NAME

top − display Linux tasks

SYNOPSIS

top −**hv** | −**abcHimMsS** −**d** *delay* −**n** *iterations* −**p** *pid* [, *pid* ...]

The traditional switches '-' and whitespace are optional.

DESCRIPTION

The **top** 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 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.

The program provides a limited interactive interface for process manipulation as well as a much more extensive interface for personal configuration −− encompassing every aspect of its operation. And while **top** is referred to throughout this document, you are free to name the program anything you wish. That new name, possibly an alias, will then be reflected on top's display and used when reading and writing a configuration file.

OVERVIEW

Documentation

The remaining Table of Contents

1. COMMAND−LINE Options
2. FIELDS / Columns
   a. DESCRIPTIONS of Fields
   b. SELECTING and ORDERING Columns
3. INTERACTIVE Commands
   a. GLOBAL Commands
   b. SUMMARY Area Commands
   c. TASK Area Commands
   d. COLOR Mapping
4. ALTERNATE−DISPLAY Mode
   a. WINDOWS Overview
   b. COMMANDS for Windows
5. FILES
   a. SYSTEM Configuration File
   b. PERSONAL Configuration File
6. STUPID TRICKS Sampler
   a. Kernel Magic
   b. Bouncing Windows
   c. The Big Bird Window

7. BUGS, 8. HISTORY Former top, 9. AUTHOR, 10. SEE ALSO

Operation

When operating top, the two most important keys are help ('h' or '?') and quit ('q') key. Alternatively, you could simply use the traditional interrupt key ('^C') when you're done.

When you start top for the first time, you'll be presented with the traditional screen elements: 1) Summary Area; 2) Message/Prompt Line; 3) Columns Header; 4) Task Area. There will, however, be some

differences when compared to the former top.

Highlighting

Summary_Area: There is no highlighting for load/uptime and only values are highlighted for other elements.

Task_Area: Tasks running (or ready to run) will be highlighted, and bold is only one way of emphasizing such processes.

Content/Labels

Summary_Area: The program name is shown, perhaps a symlink or alias. The Cpu(s) state label hints at other possibilities. The memory stats use a lower case 'k'.

Columns_Header: Will show a new field and some changed labels. More new fields will be found as you customize your top.

Note: the width of top's display will be limited to 512 positions. Displaying all fields requires a minimum of 160 characters. The remaining width could be used for the 'Command' column.

Startup Defaults

The following startup defaults assume no configuration file, thus no user customizations. Even so, items shown with an asterisk ('*') could be overridden through the command-line.

Global_defaults
'A' - Alt display Off (full-screen)
* 'd' - Delay time 3.0 seconds
'I' - Irix mode On (no, 'solaris' smp)
* 'p' - PID monitoring Off
* 's' - Secure mode Off (unsecured)
'B' - Bold disable Off
Summary_Area_defaults
'l' - Load Avg/Uptime On (thus program name)
't' - Task/Cpu states On (1+1 lines, see '1')
'm' - Mem/Swap usage On (2 lines worth)
'1' - Single Cpu On (thus 1 line if smp)
Task_Area_defaults
'b' - Bold hilite On (not 'reverse')
* 'c' - Command line Off (name, not cmdline)
* 'H' - Threads Off (show all threads)
* 'i' - Idle tasks On (show all tasks)
'R' - Reverse sort On (pids high-to-low)
* 'S' - Cumulative time Off (no, dead children)
'x' - Column hilite Off (no, sort field)
'y' - Row hilite On (yes, running tasks)
'z' - color/mono Off (no, colors)

1. COMMAND-LINE Options

The command-line syntax for top consists of:

−**hv** | -**abcHimMsS** −**d** *delay* −**n** *iterations* −**p** *pid* [,*pid*...]

The typically mandatory switches ('-') and even whitespace are completely optional.

−**a** : **Sort by memory usage**
This switch makes top to sort the processes by allocated memory

−**b** : **Batch mode** operation
Starts top in 'Batch mode', which could be useful for sending output from top to other programs or to a file. In this mode, top will not accept input and runs until the iterations limit you've set with the '-n' command−line option or until killed.

−**c** : **Command line/Program name** toggle
Starts top with the last remembered 'c' state reversed. Thus, if top was displaying command lines, now that field will show program names, and visa versa. See the 'c' interactive command for additional information.

−**d** : **Delay time** interval as: **-d ss.tt** (*seconds.tenths*)
Specifies the delay between screen updates, and overrides the corresponding value in one's personal configuration file or the startup default. Later this can be changed with the 'd' or 's' interactive commands.

Fractional seconds are honored, but a negative number is not allowed. In all cases, however, such changes are prohibited if top is running in 'Secure mode', except for root (unless the 's' command−line option was used). For additional information on 'Secure mode' see topic 5a. SYSTEM Configuration File.

−**h** : **Help**
Show library version and the usage prompt, then quit.

−**H** : **Threads** toggle
Starts top with the last remembered 'H' state reversed. When this toggle is *On*, all individual threads will be displayed. Otherwise, top displays a summation of all threads in a process.

−**i** : **Idle Processes** toggle
Starts top with the last remembered 'i' state reversed. When this toggle is *Off*, tasks that are idled or zombied will not be displayed.

−**m** : **VIRT/USED** toggle
Reports USED (sum of process rss and swap total count) instead of VIRT

−**M** : **Detect memory units**
Show memory units (k/M/G) and display floating point values in the memory summary.

−**n** : **Number of iterations** limit as: **-n number**
Specifies the maximum number of iterations, or frames, top should produce before ending.

−**p** : **Monitor PIDs** as: **-pN1 -pN2 ...** or **-pN1, N2 [,...]**
Monitor only processes with specified process IDs. This option can be given up to 20 times, or you can provide a comma delimited list with up to 20 pids. Co-mingling both approaches is permitted.

This is a command−line option only. And should you wish to return to normal operation, it is not

necessary to quit and and restart top -- just issue the '=' interactive command.

−**s** : **Secure mode** operation
Starts top with secure mode forced, even for root. This mode is far better controlled through the system configuration file (see topic 5. FILES).

−**S** : **Cumulative time mode** toggle
Starts top with the last remembered 'S' state reversed. When 'Cumulative mode' is *On*, each process is listed with the cpu time that it and its dead children have used. See the 'S' interactive command for additional information regarding this mode.

−**u** : **Monitor by user** as: **-u somebody**
Monitor only processes with an effective UID or user name matching that given.

−**U** : **Monitor by user** as: **-U somebody**
Monitor only processes with a UID or user name matching that given. This matches real, effective, saved, and filesystem UIDs.

−**v** : **Version**
Show library version and the usage prompt, then quit.

2. FIELDS / Columns

2a. DESCRIPTIONS of Fields

Listed below are top's available fields. They are always associated with the letter shown, regardless of the position you may have established for them with the 'o' (Order fields) interactive command.

Any field is selectable as the sort field, and you control whether they are sorted high-to-low or low-to-high. For additional information on sort provisions see topic 3c. TASK Area Commands.

a: **PID** -- Process Id
The task's unique process ID, which periodically wraps, though never restarting at zero.

b: **PPID** -- Parent Process Pid
The process ID of a task's parent.

c: **RUSER** -- Real User Name
The real user name of the task's owner.

d: **UID** -- User Id
The effective user ID of the task's owner.

e: **USER** -- User Name
The effective user name of the task's owner.

f: **GROUP** -- Group Name
The effective group name of the task's owner.

g: **TTY** −− Controlling Tty
The name of the controlling terminal. This is usually the device (serial port, pty, etc.) from which the process was started, and which it uses for input or output. However, a task need not be associated with a terminal, in which case you'll see '?' displayed.

h: **PR** −− Priority
The priority of the task.

i: **NI** −− Nice value
The nice value of the task. A negative nice value means higher priority, whereas a positive nice value means lower priority. Zero in this field simply means priority will not be adjusted in determining a task's dispatchability.

j: **P** −− Last used CPU (SMP)
A number representing the last used processor. In a true SMP environment this will likely change frequently since the kernel intentionally uses weak affinity. Also, the very act of running top may break this weak affinity and cause more processes to change CPUs more often (because of the extra demand for cpu time).

k: **%CPU** −− CPU usage
The task's share of the elapsed CPU time since the last screen update, expressed as a percentage of total CPU time. In a true SMP environment, if 'Irix mode' is *Off*, top will operate in 'Solaris mode' where a task's cpu usage will be divided by the total number of CPUs. You toggle 'Irix/Solaris' modes with the 'I' interactive command.

l: **TIME** −− CPU Time
Total CPU time the task has used since it started. When 'Cumulative mode' is *On*, each process is listed with the cpu time that it and its dead children has used. You toggle 'Cumulative mode' with 'S', which is a command−line option and an interactive command. See the 'S' interactive command for additional information regarding this mode.

m: **TIME+** −− CPU Time, hundredths
The same as 'TIME', but reflecting more granularity through hundredths of a second.

n: **%MEM** −− Memory usage (RES)
A task's currently used share of available **physical** memory.

o: **VIRT** −− Virtual Image (kb)
The total amount of **virtual** memory used by the task. It includes all code, data and shared libraries plus pages that have been swapped out. (Note: you can define the STATSIZE=1 environment variable and the VIRT will be calculated from the /proc/#/state VmSize field.)

p: **SWAP** −− Swapped size (kb)
Per-process swap values are now taken from /proc/#/status VmSwap field.

q: **RES** −− Resident size (kb)
The non-swapped **physical** memory a task has used.

r: **CODE** −− Code size (kb)
The amount of **physical** memory devoted to executable code, also known as the 'text resident set' size or TRS.

s: **DATA** −− Data+Stack size (kb)
The amount of **physical** memory devoted to other than executable code, also known as the 'data resident set' size or DRS.

t: **SHR** −− Shared Mem size (kb)
The amount of **shared** memory used by a task. It simply reflects memory that could be potentially shared with other processes.

u: **nFLT** −− Page Fault count
The number of **major** page faults that have occurred for a task. A page fault occurs when a process attempts to read from or write to a virtual page that is not currently present in its address space. A major page fault is when disk access is involved in making that page available.

v: **nDRT** −− Dirty Pages count
The number of pages that have been modified since they were last written to disk. Dirty pages must be written to disk before the corresponding physical memory location can be used for some other virtual page.

w: **S** −− Process Status
The status of the task which can be one of:
'**D**' = uninterruptible sleep
'**R**' = running
'**S**' = sleeping
'**T**' = traced or stopped
'**Z**' = zombie

Tasks shown as running should be more properly thought of as 'ready to run' −− their task_struct is simply represented on the Linux run-queue. Even without a true SMP machine, you may see numerous tasks in this state depending on top's delay interval and nice value.

x: **Command** −− Command **line** or Program **name**
Display the command line used to start a task or the name of the associated program. You toggle between command *line* and *name* with 'c', which is both a command−line option and an interactive command.

When you've chosen to display command lines, processes without a command line (like kernel threads) will be shown with only the program name in parentheses, as in this example:
(mdrecoveryd)

Either form of display is subject to potential truncation if it's too long to fit in this field's current width. That width depends upon other fields selected, their order and the current screen width.

Note: The 'Command' field/column is unique, in that it is not fixed-width. When displayed, this column will be allocated all remaining screen width (up to the maximum 512 characters) to provide for the potential growth of program names into command lines.

y: **WCHAN** −− Sleeping in Function
Depending on the availability of the kernel link map ('System.map'), this field will show the name or the address of the kernel function in which the task is currently sleeping. Running tasks will display a dash ('-') in this column.

Note: By displaying this field, top's own working set will be increased by over 700Kb. Your only means of reducing that overhead will be to stop and restart top.

z: **Flags** −− Task Flags
This column represents the task's current scheduling flags which are expressed in hexadecimal notation and with zeros suppressed. These flags are officially documented in <linux/sched.h>. Less formal documentation can also be found on the 'Fields select' and 'Order fields' screens.

2b. SELECTING and ORDERING Columns

After pressing the interactive commands 'f' (Fields select) or ´o' (Order fields) you will be shown a screen containing the current **fields string** followed by names and descriptions for all fields.

Here is a sample **fields string** from one of top's four windows/field groups and an explanation of the conventions used:

- Sample fields string:
  ANOPQRSTUVXbcdefgjlmyzWHIK
- The order of displayed fields corresponds to the order of the letters in that string.
- If the letter is *upper case* the corresponding field itself will then be shown as part of the task display (screen width permitting). This will also be indicated by a leading asterisk ('*'), as in this excerpt:

  ...
  *** K: %CPU = CPU usage**
  l: TIME = CPU Time
  m: TIME+ = CPU Time, hundredths
  *** N: %MEM = Memory usage (RES)**
  *** O: VIRT = Virtual Image (kb)**
  ...

Fields select screen −− the 'f' interactive command
You *toggle* the **display** of a field by simply pressing the corresponding letter.

Order fields screen −− the 'o' interactive command
You *move* a field to the **left** by pressing the corresponding **upper case** letter and to the **right** with the **lower case** letter.

2c. SUMMARY Area Fields

The summary area fields describing CPU statistics are abbreviated. They provide information about times spent in:

us = user mode
sy = system mode
ni = low priority user mode (nice)
id = idle task
wa = I/O waiting
hi = servicing IRQs
si = servicing soft IRQs
st = steal (time given to other DomU instances)

3. INTERACTIVE Commands

Listed below is a brief index of commands within categories. Some commands appear more than once −− their meaning or scope may vary depending on the context in which they are issued.

3a. *GLOBAL_Commands*
<Ret/Sp> ?, =, A, B, d, G, h, I, k, q, r, s, W, Z
3b. *SUMMARY_Area_Commands*
l, m, t, 1
3c. *TASK_Area_Commands*
Appearance: b, x, y, z
Content: c, f, H, o, S, u
Size: #, i, n
Sorting: <, >, F, O, R
3d. *COLOR_Mapping*
<Ret>, a, B, b, H, M, q, S, T, w, z, 0 - 7
4b. *COMMANDS_for_Windows*
-, _, =, +, A, a, G, g, w

3a. GLOBAL Commands

The global interactive commands are **always** available in both full−screen mode and alternate−display mode. However, some of these interactive commands are **not available** when running in 'Secure mode'.

If you wish to know in advance whether or not your top has been secured, simply ask for help and view the system summary on the second line.

<**Enter**> or <**Space**> :*Refresh_Display*
These commands do nothing, they are simply ignored. However, they will awaken top and following receipt of any input the entire display will be repainted.

Use either of these keys if you have a large delay interval and wish to see current status,

´**?**´ or ´**h**´ :*Help*
There are two help levels available. The first will provide a reminder of all the basic interactive commands. If top is *secured*, that screen will be abbreviated.

Typing 'h' or '?' on that help screen will take you to help for those interactive commands applicable to alternate−display mode.

´**=**´ :*Exit_Task_Limits*
Removes restrictions on which tasks are shown. This command will reverse any 'i' (idle tasks) and 'n' (max tasks) commands that might be active. It also provides for an 'exit' from PID monitoring. See the '-p' command−line option for a discussion of PID monitoring.

When operating in alternate−display mode this command has a slightly broader meaning.

´**A**´ :*Alternate_Display_Mode_toggle*
This command will switch between full−screen mode and alternate−display mode. See topic 4. ALTERNATE−DISPLAY Mode and the 'G' interactive command for insight into ´current' windows and field groups.

´**B**´ :*Bold_Disable/Enable_toggle*
This command will influence use of the 'bold' terminfo capability and alters **both** the summary area and task area for the ´current' window. While it is intended primarily for use with dumb terminals, it can be applied anytime.

Note: When this toggle is *On* and top is operating in monochrome mode, the **entire display** will appear as normal text. Thus, unless the 'x' and/or 'y' toggles are using reverse for emphasis, there will be no visual confirmation that they are even on.

* ´**d**´ or ´**s**´ :*Change_Delay_Time_interval*
You will be prompted to enter the delay time, in seconds, between display updates.

Fractional seconds are honored, but a negative number is not allowed. Entering 0 causes (nearly) continuous updates, with an unsatisfactory display as the system and tty driver try to keep up with top's demands. The delay value is inversely proportional to system loading, so set it with care.

If at any time you wish to know the current delay time, simply ask for help and view the system summary on the second line.

´**G**´ :*Choose_Another_Window/Field_Group*
You will be prompted to enter a number between 1 and 4 designating the window/field group which should be made the ´current' window. You will soon grow comfortable with these 4 windows, especially after experimenting with alternate−display mode.

´**I**´ :*Irix/Solaris_Mode_toggle*
When operating in 'Solaris mode' ('I' toggled *Off*), a task's cpu usage will be divided by the total number of CPUs. After issuing this command, you'll be informed of the new state of this toggle.

´**u**´ :*select a user*
You will be prompted for a UID or username. Only processes belonging to the selected user will be displayed. This option matches on the effective UID.

´**U**´ :*select a user*
You will be prompted for a UID or username. Only processes belonging to the selected user will be displayed. This option matches on the real, effective, saved, and filesystem UID.

* ´**k**´ :*Kill_a_task*
You will be prompted for a PID and then the signal to send. The default signal, as reflected in the prompt, is SIGTERM. However, you can send any signal, via number or name.

If you wish to abort the kill process, do one of the following depending on your progress:
1) at the pid prompt, just press <Enter>
2) at the signal prompt, type 0

´**q**´ :*Quit*

* ´**r**´ :*Renice_a_Task*
You will be prompted for a PID and then the value to nice it to. Entering a positive value will cause a process to lose priority. Conversely, a negative value will cause a process to be viewed more favorably by the kernel.

´**W**´ :*Write_the_Configuration_File*
This will save all of your options and toggles plus the current display mode and delay time. By issuing this command just before quitting top, you will be able restart later in exactly that same state.

´**Z**´ :*Change_Color_Mapping*
This key will take you to a separate screen where you can change the colors for the ´current' window, or for all windows. For details regarding this interactive command see topic 3d. COLOR Mapping.

* The commands shown with an asterisk ('*') are not available in 'Secure mode', nor will they be shown on the level-1 help screen.

3b. SUMMARY Area Commands

The summary area interactive commands are **always available** in both full−screen mode and alternate−display mode. They affect the beginning lines of your display and will determine the position of messages and prompts.

These commands always impact just the ´current' window/field group. See topic 4. ALTERNATE−DISPLAY Mode and the 'G' interactive command for insight into ´current' windows and field groups.

´**l**´ :*Toggle_Load_Average/Uptime* −− On/Off
This is also the line containing the program name (possibly an alias) when operating in full−screen mode or the ´current' window name when operating in alternate−display mode.

´**m**´ :*Toggle_Memory/Swap_Usage* −− On/Off
This command affects two summary area lines.

´**t**´ :*Toggle_Task/Cpu_States* −− On/Off
This command affects from 2 to many summary area lines, depending on the state of the '1' toggle and whether or not top is running under true SMP.

´**1**´ :*Toggle_Single/Separate_Cpu_States* −− On/Off
This command affects how the 't' command's Cpu States portion is shown. Although this toggle exists primarily to serve massively-parallel SMP machines, it is not restricted to solely SMP environments.

When you see 'Cpu(s):' in the summary area, the '1' toggle is *On* and all cpu information is gathered in a single line. Otherwise, each cpu is displayed separately as: 'Cpu0, Cpu1, ...'

Note: If the entire summary area has been toggled *Off* for any window, you would be left with just the **message line**. In that way, you will have maximized available task rows but (temporarily) sacrificed the program name in full−screen mode or the ´current' window name when in alternate−display mode.

3c. TASK Area Commands

The task area interactive commands are **always** available in full−screen mode.

The task area interactive commands are **never available** in alternate−display mode *if* the ´current' window's task display has been toggled *Off* (see topic 4. ALTERNATE−DISPLAY Mode).

APPEARANCE of task window

The following commands will also be influenced by the state of the global 'B' (bold disable) toggle.

´**b**´ :*Bold/Reverse_toggle*

This command will impact how the 'x' and 'y' toggles are displayed. Further, it will only be available when at least one of those toggles is *On*.

´**x**´ :*Column_Highlight_toggle*

Changes highlighting for the current sort field. You probably don't need a constant visual reminder of the sort field and top hopes that you always run with 'column highlight' *Off*, due to the cost in path-length.

If you forget which field is being sorted this command can serve as a quick visual reminder.

´**y**´ :*Row_Highlight_toggle*

Changes highlighting for "running" tasks. For additional insight into this task state, see topic 2a. DESCRIPTIONS of Fields, Process Status.

Use of this provision provides important insight into your system's health. The only costs will be a few additional tty escape sequences.

´**z**´ :*Color/Monochrome_toggle*

Switches the ´current' window between your last used color scheme and the older form of black-on-white or white-on-black. This command will alter **both** the summary area and task area but does not affect the state of the 'x', 'y' or 'b' toggles.

CONTENT of task window

´**c**´ :*Command_Line/Program_Name_toggle*

This command will be honored whether or not the 'Command' column is currently visible. Later, should that field come into view, the change you applied will be seen.

´**f**´ and ´**o**´ :*Fields_select* or *Order_fields*

These keys display separate screens where you can change which fields are displayed and their order. For additional information on these interactive commands see topic 2b. SELECTING and ORDERING Columns.

´**H**´ :*Threads_toggle*

When this toggle is *On*, all individual threads will be displayed. Otherwise, top displays a summation of all threads in a process.

´**S**´ :*Cumulative_Time_Mode_toggle*

When 'Cumulative mode' is *On*, each process is listed with the cpu time that it and its dead children have used.

When *Off*, programs that fork into many separate tasks will appear less demanding. For programs like 'init' or a shell this is appropriate but for others, like compilers, perhaps not. Experiment with two task windows sharing the same sort field but with different 'S' states and see which representation you prefer.

After issuing this command, you'll be informed of the new state of this toggle. If you wish to know in advance whether or not 'Cumulative mode' is in effect, simply ask for help and view the window summary on the second line.

´**u**´ :*Show_Specific_User_Only*
You will be prompted to enter the name of the user to display. Thereafter, in that task window only matching User ID's will be shown, or possibly no tasks will be shown.

Later, if you wish to monitor all tasks again, re-issue this command but just press <Enter> at the prompt, without providing a name.

SIZE of task window
´**i**´ :*Idle_Processes_toggle*
Displays all tasks or just active tasks. When this toggle is *Off*, idled or zombied processes will not be displayed.

If this command is applied to the last task display when in alternate−display mode, then it will not affect the window's size, as all prior task displays will have already been painted.

´**n**´ or ´#´ :*Set_Maximum_Tasks*
You will be prompted to enter the number of tasks to display. The lessor of your number and available screen rows will be used.

When used in alternate−display mode, this is the command that gives you precise control over the size of each currently visible task display, except for the very last. It will not affect the last window's size, as all prior task displays will have already been painted.

Note: If you wish to increase the size of the last visible task display when in alternate−display mode, simply decrease the size of the task display(s) above it.

SORTING of task window
For compatibility, this top supports most of the former top sort keys. Since this is primarily a service to former top users, these commands do not appear on any help screen.

command	sorted field	supported
A	start time (non-display)	No
M	%MEM	Yes
N	PID	Yes
P	%CPU	Yes
T	TIME+	Yes

Before using any of the following sort provisions, top suggests that you temporarily turn on column highlighting using the 'x' interactive command. That will help ensure that the actual sort environment matches your intent.

The following interactive commands will **only** be honored when the current sort field is **visible**. The sort field might *not* be visible because:
1) there is insufficient *Screen Width*
2) the 'f' interactive command turned it *Off*

´<´ :*Move_Sort_Field_Left*
Moves the sort column to the left unless the current sort field is the first field being displayed.

´>´ :*Move_Sort_Field_Right*
Moves the sort column to the right unless the current sort field is the last field being displayed.

The following interactive commands will **always** be honored whether or not the current sort field is visible.

´**F**´ or ´**O**´ :*Select_Sort_Field*
These keys display a separate screen where you can change which field is used as the sort column.

If a field is selected which was not previously being displayed, it will be forced *On* when you return to the top display. However, depending upon your screen width and the order of your fields, this sort field may not be displayable.

This interactive command can be a convenient way to simply verify the current sort field, when running top with column highlighting turned *Off*.

´**R**´ :*Reverse/Normal_Sort_Field_toggle*
Using this interactive command you can alternate between high-to-low and low-to-high sorts.

Note: Field sorting uses internal values, not those in column display. Thus, the TTY and WCHAN fields will violate strict ASCII collating sequence.

3d. COLOR Mapping

When you issue the 'Z' interactive command, you will be presented with a separate screen. That screen can be used to change the colors in just the ´current' window or in all four windows before returning to the top display.

Available interactive commands
4 upper case letters to select a **target**
8 numbers to select a **color**
normal toggles available
'B' :bold disable/enable
'b' :running tasks "bold"/reverse
'z' :color/mono
other commands available
'a'/'w' :apply, then go to next/prior
<Enter> :apply and exit
'q' :abandon current changes and exit

If your use 'a' or 'w' to cycle the targeted window, you will have applied the color scheme that was displayed when you left that window. You can, of course, easily return to any window and reapply different colors or turn colors *Off* completely with the 'z' toggle.

The Color Mapping screen can also be used to change the ´current' window/field group in either full−screen mode or alternate−display mode. Whatever was targeted when 'q' or <Enter> was pressed will be made current as you return to the top display.

4. ALTERNATE−DISPLAY Mode

4a. WINDOWS Overview

Field Groups/Windows:
In full−screen mode there is a single window represented by the entire screen. That single window can still be changed to display 1 of 4 different **field groups** (see the 'G' interactive command, repeated below). Each of the 4 field groups has a unique separately configurable **summary area** and its own configurable **task area**.

In alternate−display mode, those 4 underlying field groups can now be made visible simultaneously, or can be turned *Off* individually at your command.

The summary area will always exist, even if it's only the message line. At any given time only *one*

summary area can be displayed. However, depending on your commands, there could be from *zero* to *four* separate task displays currently showing on the screen.

Current Window:

The ´current' window is the window associated with the summary area and the window to which task related commands are always directed. Since in alternate−display mode you can toggle the task display *Off*, some commands might be restricted for the ´current' window.

A further complication arises when you have toggled the first summary area line *Off*. With the loss of the window name (the 'l' toggled line), you'll not easily know what window is the ´current' window.

4b. COMMANDS for Windows

´**-**´ and ´**_**´ :*Show/Hide_Window(s)_toggles*

The '-' key turns the ´current' window's task display *On* and *Off*. When *On*, that task area will show a minimum of the columns header you've established with the 'f' and 'o' commands. It will also reflect any other task area options/toggles you've applied yielding zero or more tasks.

The '_' key does the same for all task displays. In other words, it switches between the currently visible task display(s) and any task display(s) you had toggled *Off*. If all 4 task displays are currently visible, this interactive command will leave the summary area as the only display element.

* ´**=**´ and ´**+**´ :*Equalize_(re-balance)_Window(s)*

The '=' key forces the ´current' window's task display to be visible. It also reverses any 'i' (idle tasks) and 'n' (max tasks) commands that might be active.

The '+' key does the same for all windows. The four task displays will reappear, evenly balanced. They will also have retained any customizations you had previously applied, except for the 'i' (idle tasks) and 'n' (max tasks) commands.

* ´**A**´ :*Alternate_Display_Mode_toggle*

This command will switch between full−screen mode and alternate−display mode.

The first time you issue this command, all four task displays will be shown. Thereafter when you switch modes, you will see only the task display(s) you've chosen to make visible.

* ´**a**´ and ´**w**´ :*Next_Window_Forward/Backward*

This will change the ´current' window, which in turn changes the window to which commands are directed. These keys act in a circular fashion so you can reach any desired ´current' window using either key.

Assuming the window name is visible (you have not toggled 'l' *Off*), whenever the ´current' window name loses its emphasis/color, that's a reminder the task display is *Off* and many commands will be restricted.

* ´**G**´ :*Choose_Another_Window/Field_Group*

You will be prompted to enter a number between 1 and 4 designating the window/field group which should be made the ´current' window.

In full−screen mode, this command is necessary to alter the ´current' window. In alternate−display mode, it is simply a less convenient alternative to the 'a' and 'w' commands.

´**g**´ :*Change_Window/Field_Group_Name*
You will be prompted for a new name to be applied to the ´current' window. It does not require that the window name be visible (the 'l' toggle to be *On*).

* The interactive commands shown with an asterisk ('*') have use beyond alternate−display mode.
´=', 'A', 'G' are always available
´a', 'w' act the same when color mapping

5. FILES

5a. SYSTEM Configuration File

The presence of this file will influence which version of the 'help' screen is shown to an ordinary user. More importantly, it will limit what ordinary users are allowed to do when top is running. They will not be able to issue the following commands.

```
k       Kill a task
r       Renice a task
d or s  Change delay/sleep interval
```

The system configuration file is **not** created by top. Rather, you create this file manually and place it in the */etc* directory. Its name must be 'toprc' and must have no leading '.' (period). It must have only two lines.

Here is an example of the contents of */etc/toprc*:

```
s       # line 1: 'secure' mode switch
5.0     # line 2: 'delay'  interval in seconds
```

5b. PERSONAL Configuration File

This file is written as '$HOME/.your-name-4-top' + 'rc'. Use the 'W' interactive command to create it or update it.

Here is the general layout:

```
global   # line 1: the program name/alias notation
  "      # line 2: id,altscr,irixps,delay,curwin
per ea   # line a: winname,fieldscur
window   # line b: winflags,sortindx,maxtasks
  "      # line c: summclr,msgsclr,headclr,taskclr
```

If the $HOME variable is not present, top will try to write the personal configuration file to the current directory, subject to permissions.

6. STUPID TRICKS Sampler

Many of these 'tricks' work best when you give top a scheduling boost. So plan on starting him with a nice value of -10, assuming you've got the authority.

6a. Kernel Magic

For these stupid tricks, top needs full−screen mode.

-*- The user interface, through prompts and help, intentionally implies that the delay interval is limited to tenths of a second. However, you're free to set any desired delay. If you want to see Linux at his scheduling best, try a delay of .09 seconds or less.

For this experiment, under x-windows open an xterm and maximize it. Then do the following:
. provide a scheduling boost and tiny delay via:
nice -n -10 top -d.09

. keep sorted column highlighting *Off* to minimize
path length
. turn *On* reverse row highlighting for emphasis
. try various sort columns (TIME/MEM work well),
and normal or reverse sorts to bring the most
active processes into view

What you'll see is a very busy Linux doing what he's always done for you, but there was no program available to illustrate this.

-*- Under an xterm using 'white-on-black' colors, try setting top's task color to black and be sure that task highlighting is set to bold, not reverse. Then set the delay interval to around .3 seconds.

After bringing the most active processes into view, what you'll see are the ghostly images of just the currently running tasks.

-*- Delete the existing rcfile, or create a new symlink. Start this new version then type 'T' (a secret key, see topic 3c. TASK Area Commands, Sorting) followed by 'W' and 'q'. Finally, restart the program with -d0 (zero delay).

Your display will be refreshed at three times the rate of the former top, a 300% speed advantage. As top climbs the TIME ladder, be as patient as you can while speculating on whether or not top will ever reach the top.

6b. Bouncing Windows

For these stupid tricks, top needs alternate−display mode.

-*- With 3 or 4 task displays visible, pick any window other than the last and turn idle processes *Off*. Depending on where you applied 'i', sometimes several task displays are bouncing and sometimes it's like an accordion, as top tries his best to allocate space.

-*- Set each window's summary lines differently: one with no memory; another with no states; maybe one with nothing at all, just the message line. Then hold down 'a' or 'w' and watch a variation on bouncing windows −− hopping windows.

-*- Display all 4 windows and for each, in turn, set idle processes to *Off*. You've just entered the "extreme bounce" zone.

6c. The Big Bird Window

This stupid trick also requires alternate−display mode.

-*- Display all 4 windows and make sure that 1:Def is the ´current' window. Then, keep increasing window size until the all the other task displays are "pushed out of the nest".

When they've all been displaced, toggle between all visible/invisible windows. Then ponder this:
is top fibbing or telling honestly your imposed truth?

7. BUGS

Send bug reports to:
Albert D. Cahalan, <albert@users.sf.net>

The top command calculates Cpu(s) by looking at the change in CPU time values between samples. When you first run it, it has no previous sample to compare to, so these initial values are the percentages since

boot. It means you need at least two loops or you have to ignore summary output from the first loop. This is problem for example for batch mode. There is a possible workaround if you define the CPULOOP=1 environment variable. The top command will be run one extra hidden loop for CPU data before standard output.

8. HISTORY Former top

The original top was written by Roger Binns, based on Branko Lankester's <lankeste@fwi.uva.nl> ps program.

Robert Nation <nation@rocket.sanders.lockheed.com> adapted it for the proc file system.

Helmut Geyer <Helmut.Geyer@iwr.uni-heidelberg.de> added support for configurable fields.

Plus many other individuals contributed over the years.

9. AUTHOR

This entirely new and enhanced replacement was written by:
Jim / James C. Warner, <warnerjc@worldnet.att.net>

With invaluable help from:
Albert D. Cahalan, <albert@users.sf.net>
Craig Small, <csmall@small.dropbear.id.au>

10. SEE ALSO

free(1), **ps**(1), **uptime**(1), **atop**(1), **slabtop**(1), **vmstat**(8), **w**(1).