Stress Testing CPU with dd command

One of the really interesting ways of stress testing CPUs on your system is using **dd command**. By making it work with /dev/zero and /dev/null (which are not real devices in terms of disk I/O), you’re effectively getting **dd** to shift a whole lot of zero bytes between addresses in virtual memory – thus causing CPU (and possibly a bit of RAM) stress.

Screenshot above shows how a stressed system looks – all CPU cores are 100% busy.

**How To Stress CPU with dd command**

My example refers to **/proc/cpuinfo** file, which is Linux specific. We confirm the number of CPU cores available on the system and then start a number of dd processes in background:

```
 greys@xps:~/proj$ for i in `seq 1 $(cat /proc/cpuinfo | grep "physical id" | wc -l)`; do dd if=/dev/zero of=/dev/null & done
 [1] 6424
 [2] 6425
 [3] 6426
```
How To Check CPU status with htop

If you don’t have the htop command on your Linux system yet, you can easily install it. On my Dell XPS 13 laptop with Linux Mint it means this:

greys@xps:/proj$ sudo apt install htop
[sudo] password for greys:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
htop
0 upgraded, 1 newly installed, 0 to remove and 245 not upgraded.
Need to get 80.0 kB of archives.
After this operation, 221 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 htop amd64 2.1.0-3 [80.0 kB]
Fetched 80.0 kB in 0s (463 kB/s)
Selecting previously unselected package htop.
(Reading database ... 259002 files and directories currently installed.)
Preparing to unpack .../htop_2.1.0-3_amd64.deb ...
Unpacking htop (2.1.0-3) ...
Processing triggers for mime-support (3.60ubuntu1) ...
Processing triggers for desktop-file-utils (0.23+linuxmint4) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
Setting up htop (2.1.0-3) ...

After we run htop:
Once stress testing is done, we need to kill all the `dd` processes:

greys@xps:~/proj$ pkill dd
pkill: killing pid 2 failed: Operation not permitted
pkill: killing pid 135 failed: Operation not permitted
[1] Terminated dd if=/dev/zero of=/dev/null
[2] Terminated dd if=/dev/zero of=/dev/null
[3] Terminated dd if=/dev/zero of=/dev/null
[6] Terminated dd if=/dev/zero of=/dev/null
[7] Terminated dd if=/dev/zero of=/dev/null
[8] Terminated dd if=/dev/zero of=/dev/null
[5] Terminated dd if=/dev/zero of=/dev/null

... and then re-run htop again:

greys@xps:~/proj$ htop
That's it for today!

See Also

- Monitor processes, CPU and RAM with htop
- Test disk throughput with dd
- ISO to bootable USB in MacOS
- dd command