

How to Limit the Use of CPU in Linux?

written by sysadmin | 3 May 2025

A large application that consumes a lot of CPU on a Linux server will cause unusual server conditions. Therefore, you have to limit the use of the CPU for the application.

Problem

How to limit the use of CPU in Linux?

Solution

You can use the `cpulimit` tool to limit CPU use. Below is the command to install the tool in Linux:

Debian/Ubuntu

```
sudo apt update
sudo apt-get install cpulimit -y
```

RockyLinux/AlmaLinux/CentOS

```
yum install epel-release -y
yum install cpulimit -y
```

OpenSUSE15

```
zypper install -y cpulimit
```

Here are some methods using the `cpulimit` command:

A. Using `--pid` option

To use the `--pid` or `-p` option, you need to know the PID of an application that you want to limit its CPU usage. To run this `cpulimit` tool, use the format below:

```
cpulimit --pid xxx --limit xxx --background
```

Where PID is the ID number of the ongoing application, you can see by using the format below:

```
ps aux | grep application_name
```

The limit is a percentage figure from the CPU that you want to limit for the use of the application. We will use the bash script to simulate this CPU server to be high. Create a **high_cpu.sh** file and copy the script below

```
#!/bin/bash

# Simple infinite loop that uses 100% CPU
while true
do
    : # No-op (no operation) to keep CPU busy
done
```


Run the command below to give permission and run this bash script:

```
chmod +x high_cpu.sh
./high_cpu.sh &
```

Run the **top** command on another terminal, and as you can see in the image below, the script uses 99 percent of the CPU on the server:

```
top - 14:53:11 up 31 min, 2 users, load average: 0.61, 0.59, 0.31
Tasks: 102 total, 2 running, 98 sleeping, 2 stopped, 0 zombie
%Cpu(s): 99.0 us, 0.0 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 1.0 si, 0.0 st
MiB Mem : 961.7 total, 486.8 free, 285.5 used, 333.5 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 676.2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1198	sysadmin	20	0	4752	3200	2944	R	99.0	0.3	0:22.53	high_cpu.sh
1089	sysadmin	20	0	9376	5632	3456	R	1.0	0.6	0:01.23	top
1	root	20	0	21896	12932	9476	S	0.0	1.3	0:00.68	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_p



Before using the cputlimit command

Now run the cputlimit command by limiting this script to run using only 50 percent of the CPU on this server:

```
cputlimit --pid 1198 --limit 50 --background
```

```
sysadmin@Ubuntu2404:~$ ./high_cpu.sh &
[3] 1198
sysadmin@Ubuntu2404:~$ cputlimit --pid 1198 --limit 50 --background
sysadmin@Ubuntu2404:~$ Process 1198 detected
```

Run the cputlimit command

If you look at the top command, this script is no longer utilizing 99 percent of the CPU on this server.

```
top - 14:57:52 up 35 min, 2 users, load average: 0.71, 0.66, 0.42
Tasks: 101 total, 1 running, 97 sleeping, 3 stopped, 0 zombie
%Cpu(s): 5.9 us, 0.0 sy, 0.0 ni, 94.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 961.7 total, 486.8 free, 285.5 used, 333.4 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 676.2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1198	sysadmin	20	0	4752	3200	2944	T	50.0	0.3	3:17.98	high_cpu.sh
1089	sysadmin	20	0	9376	5632	3456	R	1.0	0.6	0:02.17	top
1	root	20	0	21896	12932	9476	S	0.0	1.3	0:00.68	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/R-rcu_p

After using the cputlimit command

WARNING

I don't think there is a problem when an application consumes the CPU more than the limit you set after running the cputlimit command. For example, you limit the CPU to 50 percent using the cputlimit command, but in reality, the application runs with more than 50 percent in the use of CPU usage on your server. But at least the application doesn't take up a lot of CPU.

The disadvantage of using the **--pid** option is that if the application is restarted, the PID of the application will change, and you have to change the cputlimit command. So,

there is another option where you just write the name of the application in the `cpulimit` command using the `--exe` option.

B. Using `--exe` option

To use the `--exe` or `-e` option, you need to know the application name that you want to limit its CPU usage. To run this `cpulimit` tool, use the format below:

```
cpulimit --exe application_name --limit xxx --background
```

We will use the bash script to simulate a CPU server to be high with many PIDs. Create a `pids_high_cpu.sh` file in the folder `/etc` and copy the script below:

```
#!/bin/bash

# Number of processes to spawn
num_processes=5

# Function to keep the CPU busy in each child process
cpu_intensive_task() {
    while true
    do
        : # No-op command to keep the CPU busy
    done
}

# Spawn the specified number of processes, redirecting output to /dev/null
for ((i=0; i<num_processes; i++))
do
    cpu_intensive_task > /dev/null 2>&1 & # Run the task in the background
and suppress output
done

# Optionally, wait for all background processes to complete (won't happen
unless manually killed)
wait
```

Run the command below to give permission and run this bash script:

```
chmod +x /etc/pids_high_cpu.sh
/etc/pids_high_cpu.sh &
```

Run the **top** command on another terminal. As you can see in the image below, the script has many PIDs and uses a lot of CPU resources on the server:

```
top - 16:21:55 up 1:59, 3 users, load average: 3.95, 4.86, 15.72
Tasks: 111 total, 6 running, 105 sleeping, 0 stopped, 0 zombie
%Cpu(s): 78.9 us, 15.8 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 5.3 si, 0.0 st
MiB Mem : 961.7 total, 284.4 free, 304.1 used, 546.9 buff/cache
MiB Swap: 4096.0 total, 4081.7 free, 14.3 used. 657.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
29201	root	20	0	4752	1688	1536	R	21.6	0.2	0:05.11	pids_high_cpu.s
29200	root	20	0	4752	1688	1536	R	20.6	0.2	0:05.10	pids_high_cpu.s
29202	root	20	0	4752	1688	1536	R	20.6	0.2	0:05.10	pids_high_cpu.s
29203	root	20	0	4752	1688	1536	R	19.6	0.2	0:05.10	pids_high_cpu.s
29199	root	20	0	4752	1688	1536	R	18.6	0.2	0:05.09	pids_high_cpu.s
887	sysadmin	20	0	14960	3884	3072	S	1.0	0.4	0:01.82	sshd
29065	sysadmin	20	0	9376	5632	3456	R	1.0	0.6	0:04.15	top
1	root	20	0	21872	13184	9600	S	0.0	1.3	0:03.51	systemd

Display the app with many PIDs

Run the **cpulimit** command to restrict this script to utilizing just 50% of the server's CPU:

```
sudo cpulimit --exe pids_high_cpu.sh --limit 50 --background
```

```
root@Ubuntu2404:~# /etc/pids_high_cpu.sh &
[1] 29198
root@Ubuntu2404:~#
root@Ubuntu2404:~# ps aux | grep pids_high_cpu
root    29198  0.0  0.3  4752  3328 pts/3    S   16:21   0:00 /bin/bash /etc/pids_high_cpu.sh
root    29199 19.9  0.1  4752  1688 pts/3    R   16:21   0:19 /bin/bash /etc/pids_high_cpu.sh
root    29200 19.9  0.1  4752  1688 pts/3    R   16:21   0:19 /bin/bash /etc/pids_high_cpu.sh
root    29201 19.9  0.1  4752  1688 pts/3    R   16:21   0:19 /bin/bash /etc/pids_high_cpu.sh
root    29202 19.9  0.1  4752  1688 pts/3    R   16:21   0:19 /bin/bash /etc/pids_high_cpu.sh
root    29203 19.9  0.1  4752  1688 pts/3    R   16:21   0:19 /bin/bash /etc/pids_high_cpu.sh
root    29205  0.0  0.2  4088  2048 pts/3    S+  16:23   0:00 grep --color=auto pids_high_cpu
root@Ubuntu2404:~#
root@Ubuntu2404:~# sudo cpulimit --exe pids_high_cpu.sh --limit 50 --background
root@Ubuntu2404:~#
```

Run the **cpulimit** command with the **--exe** option

C. Using **--path** option

Besides using the **--exe** option, you can also use the **--path** option or **-P** option. To use this option, you have to know the path of the application that you want to limit its CPU usage. Use the format below to run the **--path** option in the

cpulimit command:

```
cpulimit --path /folder/path/of/the/application --limit xxx --background
```

For the same case, type the command below to use the cpulimit command with the **--path** option to limit CPU usage to **65** percent for **pids_high_cpu.sh** application in the **/etc** folder:

```
cpulimit --path /etc/pids_high_cpu.sh --limit 65 --background
```

WARNING

You can enter the cpulimit command into the crontab so the command will run automatically after the device restarts. But, the cpulimit command that you enter into the crontab can only use the **--exe** or **--path** option because, by default, the application name or application path will never change after the server restarts. Here is an example of the crontab:

```
@reboot          cpulimit --path /etc/pids_high_cpu.sh --limit 65 --background
```

Note

If you see an application running that generates a lot of PIDs, as shown in the image below:

```

[ 08:71] 4[
[ 38:83] 5[
[ 35:25] 6[
[ 50:05] 7[
7,896/15.36 tasks: 102, 1072 thr, 356 kthr, 6 running
300k/2,800 load average: 2.76 2.98 3.06
uptime: 02:30:03

Main | / 0
PID USER      PRI  NI  VIRT   RES   SHR   CPU%MEM  TIME- Command
2277 sysadmin  20   0 32.86 498M 245M R  97.1 3.2 2:15.89 /opt/vivaldi/vivaldi-bin
2678 sysadmin  20   0 32.86 498M 245M R  46.1 3.2 3:03.79 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15529 sysadmin  20   0 13956 675M 147M R  39.9 4.3 2:38.93 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15830 sysadmin  20   0 13956 674M 151M R  35.0 5.0 0:32.71 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15883 sysadmin  20   0 13956 675M 147M R  33.2 4.3 0:28.85 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
2676 sysadmin  20   0 32.86 498M 245M R  26.4 3.2 0:07.75 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
2688 sysadmin  20   0 32.86 498M 245M R  9.8 3.2 0:28.64 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
18350 sysadmin  20   0 13784 296 584 R  9.8 0.0 0:07.75 httpd
18371 sysadmin  20   0 32228 151M 1724 R  8.0 1.2 0:11.49 /usr/bin/perl /usr/bin/shutter
1898 sysadmin  20   0 9758 167M 115M R  7.4 1.1 7:08.89 /usr/lib/xorg/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
2655 sysadmin  20   0 32.86 498M 245M R  4.9 3.2 5:00.98 /opt/vivaldi/vivaldi-bin
19681 sysadmin  20   0 13946 149M 8576 R  4.3 1.0 0:00.27 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8177 sysadmin  20   0 46.76 135M 118M R  3.7 0.9 4:25.86 /opt/vivaldi/vivaldi-bin --type=utility --utility-sub-type=media.mojom.AudioService --lang=en-US --running-vivaldi --service-sandbox-type=none --crashpad-handler-pid=2661
15529 sysadmin  20   0 13956 675M 147M R  3.7 4.3 0:14.83 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15555 sysadmin  20   0 13956 675M 147M R  3.7 4.3 0:04.51 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15756 sysadmin  20   0 13956 674M 151M R  3.7 5.6 2:54.76 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
1671 sysadmin  20   0 4700 4.68M 468 R  3.1 0.3 0:19.92 xfce4-panel --display :8.0 --sm-client-id 2c2a2f6a-b099-4dc6-a30d-d054fc50b0a3
15334 sysadmin  20   0 13956 675M 147M R  3.1 4.3 0:06.36 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15559 sysadmin  20   0 13956 675M 147M R  3.1 4.3 0:03.93 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
15560 sysadmin  20   0 13956 675M 147M R  3.1 4.3 0:03.68 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
2681 sysadmin  20   0 32.86 498M 245M R  2.5 3.2 1:55.85 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8090 sysadmin  20   0 10406 607M 170M R  2.5 3.9 0:13.66 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8178 sysadmin  20   0 32.76 135M 115M R  2.5 0.9 2:06.94 /opt/vivaldi/vivaldi-bin --type=utility --utility-sub-type=media.mojom.AudioService --lang=en-US --running-vivaldi --service-sandbox-type=none --crashpad-handler-pid=2661
1375 sysadmin  -21   0 1248 2.584 6616 R  1.8 0.1 3:38.51 /usr/bin/pipewire
1658 sysadmin  20   0 8470 99.0M 8768 R  1.8 0.6 1:31.12 xfwm4 --display :8.0 --sm-client-id 233ab71c4-49e5-4a8f-86a4-6161ea34aa
2150 sysadmin  20   0 13946 607M 170M R  1.8 2.1 1:03.06 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8174 sysadmin  20   0 14846 607M 170M R  1.8 3.9 2:24.06 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8176 sysadmin  20   0 14846 607M 170M R  1.8 3.9 1:38.12 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
18316 sysadmin  20   0 13946 149M 8576 R  1.8 1.0 0:00.05 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
1377 sysadmin  20   0 13946 607M 170M R  1.2 3.9 1:35.87 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
1783 sysadmin  20   0 4698 5.492 236 R  1.2 0.3 0:13.61 xfce4-terminal --geometry=238x57 --display :8.0 --role=xfce4-terminal-1749064703-845186640 --workspace 0 --show-menubar --show-borders --hide-toolbar --active-tab --worki
5699 sysadmin  20   0 32.76 135M 115M R  1.2 0.9 2:16.41 /opt/vivaldi/vivaldi-bin --type=utility --utility-sub-type=media.mojom.AudioService --lang=en-US --running-vivaldi --service-sandbox-type=none --crashpad-handler-pid=2661
7423 sysadmin  20   0 13946 123M 8064 R  1.2 0.8 0:28.68 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --extension-process --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8205 sysadmin  20   0 13946 607M 170M R  1.2 3.9 1:35.87 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
1561 sysadmin  9  11 1558 5.672 884 R  0.6 0.4 2:16.24 /usr/bin/pipewire-pulse
1698 sysadmin  20   0 4548 4.120 8056 R  0.6 0.3 0:00.27 /usr/libexec/xdg-desktop-gateway/xfce4/panel/wrapper-2.0 /usr/lib/xdg/64-linux-gnu/xfce4/panel/plugins/libwhiskermenu.so.1 14680071 whiskermenu Whisker Menu Show a menu to easily
2784 sysadmin  20   0 32.66 189M 144M R  0.6 1.2 2:03.06 /opt/vivaldi/vivaldi-bin --type=utility --utility-sub-type=network.mojom.NetworkService --lang=en-US --running-vivaldi --service-sandbox-type=none --crashpad-handler-pid=2661
2848 sysadmin  20   0 13946 204M 100M R  0.6 1.3 0:55.67 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
6874 sysadmin  20   0 41.11 131M 1784 R  0.6 0.8 0:07.73 /opt/microsoft/msedge/msedge --type=gpu-process --crashpad-handler-pid=6634 --enable-crash-reporter= --change-stack-guard-on-fork=enable --gpu-preferences=UAAAAAAAAAAGAA
6944 sysadmin  20   0 13956 295M 128M R  0.6 1.9 0:48.16 /opt/microsoft/msedge/msedge --type=renderer --crashpad-handler-pid=6634 --enable-crash-reporter= --change-stack-guard-on-fork=enable --lang=en-US --js-flags=--ms-user-lc
7322 sysadmin  20   0 26278 148M 185M R  0.6 0.9 0:19.85 /home/sysadmin/Documents/WindTerm 2.7.0/WindTerm
8991 sysadmin  20   0 14846 607M 170M R  0.6 3.9 0:28.50 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8171 sysadmin  20   0 14846 607M 170M R  0.6 3.9 1:13.80 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8173 sysadmin  20   0 14846 607M 170M R  0.6 3.9 0:29.43 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8175 sysadmin  20   0 14846 607M 170M R  0.6 3.9 0:12.72 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1
8183 sysadmin  20   0 14846 607M 170M R  0.6 3.9 0:12.72 /opt/vivaldi/vivaldi-bin --type=renderer --crashpad-handler-pid=2661 --enable-crash-reporter=stable --change-stack-guard-on-fork=enable --lang=en-US --num-raster-threads=1

```

The application with many PIDs

From the picture above, you can see that the application Vivaldi has many PIDs. So, you have to write a script to restrict apps with numerous PIDs. Here is a copy of the bash script:

```

#!/bin/bash

LIMIT=50
PROCESS_NAME="vivaldi-bin"

# Kill existing cpulimit instances related to vivaldi-bin
pkill -f "cpulimit -p" 2>/dev/null

# Get all vivaldi PIDs
PIDS=$(pidof $PROCESS_NAME)

if [[ -z "$PIDS" ]]; then
    echo "[$(date)] $PROCESS_NAME not running."
    exit 0
fi

echo "[$(date)] Limiting CPU for PIDs: $PIDS"

# Start cpulimit for each PID in background
for pid in $PIDS; do
    cpulimit -p "$pid" -l "$LIMIT" > /dev/null 2>&1 &
done

```

Change the **LIMIT** and **PROCESS_NAME** sections according to your needs and permit the script to be executed. If the script runs, it will limit the CPU to the PIDs.

```
sysadmin@LinuxMint:~$ ps aux | grep cpulimit
root      20789  0.0  0.0   2696 1664 pts/5    T-   13:43  0:00 cpulimit -p 20523 16580 15756 15629 15529 12859 11858 11696 9891 9863 9834 9802 9694 8564 8513 8454 8148 8090 8081 7407 5669 5135 3150 2868 28
33 2819 2796 2755 2729 2711 2698 2696 2672 2670 2669 2655 -1 30
root      20799  0.0  0.0   2696 1664 pts/5    T-   13:43  0:00 cpulimit -p 20523 16580 15756 15629 15529 12859 11858 11696 9891 9863 9834 9802 9694 8564 8513 8454 8148 8090 8081 7407 5669 5135 3150 2868 28
33 2819 2796 2755 2729 2711 2698 2696 2672 2670 2669 2655 -1 30
root      20821  0.0  0.0   2696 1664 pts/5    T-   13:43  0:00 cpulimit -p 20889 28523 16580 15756 15629 15529 12859 11858 11696 9891 9863 9834 9802 9694 8564 8513 8454 8148 8090 8081 7407 5669 5135 3150 2
868 2823 2819 2796 2755 2729 2711 2698 2696 2672 2670 2669 2655 -1 30
root      20838  0.0  0.0   2696 1664 pts/5    T-   13:43  0:00 cpulimit -p 20889 28523 16580 15756 15629 15529 12859 11858 11696 9891 9863 9834 9802 9694 8564 8513 8454 8148 8090 8081 7407 5669 5135 3150 2
868 2823 2819 2796 2755 2729 2711 2698 2696 2672 2670 2669 2655 -1 30
sysadmin  30856  0.0  0.0   9280 2304 pts/2    S+   14:37  0:00 grep --color=auto cpulimit
sysadmin@LinuxMint:~$
```

Check the cpulimit command that runs in the background

Enter the script into the crontab using the script below:

```
#Limit CPU vivaldi
*/30 * * * * /root/limit_vivaldi.sh
@reboot /root/limit_vivaldi.sh
```

The script will restart every 30 minutes, and if the device restarts, the script will start automatically.

References

- tecmint.com
- linuxsec.org
- id.ubunlog.com
- linuxsec.org
- youtube.com