

```
l httpd sshd dnsmasq pulseaudio conky tor Telegram firefox "[[:di
." LISTEN ESTABLISHED TIME_WAIT
d established)
Foreign Address      State      PID/Program name
0.0.0.0:*            LISTEN     380/dnsmasq
0.0.0.0:*            LISTEN     370/sshd
0.0.0.0:*            LISTEN     6363/cupsd
0.0.0.0:*            LISTEN     376/tor
0.0.0.0:*            LISTEN     478/pulseaudio
8.39.54.57:443       ESTABLISHED 496/firefox
149.154.167.91:80     ESTABLISHED 4082/Telegram
176.34.244.212:80     ESTABLISHED 520/conky
149.154.167.91:443    ESTABLISHED 4082/Telegram
54.149.244.33:443     ESTABLISHED 496/firefox
192.168.0.5:2049      ESTABLISHED -
192.168.0.5:443      TIME_WAIT  -
:::*                LISTEN     14629/httpd
:::*                LISTEN     380/dnsmasq
:::*                LISTEN     370/sshd
:::*                LISTEN     6363/cupsd
:::*                LISTEN     373/mpd
:::*                LISTEN     478/pulseaudio
0.0.0.0:*            380/dnsmasq
:::*                380/dnsmasq
```

Shell Scripting: A Comprehensive Guide

Learn how to write powerful shell scripts to automate tasks and improve productivity. This presentation covers essential techniques and provides practical examples.

Listing Files in a Directory

Q1) Write a shell script that displays a list of all the files in the current directory.

Use the following code to accomplish this:

```
#!/bin/bash/  
  
for entry in $(ls $search_dir); do  
  
    echo $entry  
  
done
```

Output

Running the script will produce the desired result:

```
a.txt b.txt s2.sh s3.sh s4.sh s.sh
```

Checking File Types

Q4) Write a shell script that receives any number of file names as arguments, checks if every argument supplied is a file or a directory and reports accordingly. When ever the argument is a file or directory.

```
#!/bin/bash

for arg in "$@"; do

    if [ -e "$arg" ]; then

        if [ -d "$arg" ]; then

            echo "$arg is a directory."

        elif [ -f "$arg" ]; then

            echo "$arg is a file."

        else

            echo "$arg exists but is neither a file nor
a directory."

        fi

    else

        echo "$arg does not exist." fi done
```

Output

Running the script will produce the desired result:

```
bash bash.sh bash.sh
```

```
bash.sh is a file.
```