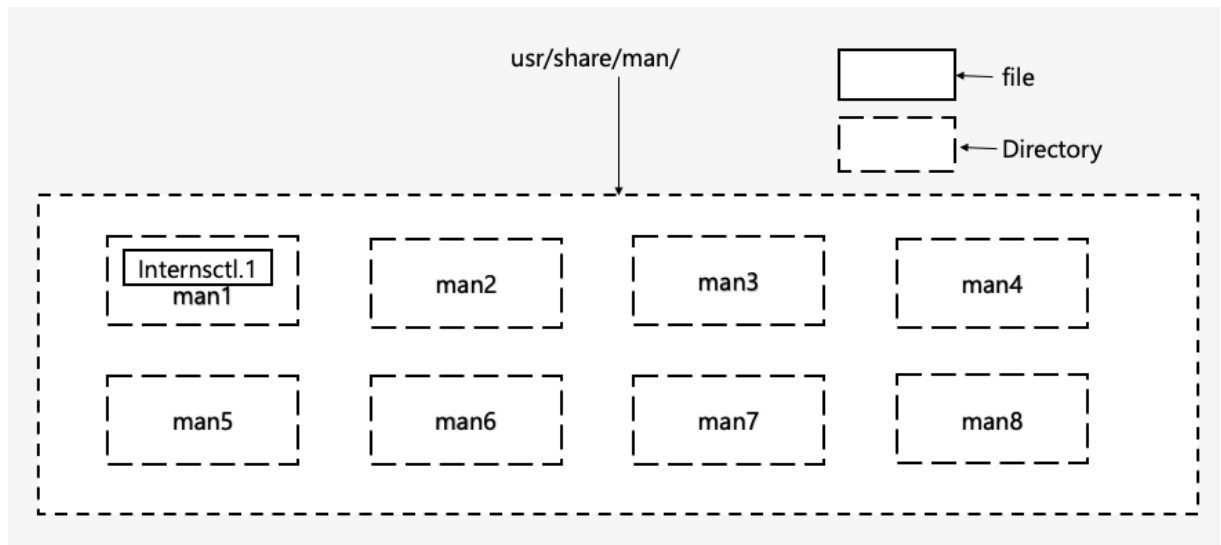
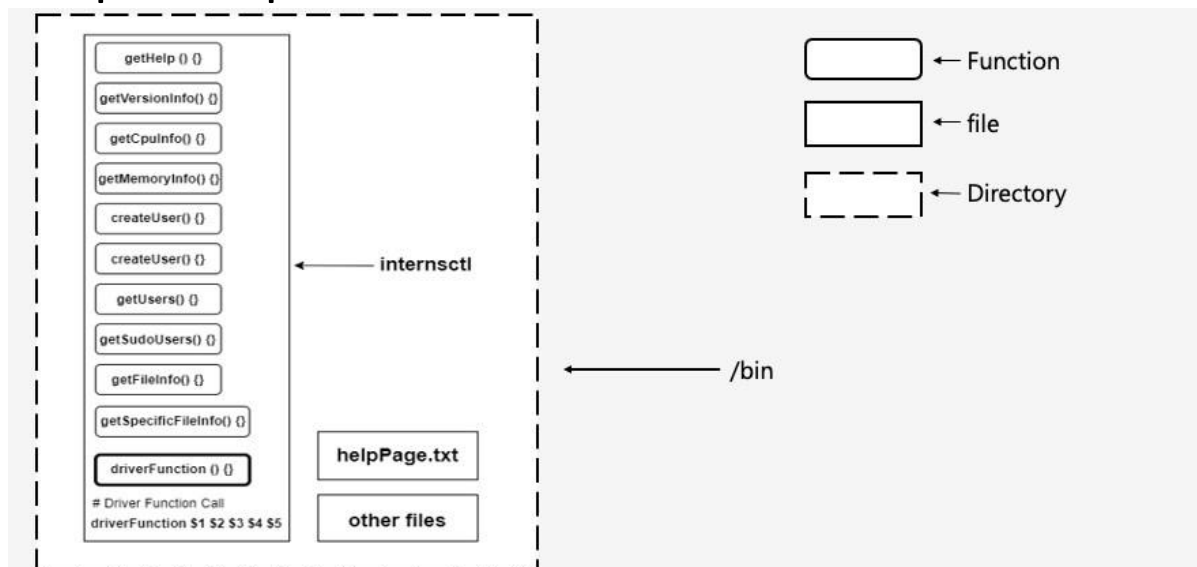


Workflow



Script Setup



Section A

1. Creating manual (man) page**

- **Step 1 :**

- Login as a root user by running the command `sudo -i` (If it asks for the administrative password, Enter it).
- Now using `cd` command move into to the standrad location in filesystem : `/usr/share/man`, where manual pages of all the commands are normally stored in **nroff(1)** format.
- Then run `ls` command to list all the directories in that location. Here in this location, each man page is categorized in a specific section (directory), different directories (e.g., `man1`, `man2`, `man3`...) store man pages for different category of commands. See below -

```
man1 - User Commands
man2 - System Calls
man3 - C Library Functions
man4 - Devices and Special Files
man5 - File Formats and Conventions
man6 - Games et. al
man7 - Miscellaneous
man8 - System Administration tools and Daemons
```

Now since **internsctl** is a **user command**, we will create and store the manual page file in `/man1` directory.

- **Step 2 :**

- From the current directory, navigate to `/man1` directory using `cd man1` command.
- Create the source file of the man page using the command `touch` followed by `<File_Name>.<Section_Index>`.

File_Name : The command whose manual page to be created.

Section_Index : For `man1` - it'll be **1**, For `man2` - it'll be **2**, and so on.

In this case it will be : **`touch internsctl.1`**

- **Step 3 :**

- Now run `nano internsctl.1` to edit the source file in nano text editor. Copy and paste the following script into the source file or write it from yourself and save it.
- `.\" Manual (man) page of internsctl`
- `.TH internsctl 1 "10 July 2023" "0.1.0" "Custom Command"`
- `.SH NAME`
- `internsctl`
- `.SH SYNOPSIS`

```

o internsctl cpu getinfo |
o .brinternsctl memory getinfo |
o .brinternsctl user create <username> |
o internsctl user list |
o internsctl user list --sudo-only |
o internsctl file getinfo <file-name> |
o internsctl file getinfo [options] <file-name>
o .SH DESCRIPTION
o Display cpu and memory information, create new user, list all
  users, list all users with sudo permissions, get file
  information, get specific information of file.
o .SH OPTIONS
o .TP
o .BR \-\-size ", " \-s                print " " file " "
  size
o .TP
o .BR \-\-permissions ", " \-p          print " " file " "
  permissions
o .TP
o .BR \-\-owner ", " \-o                print " " file " "
  owner
o .TP
o .BR \-\-last-modified ", " \-m        print " " last " "
  modified " " date " " and " " time " " of " " the " " file
o .SH BUGS
o No known bugs.
o .SH AUTHOR
o Shikha Rajput

```

• **Step 4 :**

- o Run `man internsctl` from terminal to check the manual page of the `internsctl`.
- o [2. Creating function to display the help text through the command `internsctl --help`](#)
- Create a file `internsctl` in `/bin` directory.
- Copy and paste the following code into that file and save it.
- `getHelp () {`
- `cat /usr/bin/helpPage.txt`
- `}`
- Now create another file `helpPage.txt` in the same directory and copy and paste the following help text into that file and save it.
- Usage: '`internsctl cpu getinfo`' -> Get cpu information of the local server.
- '`internsctl memory getinfo`' -> Get memory information of the local server.
- '`internsctl user create <username>`' -> Create a new user on the local server.
- '`internsctl user list`' -> List all the regular users present on the local server.
- '`internsctl user list --sudo-only`' -> List all the users with sudo permissions on the local server.

- `'internsctl file getinfo <file-name>' ->` Get information about a file.
- `'internsctl file getinfo [options] <file-name>' ->` Get specific information about a file.
-
- Mandatory arguments to long options are mandatory for short options too.
- `--size, -s` print file size
- `--permissions, -p` print file permissions
- `--owner, -o` print file owner
- `--last-modified, -m` print last modified date and time of the file
-
- `--help` display help text and exit
- `--version` output version information and exit
-
- Exit status:
- 0 if OK,
- 1 if minor problems (e.g., cannot access subdirectory),
- 2 if serious trouble (e.g., cannot access command-line argument).

[3. Creating function to display version of the command through `internsctl --version`](#)

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- `getVersionInfo () {`
- `echo "internsctl 0.1.0"`
- `echo "Copyright (C) 2023 XenonStack "`
- `}`

[Section B](#)

[Part 1 | Level Easy](#)

[1. Creating function to get cpu information of server through the command `internsctl cpu getinfo`](#)

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- `getCpuInfo () {`
- `lscpu`
- `}`
-

[2. Creating function to get memory information of server through the command `internsctl memory getinfo`](#)

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- `getMemoryInfo () {`
- `free`
- `}`

Part 2 | Level Intermediate

1. Creating function to create a new user on server through the command `internsctl user create <username>**`

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- ```
createUser () {
```
- ```
    sudo adduser $3
```
- ```
}
```

### 2. Creating function to list all the regular users present on the server through the command `internsctl user list**`

---

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- ```
getUsers () {
```
- ```
 cut -d: -f1 /etc/passwd
```
- ```
}
```

3. Creating function to list all the users with sudo permissions on the server through the command `internsctl user list --sudo-only**`

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- ```
getSudoUsers () {
```
- ```
    getent group sudo | cut -d: -f4
```
- ```
}
```

## Part 3 | Advanced Level

---

### 1. Creating function to get some information about a file through the command `internsctl file getinfo <file-name>**`

---

- Add the following code into the file `internsctl` present in `/bin` folder and save it.
- ```
getFileInfo () {
```
- ```
 if test -f "$3"; then
```
- ```
        echo "File: $3"
```
- ```
 displayPermissions() {
```
- ```
            case "$1" in
```
- ```
 0) echo "no";;
```
- ```
                1) echo "--x";;
```
- ```
 2) echo "-w-";;
```
- ```
                3) echo "-wx";;
```
- ```
 4) echo "r--";;
```
- ```
                5) echo "r-x";;
```
- ```
 6) echo "rw-";;
```
- ```
                7) echo "rwx";;
```
- ```
 esac
```
- ```
        }
```
- ```
 permissions=$(stat -c%a "$3")
```
- ```
        user=${permissions:0:1}
```
- ```
 group=${permissions:1:1}
```
- ```
        others=${permissions:2:1}
```
- ```
 echo "Access: -$(displayPermissions $user)$(displayPermissions
```
- ```
        $group)$(displayPermissions $others)"
```
- ```
 myFileSize=$(wc -c $3 | awk '{print $1}')
```
- ```
        echo "Size(B): $myFileSize"
```
- ```
 echo "Owner: $(stat -c '%U' $3)"
```

- else
- echo "internsctl: cannot access '\$3': No such file in current directory"
- fi
- }

## 2. Creating function to get specific information about a file through the command `internsctl file getinfo [options] <file-name>`\*\*

- Add the following code into the file `internsctl` present in `/bin` folder and save it.

```

• getSpecificFileInfo () {
• case "$3" in
• --size | -s)
• if test -f "$4"; then
• myFileSize=$(wc -c $4 | awk '{print $1}')
• if [$myFileSize -ge 1000]; then
• myFileSize=$(echo "$myFileSize *
0.001"|bc)
• printf "%.2f kilobytes\n"
$myFileSize
• else
• echo "$myFileSize bytes"
• fi
• else
• echo "internsctl: cannot access '$4': No
such file in current directory"
• fi ;;
•
• "--permissions" | "-p")
• if test -f "$4"; then
• displayPermissions() {
• case "$1" in
• 0) echo "no";;
• 1) echo "--x";;
• 2) echo "-w-";;
• 3) echo "-wx";;
• 4) echo "r--";;
• 5) echo "r-x";;
• 6) echo "rw-";;
• 7) echo "rwx";;
• esac
• }
• permissions=$(stat -c%a "$4")
• user=${permissions:0:1}
• group=${permissions:1:1}
• others=${permissions:2:1}
• echo "-$(displayPermissions
$user)$(displayPermissions $group)$(displayPermissions $others)"
• else
• echo "internsctl: cannot access '$4': No
such file in current directory"
• fi ;;
•
• "--owner" | "-o")
• if test -f "$4"; then
• echo "$(stat -c '%U' $4)"
• else

```

```

• echo "internsctl: cannot access '$4': No
such file in current directory"
• fi ;;
•
• "--last-modified" | "-m")
• if test -f "$4"; then
• echo "$(stat -c '%y' $4)"
• else
• echo "internsctl: cannot access '$4': No
such file in current directory"
• fi ;;
•
• *)
• if ["${3:0:1}" = "-"]; then
• echo "internsctl: invalid option"
• printf "\nUsage:\n internsctl file getinfo
[options] <file-name>\n"
• printf "\nTry 'internsctl --help' for more
information.\n"
• else
• printf "error: too many arguments\n"
• printf "\nUsage:\n internsctl file getinfo
<file-name>\n"
• printf "\n Try 'internsctl --help' for
additional help text.\n"
• fi ;;
• esac
• }

```

### **Options :**

--size, -s to print size  
 --permissions, -p to print file permissions  
 --owner, -o to print file owner  
 --last-modified, -m to print last modification time and date