Top Linux command line

Contents

1 Article purpose

This article provides the basic information needed to start using the Linux® tool: **top**[1].

2 Introduction

The following table provides a brief description of the tool, as well as its availability depending on the software packages:

: this tool is either present (ready to use or to be activated), or can be integrated and activated on the software package.

: this tool is not present and cannot be integrated, or it is present but cannot be activated on the software package.

Tool			STM32MPU Embedded Software distribution			STM32MPU Embedded Software distribution for Android™		
Name	Category	Purpose	Starter Package	Developer Package	Distribution Package	Starter Package	Developer Package	Distribution Package
		The **top**[1] program provides a dynamic real-time view of a running system. It can display system summary information as well as a list of tasks currently being managed by the Linux						

Tool			STM32MPU Embedded Software distribution			STM32MPU Embedded Software distribution for Android™		
Name	Category	Purpose	Starter Package	Developer Package	Distribution Package	Starter Package	Developer Package	Distribution Package
top	Monitoring tools	kernel. The types of system summary information shown and the types, order and size of information displayed for tasks are all user configurable and that configuration can be made persistent across restarts	✓	✓	✓	✓	✓	✓

Tool			STM32MPU Embedded Software distribution			STM32MPU Embedded Software distribution for Android™		
Name	Category	Purpose	Starter Package	Developer Package	Distribution Package	Starter Package	Developer Package	Distribution Package
		. (*Extracted from man page*[1])						

3 Installing the trace and debug tool on your target board

3.1 Using the STM32MPU Embedded Software Distribution

top is installed by default and is ready to be used with all STM32MPU Embedded Software Packages.

```
Board $> which top
/usr/bin/top
```

It is integrated into the Weston image distribution by using the following meta-st package recipe to enable the procps module: *layers/meta-st/meta-st-openstlinux/recipes-st/packagegroups/packagegroup-framework-tools.bb*.

```
RDEPENDS_packagegroup-framework-tools-core = "\
   grep            \
   util-linux      \
   util-linux-lscpu \
   procps          \
   kbd             \
   ...
```

3.2 Using the STM32MPU Embedded Software distribution for Android™

top is installed by default and ready to be used with all STM32MPU Embedded Software Packages for Android™.

It comes with the **toybox**:

```
Board $> which top | xargs ls -la
/system/bin/top -> toybox
```

4 Getting started

- Displaying individual threads: without this command-line option a summation of all threads in each process is shown

```
Board $> top -H
top - 16:50:23 up  4:31,  1 user,  load average: 0.38, 0.38, 0.37
Threads: 130 total,   1 running, 129 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.3 us, 10.9 sy,  0.0 ni, 84.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    871.9 total,    673.1 free,    107.7 used,     91.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    736.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  895 root      20   0    2652   1608   1220 R  25.0   0.2   0:00.08 top
    1 root      20   0   24164   4808   3104 S   0.0   0.5   0:17.88 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.01 kworker/0:0H-kblockd
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.04 kworker/u4:0-
events_unbound
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0   0.0   0:01.42 ksoftirqd/0
   10 root      20   0       0      0      0 I   0.0   0.0   0:03.19 rcu_preempt
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_sched
   12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_bh
   13 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
   16 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/1
   17 root      20   0       0      0      0 S   0.0   0.0   0:01.64 ksoftirqd/1
   19 root       0 -20       0      0      0 I   0.0   0.0   0:00.07 kworker/1:0H-kblockd
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.04 kdevtmpfs
   21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   22 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   24 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
   26 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kcompactd0
   27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
   28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
   29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 ata_sff
   30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
   31 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
   32 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rpciod
   33 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/u5:0
   34 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 xprtiod
   35 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kswapd0
   36 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 nfsiod
```

- Displaying only given PID:

```
Board $> top -p<PID>
```

<PID> can be a given known value, or it could be useful to use the *pidof* macro:

```
Board $> top -p`pidof netdata`
top - 16:59:39 up  4:40,  1 user,  load average: 0.46, 0.46, 0.41
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
```

```
%Cpu(s):  0.0 us,  2.9 sy,  0.0 ni, 97.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    871.9 total,    673.2 free,    107.6 used,     91.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    736.6 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 389 root      20   0  131340  15528   2036 S   0.0   1.7   2:39.63 netdata
```

and by adding **-H** option:

```
Board $> top -p`pidof netdata` -H
top - 16:58:15 up  4:39,  1 user,  load average: 0.62, 0.48, 0.41
Threads:  12 total,   0 running,  12 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.5 sy,  0.0 ni, 99.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    871.9 total,    673.1 free,    107.6 used,     91.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    736.6 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 407 root      20   0  131340  15528   2036 S   0.3   1.7   1:06.85 netdata
 409 root      20   0  131340  15528   2036 S   0.3   1.7   0:46.08 netdata
 389 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.23 netdata
 406 root      20   0  131340  15528   2036 S   0.0   1.7   0:13.30 netdata
 408 root      20   0  131340  15528   2036 S   0.0   1.7   0:05.31 netdata
 411 root      20   0  131340  15528   2036 S   0.0   1.7   0:03.51 netdata
 412 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.09 netdata
 413 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.08 netdata
 414 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.92 netdata
 418 root      20   0  131340  15528   2036 S   0.0   1.7   0:01.24 netdata
 419 root      20   0  131340  15528   2036 S   0.0   1.7   0:21.12 netdata
 425 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.00 netdata
```

Note that several PIDs can be given by using the following syntax:

```
Board $> top -p<PID1>,<PID2>,<PID3>...
```

- Sorting informations

When the top command result is displayed, type "F", then select a filter: please see highlighted part below:

```
Fields Management for window 1:Def, whose current sort field is %CPU
  Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
  'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id             SUPGRPS = Supp Groups Names
* USER    = Effective User Name    TGID    = Thread Group Id
* PR      = Priority               OOMa    = OOMEM Adjustment
* NI      = Nice Value             OOMs    = OOMEM Score current
* VIRT    = Virtual Image (KiB)    ENVIRON = Environment vars
* RES     = Resident Size (KiB)    vMj     = Major Faults delta
* SHR     = Shared Memory (KiB)    vMn     = Minor Faults delta
* S       = Process Status         USED    = Res+Swap Size (KiB)
* %CPU    = CPU Usage              nsIPC   = IPC namespace Inode
* %MEM    = Memory Usage (RES)     nsMNT   = MNT namespace Inode
* TIME+   = CPU Time, hundredths   nsNET   = NET namespace Inode
* COMMAND = Command Name/Line      nsPID   = PID namespace Inode
  PPID    = Parent Process pid     nsUSER  = USER namespace Inode
  UID     = Effective User Id      nsUTS   = UTS namespace Inode
  RUID    = Real User Id           LXC     = LXC container name
  RUSER   = Real User Name         RSan    = RES Anonymous (KiB)
  SUID    = Saved User Id          RSfd    = RES File-based (KiB)
  SUSER   = Saved User Name        RSlk    = RES Locked (KiB)
```

```
  GID     = Group Id                RSsh    = RES Shared (KiB)
  GROUP   = Group Name              CGNAME  = Control Group name
  PGRP    = Process Group Id        NU      = Last Used NUMA node
  TTY     = Controlling Tty
  TPGID   = Tty Process Grp Id
  SID     = Session Id
  nTH     = Number of Threads
  P       = Last Used Cpu (SMP)
  TIME    = CPU Time
  SWAP    = Swapped Size (KiB)
  CODE    = Code Size (KiB)
  DATA    = Data+Stack (KiB)
  nMaj    = Major Page Faults
  nMin    = Minor Page Faults
  nDRT    = Dirty Pages Count
  WCHAN   = Sleeping in Function
  Flags   = Task Flags <sched.h>
  CGROUPS = Control Groups
  SUPGIDS = Supp Groups IDs
```

- Displaying where threads are sleeping

When the top command result is displayed, type "F", then go to the WCHAN parameter. Type "d" or <space> to toggle the display of this field.

```
Fields Management for window 1:Def, whose current sort field is %CPU
  Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
  'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id              SUPGRPS = Supp Groups Names
* USER    = Effective User Name     TGID    = Thread Group Id
* PR      = Priority                OOMa    = OOMEM Adjustment
* NI      = Nice Value              OOMs    = OOMEM Score current
* VIRT    = Virtual Image (KiB)     ENVIRON = Environment vars
* RES     = Resident Size (KiB)     vMj     = Major Faults delta
* SHR     = Shared Memory (KiB)     vMn     = Minor Faults delta
* S       = Process Status          USED    = Res+Swap Size (KiB)
* %CPU    = CPU Usage               nsIPC   = IPC namespace Inode
* %MEM    = Memory Usage (RES)      nsMNT   = MNT namespace Inode
* TIME+   = CPU Time, hundredths    nsNET   = NET namespace Inode
* COMMAND = Command Name/Line       nsPID   = PID namespace Inode
  PPID    = Parent Process pid      nsUSER  = USER namespace Inode
  UID     = Effective User Id       nsUTS   = UTS namespace Inode
  RUID    = Real User Id            LXC     = LXC container name
  RUSER   = Real User Name          RSan    = RES Anonymous (KiB)
  SUID    = Saved User Id           RSfd    = RES File-based (KiB)
  SUSER   = Saved User Name         RSlk    = RES Locked (KiB)
  GID     = Group Id                RSsh    = RES Shared (KiB)
  GROUP   = Group Name              CGNAME  = Control Group name
  PGRP    = Process Group Id        NU      = Last Used NUMA node
  TTY     = Controlling Tty
  TPGID   = Tty Process Grp Id
  SID     = Session Id
  nTH     = Number of Threads
  P       = Last Used Cpu (SMP)
  TIME    = CPU Time
  SWAP    = Swapped Size (KiB)
  CODE    = Code Size (KiB)
  DATA    = Data+Stack (KiB)
  nMaj    = Major Page Faults
```

```
  nMin    = Minor Page Faults
  nDRT    = Dirty Pages Count
* WCHAN   = Sleeping in Function
  Flags   = Task Flags <sched.h>
  CGROUPS = Control Groups
  SUPGIDS = Supp Groups IDs
```

This option enables the display of wchan.

```
Board $> top -p `pidof netdata`,`pidof python3`
top - 17:19:15 up  5:00,  1 user,  load average: 0.47, 0.45, 0.45
Tasks:   2 total,   0 running,   2 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  1.0 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    871.9 total,    672.9 free,    107.8 used,     91.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    736.4 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND WCHAN
 389 root      20   0  131340  15528   2036 S   0.7   1.7   2:50.72 netdata sys_pause
 424 root      20   0   43496  14860   5680 S   0.0   1.7   0:33.02 python3 poll_sche+
```

- Showing only one iteration of top and putting it in batch mode (useful for putting in a file)

```
Board $> top -p`pidof netdata` -H -b -n1
top - 17:26:53 up  5:07,  1 user,  load average: 0.46, 0.47, 0.45
Threads:  12 total,   0 running,  12 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  2.9 sy,  0.0 ni, 97.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    871.9 total,    672.9 free,    107.8 used,     91.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    736.5 avail Mem

 PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 389 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.23 netdata
 406 root      20   0  131340  15528   2036 S   0.0   1.7   0:14.67 netdata
 407 root      20   0  131340  15528   2036 S   0.0   1.7   1:13.64 netdata
 408 root      20   0  131340  15528   2036 S   0.0   1.7   0:05.83 netdata
 409 root      20   0  131340  15528   2036 S   0.0   1.7   0:50.85 netdata
 411 root      20   0  131340  15528   2036 S   0.0   1.7   0:03.92 netdata
 412 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.10 netdata
 413 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.09 netdata
 414 root      20   0  131340  15528   2036 S   0.0   1.7   0:01.00 netdata
 418 root      20   0  131340  15528   2036 S   0.0   1.7   0:01.38 netdata
 419 root      20   0  131340  15528   2036 S   0.0   1.7   0:23.24 netdata
 425 root      20   0  131340  15528   2036 S   0.0   1.7   0:00.00 netdata
```

5 References

- 1.01.11.2 http://linux.die.net/man/1/top

Central processing unit

Inter-Processor Communication

TeleTYpewriter

symetric multiprocessing