

Example to use the Linux TOP command

Michael C.

Feb 20 2020

It is possible to use the Lua **os.execute** functionality to run a Linux command. Note that the `os.execute` doesn't return the output of the command, just the return state code. It is recommended to use **popen**.

Also note that running **top** itself is that it doesn't quit by default, so an `os.execute` OR a `popen` will never return. The solution would be to run **top** as noted below, using the **top** and **grep** command to parse the returned data:

```
top -b -n 1 |grep ^CPU
```

This generates output like this, if run from the command prompt on the Linux shell:

```
CPU: 11.1% usr 10.0% sys 0.0% nic 70.0% idle 0.0% io 0.0% irq 0.0% sirq
```

To show up to a certain section, such as up to the CPU value, use the following **grep** switches.

```
top -b -n 1 |grep ^CPU|cut -d" " -f 1-2
```

This generates output like this, if run at the command prompt at the Linux shell:

```
CPU: 11.1%
```

To include the values in a PHD application, a Lua function is needed. This example generates a string **s** with the values returned from the **cmd** Linux command:

```
function os.capture(cmd, raw)
  --cmd is the desired Linux command string
  --raw is the switch to show the raw returned value with the CR/LF.

  local f = assert(io.popen(cmd, 'r'))
  local s = assert(f:read('*a'))

  f:close()

  if raw then
    return s
  end

  s = string.gsub(s, '^%s+', '')
  s = string.gsub(s, '%s+$', '')
  s = string.gsub(s, '[\n\r]+', ' ')

  return s
end
```

Then, they could call it like this:

```
local cpu_usage = os.capture("top -b -n 1 |grep ^CPU|cut -d" " -f 1-2", false)
```

After this call, the local `cpu_usage` variable will be a string containing "CPU: xx.x%".

This example shows how to set just the value, without the "CPU: "

```
local cpu_usage = os.capture("top -b -n 1 |grep ^CPU|cut -d" " -f 2", false)
```

After this call, the local `cpu_usage` variable will be a string containing "xx.x%".

An example of the file created is shown below. Note this can be exported to Excel and sorted for further analysis if needed.

```
[2020-03-11 11:11:51.217]
[2020-03-11 11:11:51.234] [00:36:50 phd50 /root] top -b
[2020-03-11 11:11:55.256] Mem: 22504K used, 101692K free, 32K shrd, 0K buff, 4760K cached
[2020-03-11 11:11:55.262] CPU: 50.0% usr 28.5% sys 0.0% nic 14.2% idle 0.0% io 0.0% irq 7.1% sirq
[2020-03-11 11:11:55.269] Load average: 1.08 0.80 0.65 1/58 982
[2020-03-11 11:11:55.272] PID PPID USER STAT VSZ %VSZ CPU %CPU COMMAND
[2020-03-11 11:11:55.277] 761 1 root R 35352 28.3 0 52.6 sbengine -v Application.gapp
[2020-03-11 11:11:55.284] 777 1 root S 1912 1.5 0 13.1 gstreamer-backend -pv4l2src device=/dev/video0 ! fbdevsink sync=true device=/dev/fb5 -v
[2020-03-11 11:11:55.296] 982 962 root R 1108 0.8 0 13.1 top -b
[2020-03-11 11:11:55.301] 688 1 root S 3300 2.6 0 6.5 /usr/local/parker/arm/bin/nexus_vybrid -c /usr/local/parker/config/nx_config.xml
[2020-03-11 11:11:55.312] 689 1 root S 10344 8.3 0 0.0 /usr/local/parker/arm/bin/autoinstaller -c /usr/local/parker/config/ai.xml
[2020-03-11 11:11:55.322] 760 1 root S 2828 2.2 0 0.0 j1939d -c /root/j1939.json -a j1939
[2020-03-11 11:11:55.330] 655 1 root S 2460 1.9 0 0.0 /usr/local/parker/arm/bin/sidekick -c /usr/local/parker/config/sk_config.xml
[2020-03-11 11:11:55.340] 690 1 root S 2148 1.7 0 0.0 /usr/local/parker/arm/bin/pds
[2020-03-11 11:11:55.347] 654 1 root S 2024 1.6 0 0.0 /usr/local/parker/arm/bin/ncc spi0.0
[2020-03-11 11:11:55.354] 641 1 root S 2012 1.6 0 0.0 /usr/local/parker/arm/bin/versup
[2020-03-11 11:11:55.361] 746 1 user S 1116 0.9 0 0.0 -sh
[2020-03-11 11:11:55.366] 1 0 root S 1112 0.8 0 0.0 init
[2020-03-11 11:11:55.370] 962 746 root S 1112 0.8 0 0.0 -sh
[2020-03-11 11:11:55.375] 611 1 root S 1100 0.8 0 0.0 syslogd
[2020-03-11 11:11:55.380] 725 1 root S 1040 0.8 0 0.0 /usr/sbin/dropbear -w
[2020-03-11 11:11:55.386] 3 2 root SW 0 0.0 0 0.0 [ksoftirqd/0]
[2020-03-11 11:11:55.392] 569 2 root SW 0 0.0 0 0.0 [ubi_bgt0d]
[2020-03-11 11:11:55.397] 558 2 root SW 0 0.0 0 0.0 [kworker/u2:2]
[2020-03-11 11:11:55.402] 525 2 root SW 0 0.0 0 0.0 [irq/76-st1232]
[2020-03-11 11:11:55.408] 7 2 root SW 0 0.0 0 0.0 [watchdog/0]
[2020-03-11 11:11:55.413] 310 2 root SW 0 0.0 0 0.0 [kworker/0:1]
[2020-03-11 11:11:55.418] 9 2 root SW 0 0.0 0 0.0 [kdevtmpfs]
[2020-03-11 11:11:55.423] 572 2 root SW 0 0.0 0 0.0 [ubifs_bgt0_0]
[2020-03-11 11:11:55.431] 8 2 root SW< 0 0.0 0 0.0 [khelper]
[2020-03-11 11:11:55.436] 6 2 root SW 0 0.0 0 0.0 [kworker/u2:0]
[2020-03-11 11:11:55.439] 4 2 root SW 0 0.0 0 0.0 [kworker/0:0]
[2020-03-11 11:11:55.445] 5 2 root SW< 0 0.0 0 0.0 [kworker/0:0H]
[2020-03-11 11:11:55.450] 38 2 root SW 0 0.0 0 0.0 [khungtaskd]
[2020-03-11 11:11:55.455] 321 2 root SW 0 0.0 0 0.0 [kswapd0]
[2020-03-11 11:11:55.461] 186 2 root SW< 0 0.0 0 0.0 [bioset]
[2020-03-11 11:11:55.465] 188 2 root SW< 0 0.0 0 0.0 [kblockd]
[2020-03-11 11:11:55.471] 2 0 root SW 0 0.0 0 0.0 [kthreadd]
[2020-03-11 11:11:55.476] 557 2 root SW< 0 0.0 0 0.0 [deferwq]
[2020-03-11 11:11:55.480] 368 2 root SW 0 0.0 0 0.0 [fsnotify_mark]
[2020-03-11 11:11:55.486] 494 2 root SW 0 0.0 0 0.0 [spi0]
[2020-03-11 11:11:55.490] 936 2 root SW< 0 0.0 0 0.0 [ci_otg]
[2020-03-11 11:11:55.496] 183 2 root SW< 0 0.0 0 0.0 [writeback]
[2020-03-11 11:11:55.502] 185 2 root SW< 0 0.0 0 0.0 [crypto]
[2020-03-11 11:12:00.361] Mem: 22504K used, 101692K free, 32K shrd, 0K buff, 4760K cached
[2020-03-11 11:12:00.368] CPU: 47.6% usr 9.2% sys 0.0% nic 42.8% idle 0.0% io 0.0% irq 0.2% sirq
[2020-03-11 11:12:00.373] Load average: 1.08 0.80 0.65 2/58 982
```