

NAME

atop - Advanced System & Process Monitor

SYNOPSIS

Interactive Usage:

```
atop [-g|-m|-d|-n|-u|-p|-s|-c|-v|-o|-y] [-C|-M|-D|-N|-A] [-afFG1xR] [-L linelen] [-Plabel[,label]...]
[ interval [ samples ]]
```

Writing and reading raw logfiles:

```
atop -w rawfile [-a] [-S] [ interval [ samples ]]
atop -r [ rawfile ] [-b hh:mm ] [-e hh:mm ] [-g|-m|-d|-n|-u|-p|-s|-c|-v|-o|-y] [-C|-M|-D|-N|-A]
[-fFG1xR] [-L linelen] [-Plabel[,label]...]
```

DESCRIPTION

The program *atop* is an interactive monitor to view the load on a Linux system. It shows the occupation of the most critical hardware resources (from a performance point of view) on system level, i.e. cpu, memory, disk and network.

It also shows which processes are responsible for the indicated load with respect to cpu and memory load on process level. Disk load is shown per process if "storage accounting" is active in the kernel. Network load is shown per process if the kernel module 'netatop' has been installed.

Every *interval* (default: 10 seconds) information is shown about the resource occupation on system level (cpu, memory, disks and network layers), followed by a list of processes which have been active during the last interval (note that all processes that were unchanged during the last interval are not shown, unless the key 'a' has been pressed). If the list of active processes does not entirely fit on the screen, only the top of the list is shown (sorted in order of activity).

The intervals are repeated till the number of *samples* (specified as command argument) is reached, or till the key 'q' is pressed in interactive mode.

When *atop* is started, it checks whether the standard output channel is connected to a screen, or to a file/pipe. In the first case it produces screen control codes (via the ncurses library) and behaves interactively; in the second case it produces flat ASCII-output.

In interactive mode, the output of *atop* scales dynamically to the current dimensions of the screen/window. If the window is resized horizontally, columns will be added or removed automatically. For this purpose, every column has a particular weight. The columns with the highest weights that fit within the current width will be shown.

If the window is resized vertically, lines of the process/thread list will be added or removed automatically.

Furthermore in interactive mode the output of *atop* can be controlled by pressing particular keys. However it is also possible to specify such key as **flag** on the command line. In that case *atop* switches to the indicated mode on beforehand; this mode can be modified again interactively. Specifying such key as flag is especially useful when running *atop* with output to a pipe or file (non-interactively). These flags are the same as the keys that can be pressed in interactive mode (see section INTERACTIVE COMMANDS).

Additional flags are available to support storage of atop-data in raw format (see section RAW DATA STORAGE).

PROCESS ACCOUNTING

With every interval, *atop* reads the kernel administration to obtain information about all running processes. However, it is likely that during the interval also processes have terminated. These processes might have consumed system resources during this interval as well before they terminated. Therefore, *atop* tries to read the process accounting records that contain the accounting information of terminated processes and report these processes too. Only when the process accounting mechanism in the kernel is activated, the kernel writes such process accounting record to a file for every process that terminates.

There are various ways for *atop* to get access to the process accounting records (tried in this order):

1. When the environment variable ATOPACCT is set, it specifies the name of the process accounting file. In that case, process accounting for this file should have been activated on beforehand. Before opening this file for reading, *atop* drops its root privileges (if any).

When this environment variable is present but its contents is empty, process accounting will not be used at all.

2. **This is the preferred way of handling process accounting records!**

When the *atopacctd* daemon is active, it has activated the process accounting mechanism in the kernel and transfers to original accounting records to shadow files. In that case, *atop* drops its root privileges and opens the current shadow file for reading.

This way is preferred, because the *atopacctd* daemon maintains full control of the sizes of the original process accounting file (written by the kernel) and the shadow files (read by the *atop* processes). For further information, refer to the **atopacctd** man page.

3. When the *atopacctd* daemon is not active, *atop* verifies if the process accounting mechanism has been switched on via the separate **psacct** package. In that case, the file **/var/account/pacct** is in use as process accounting file and *atop* opens this file for reading.

4. As a last possibility, *atop* itself tries to activate the process accounting mechanism (requires root privileges) using the file **/tmp/atop.d/atop.acct** (to be written by the kernel, to be read by *atop* itself). Process accounting remains active as long as at least one *atop* process is alive. Whenever the last *atop* process stops (either by pressing 'q' or by 'kill -15'), it deactivates the process accounting mechanism again. Therefore you should never terminate *atop* by 'kill -9', because then it has no chance to stop process accounting. As a result, the accounting file may consume a lot of disk space after a while.

To avoid that the process accounting file consumes too much disk space, *atop* verifies at the end of every sample if the size of the process accounting file exceeds 200 MiB and if this *atop* process is the only one that is currently using the file. In that case the file is truncated to a size of zero.

Notice that root-privileges are required to switch on/off process accounting in the kernel. You can start *atop* as a root user or specify *setuid-root* privileges to the executable file. In the latter case, *atop* switches on process accounting and drops the root-privileges again.

If *atop* does not run with root-privileges, it does not show information about finished processes. It indicates this situation with the message 'no procacct' in the top-right corner (instead of the counter that shows the number of exited processes).

When during one interval a lot of processes have finished, *atop* might grow tremendously in memory when reading all process accounting records at the end of the interval. To avoid such excessive growth, *atop* will never read more than 50 MiB with process information from the process accounting file per interval (approx. 70000 finished processes). In interactive mode a warning is given whenever processes have been skipped for this reason.

COLORS

For the resource consumption on system level, *atop* uses colors to indicate that a critical occupation percentage has been (almost) reached. A critical occupation percentage means that is likely that this load causes a noticeable negative performance influence for applications using this resource. The critical percentage depends on the type of resource: e.g. the performance influence of a disk with a busy percentage of 80% might be more noticeable for applications/user than a CPU with a busy percentage of 90%.

Currently *atop* uses the following default values to calculate a weighted percentage per resource:

Processor

A busy percentage of 90% or higher is considered 'critical'.

Disk

A busy percentage of 70% or higher is considered 'critical'.

Network

A busy percentage of 90% or higher for the load of an interface is considered 'critical'.

Memory

An occupation percentage of 90% is considered 'critical'. Notice that this occupation percentage is the accumulated memory consumption of the kernel (including slab) and all processes; the memory for the page cache ('cache' and 'buff' in the MEM-line) and the reclaimable part of the slab ('slrec') is not implied!

If the number of pages swapped out ('swout' in the PAG-line) is larger than 10 per second, the memory resource is considered 'critical'. A value of at least 1 per second is considered 'almost critical'. If the committed virtual memory exceeds the limit ('vmcom' and 'vmlim' in the SWP-line), the SWP-line is colored due to overcommitting the system.

Swap

An occupation percentage of 80% is considered 'critical' because swap space might be completely exhausted in the near future; it is not critical from a performance point-of-view.

These default values can be modified in the configuration file (see separate man-page of atoprc).

When a resource exceeds its critical occupation percentage, the concerning values in the screen line are colored red by default.

When a resource exceeded (default) 80% of its critical percentage (so it is almost critical), the concerning values in the screen line are colored cyan by default. This 'almost critical percentage' (one value for all resources) can be modified in the configuration file (see separate man-page of atoprc).

The default colors red and cyan can be modified in the configuration file as well (see separate man-page of atoprc).

With the key 'x' (or flag -x), the use of colors can be suppressed.

NETATOP MODULE

Per-process and per-thread network activity can be measured by the *netatop* kernel module. You can download this kernel module from the website (mentioned at the end of this manual page) and install it on your system if the kernel version is 2.6.24 or newer.

When *atop* gathers counters for a new interval, it verifies if the *netatop* module is currently active. If so, *atop* obtains the relevant network counters from this module and shows the number of sent and received packets per process/thread in the generic screen. Besides, detailed counters can be requested by pressing the 'n' key.

When the *netatopd* daemon is running as well, *atop* also reads the network counters of exited processes that are logged by this daemon (comparable with process accounting).

More information about the optional *netatop* kernel module and the *netatopd* daemon can be found in the concerning man-pages and on the website mentioned at the end of this manual page.

INTERACTIVE COMMANDS

When running *atop* interactively (no output redirection), keys can be pressed to control the output. In general, lower case keys can be used to show other information for the active processes and upper case keys can be used to influence the sort order of the active process/thread list.

g Show generic output (default).

Per process the following fields are shown in case of a window-width of 80 positions: process-id, cpu consumption during the last interval in system and user mode, the virtual and resident memory growth of the process.

The subsequent columns depend on the used kernel:

When the kernel supports "storage accounting" ($\geq 2.6.20$), the data transfer for read/write on disk, the status and exit code are shown for each process. When the kernel does not support "storage accounting", the username, number of threads in the thread group, the status and exit code are shown. When the kernel module 'netatop' is loaded, the data transfer for send/receive of network packets is shown for each process.

The last columns contain the state, the occupation percentage for the chosen resource (default: cpu) and the process name.

When more than 80 positions are available, other information is added.

m Show memory related output.

Per process the following fields are shown in case of a window-width of 80 positions: process-id,

minor and major memory faults, size of virtual shared text, total virtual process size, total resident process size, virtual and resident growth during last interval, memory occupation percentage and process name.

When more than 80 positions are available, other information is added.

d Show disk-related output.

When "storage accounting" is active in the kernel, the following fields are shown: process-id, amount of data read from disk, amount of data written to disk, amount of data that was written but has been withdrawn again (WCANCL), disk occupation percentage and process name.

n Show network related output.

Per process the following fields are shown in case of a window-width of 80 positions: process-id, thread-id, total bandwidth for received packets, total bandwidth for sent packets, number of received TCP packets with the average size per packet (in bytes), number of sent TCP packets with the average size per packet (in bytes), number of received UDP packets with the average size per packet (in bytes), number of sent UDP packets with the average size per packet (in bytes), the network occupation percentage and process name.

This information can only be shown when kernel module 'netatop' is installed.

When more than 80 positions are available, other information is added.

s Show scheduling characteristics.

Per process the following fields are shown in case of a window-width of 80 positions: process-id, number of threads in state 'running' (R), number of threads in state 'interruptible sleeping' (S), number of threads in state 'uninterruptible sleeping' (D), scheduling policy (normal timesharing, realtime round-robin, realtime fifo), nice value, priority, realtime priority, current processor, status, exit code, state, the occupation percentage for the chosen resource and the process name.

When more than 80 positions are available, other information is added.

v Show various process characteristics.

Per process the following fields are shown in case of a window-width of 80 positions: process-id, user name and group, start date and time, status (e.g. exit code if the process has finished), state, the occupation percentage for the chosen resource and the process name.

When more than 80 positions are available, other information is added.

c Show the command line of the process.

Per process the following fields are shown: process-id, the occupation percentage for the chosen resource and the command line including arguments.

o Show the user-defined line of the process.

In the configuration file the keyword *ownprocline* can be specified with the description of a user-defined output-line.

Refer to the man-page of **atoprc** for a detailed description.

y Show the individual threads within a process (toggle).

Single-threaded processes are still shown as one line.

For multi-threaded processes, one line represents the process while additional lines show the activity per individual thread (in a different color). Depending on the option 'a' (all or active toggle), all threads are shown or only the threads that were active during the last interval.

Whether this key is active or not can be seen in the header line.

- u** Show the process activity accumulated per user.

Per user the following fields are shown: number of processes active or terminated during last interval (or in total if combined with command ‘a’), accumulated cpu consumption during last interval in system and user mode, the current virtual and resident memory space consumed by active processes (or all processes of the user if combined with command ‘a’).

When "storage accounting" is active in the kernel, the accumulated read and write throughput on disk is shown. When the kernel module ‘netatop’ has been installed, the number of received and sent network packets are shown.

The last columns contain the accumulated occupation percentage for the chosen resource (default: cpu) and the user name.

- p** Show the process activity accumulated per program (i.e. process name).

Per program the following fields are shown: number of processes active or terminated during last interval (or in total if combined with command ‘a’), accumulated cpu consumption during last interval in system and user mode, the current virtual and resident memory space consumed by active processes (or all processes of the user if combined with command ‘a’).

When "storage accounting" is active in the kernel, the accumulated read and write throughput on disk is shown. When the kernel module ‘netatop’ has been installed, the number of received and sent network packets are shown.

The last columns contain the accumulated occupation percentage for the chosen resource (default: cpu) and the program name.

- C** Sort the current list in the order of cpu consumption (default). The one-but-last column changes to “CPU”.
- M** Sort the current list in the order of resident memory consumption. The one-but-last column changes to “MEM”.
- D** Sort the current list in the order of disk accesses issued. The one-but-last column changes to “DSK”.
- N** Sort the current list in the order of network bandwidth (received and transmitted). The one-but-last column changes to “NET”.
- A** Sort the current list automatically in the order of the most busy system resource during this interval. The one-but-last column shows either “ACPU”, “AMEM”, “ADSK” or “ANET” (the preceding ‘A’ indicates automatic sorting-order). The most busy resource is determined by comparing the weighted busy-percentages of the system resources, as described earlier in the section COLORS. This option remains valid until another sorting-order is explicitly selected again. A sorting-order for disk is only possible when "storage accounting" is active. A sorting-order for network is only possible when the kernel module ‘netatop’ is loaded.

Miscellaneous interactive commands:

- ?** Request for help information (also the key ‘h’ can be pressed).
- V** Request for version information (version number and date).
- R** Gather and calculate the proportional set size of processes (toggle). Gathering of all values that are needed to calculate the PSIZE of a process is a relatively time-consuming task, so this key should only be active when analyzing the resident memory consumption of processes.
- x** Suppress colors to highlight critical resources (toggle). Whether this key is active or not can be seen in the header line.
- z** The pause key can be used to freeze the current situation in order to investigate the output on the screen. While *atop* is paused, the keys described above can be pressed to show other information about the current list of processes. Whenever the pause key is pressed again, *atop* will continue with a next sample.

- i** Modify the interval timer (default: 10 seconds). If an interval timer of 0 is entered, the interval timer is switched off. In that case a new sample can only be triggered manually by pressing the key 't'.
- t** Trigger a new sample manually. This key can be pressed if the current sample should be finished before the timer has exceeded, or if no timer is set at all (interval timer defined as 0). In the latter case *atop* can be used as a stopwatch to measure the load being caused by a particular application transaction, without knowing on beforehand how many seconds this transaction will last.

When viewing the contents of a raw file, this key can be used to show the next sample from the file.
- T** When viewing the contents of a raw file, this key can be used to show the previous sample from the file.
- b** When viewing the contents of a raw file, this key can be used to branch to a certain timestamp within the file (either forward or backward).
- r** Reset all counters to zero to see the system and process activity since boot again.

When viewing the contents of a raw file, this key can be used to rewind to the beginning of the file again.
- U** Specify a search string for specific user names as a regular expression. From now on, only (active) processes will be shown from a user which matches the regular expression. The system statistics are still system wide. If the Enter-key is pressed without specifying a name, (active) processes of all users will be shown again.
Whether this key is active or not can be seen in the header line.
- I** Specify a list with one or more PIDs to be selected. From now on, only processes will be shown with a PID which matches one of the given list. The system statistics are still system wide. If the Enter-key is pressed without specifying a PID, all (active) processes will be shown again.
Whether this key is active or not can be seen in the header line.
- P** Specify a search string for specific process names as a regular expression. From now on, only processes will be shown with a name which matches the regular expression. The system statistics are still system wide. If the Enter-key is pressed without specifying a name, all (active) processes will be shown again.
Whether this key is active or not can be seen in the header line.
- /** Specify a specific command line search string as a regular expression. From now on, only processes will be shown with a command line which matches the regular expression. The system statistics are still system wide. If the Enter-key is pressed without specifying a string, all (active) processes will be shown again.
Whether this key is active or not can be seen in the header line.
- S** Specify search strings for specific logical volume names, specific disk names and specific network interface names. All search strings are interpreted as a regular expressions. From now on, only those system resources are shown that match the concerning regular expression. If the Enter-key is pressed without specifying a search string, all (active) system resources of that type will be shown again.
Whether this key is active or not can be seen in the header line.
- a** The 'all/active' key can be used to toggle between only showing/accumulating the processes that were active during the last interval (default) or showing/accumulating all processes.
Whether this key is active or not can be seen in the header line.
- G** By default, *atop* shows/accumulates the processes that are alive and the processes that are exited during the last interval. With this key (toggle), showing/accumulating the processes that are exited can be suppressed.
Whether this key is active or not can be seen in the header line.
- f** Show a fixed (maximum) number of header lines for system resources (toggle). By default only the lines are shown about system resources (CPUs, paging, logical volumes, disks, network interfaces) that really have been active during the last interval. With this key you can force *atop* to show lines of

inactive resources as well.

Whether this key is active or not can be seen in the header line.

- F** Suppress sorting of system resources (toggle). By default system resources (CPUs, logical volumes, disks, network interfaces) are sorted on utilization.
Whether this key is active or not can be seen in the header line.
- 1** Show relevant counters as an average per second (in the format ‘.../s’) instead of as a total during the interval (toggle).
Whether this key is active or not can be seen in the header line.
- l** Limit the number of system level lines for the counters per-cpu, the active disks and the network interfaces. By default lines are shown of all CPUs, disks and network interfaces which have been active during the last interval. Limiting these lines can be useful on systems with huge number CPUs, disks or interfaces in order to be able to run *atop* on a screen/window with e.g. only 24 lines. For all mentioned resources the maximum number of lines can be specified interactively. When using the flag **-l** the maximum number of per-cpu lines is set to 0, the maximum number of disk lines to 5 and the maximum number of interface lines to 3. These values can be modified again in interactive mode.
- k** Send a signal to an active process (a.k.a. kill a process).
- q** Quit the program.
- PgDn**
Show the next page of the process/thread list.
With the arrow-down key the list can be scrolled downwards with single lines.
- ^F** Show the next page of the process/thread list (forward).
With the arrow-down key the list can be scrolled downwards with single lines.
- PgUp**
Show the previous page of the process/thread list.
With the arrow-up key the list can be scrolled upwards with single lines.
- ^B** Show the previous page of the process/thread list (backward).
With the arrow-up key the list can be scrolled upwards with single lines.
- ^L** Redraw the screen.

RAW DATA STORAGE

In order to store system and process level statistics for long-term analysis (e.g. to check the system load and the active processes running yesterday between 3:00 and 4:00 PM), *atop* can store the system and process level statistics in compressed binary format in a raw file with the flag **-w** followed by the filename. If this file already exists and is recognized as a raw data file, *atop* will append new samples to the file (starting with a sample which reflects the activity since boot); if the file does not exist, it will be created.

By default only processes which have been active during the interval are stored in the raw file. When the flag **-a** is specified, all processes will be stored.

The interval (default: 10 seconds) and number of samples (default: infinite) can be passed as last arguments. Instead of the number of samples, the flag **-S** can be used to indicate that *atop* should finish anyhow before midnight.

A raw file can be read and visualized again with the flag **-r** followed by the filename. If no filename is specified, the file `/var/log/atop/atop_YYYYMMDD` is opened for input (where *YYYYMMDD* are digits representing the current date). If a filename is specified in the format *YYYYMMDD* (representing any valid date), the file `/var/log/atop/atop_YYYYMMDD` is opened. If a filename with the symbolic name **y** is specified, yesterday’s daily logfile is opened (this can be repeated so ‘yyyy’ indicates the logfile of four days ago).

The samples from the file can be viewed interactively by using the key ‘t’ to show the next sample, the key ‘T’ to show the previous sample, the key ‘b’ to branch to a particular time or the key ‘r’ to rewind to the begin of the file.

When output is redirected to a file or pipe, **atop** prints all samples in plain ASCII. The default line length is

80 characters in that case; with the flag **-L** followed by an alternate line length, more (or less) columns will be shown.

With the flag **-b** (begin time) and/or **-e** (end time) followed by a time argument of the form HH:MM, a certain time period within the raw file can be selected.

When **atop** is installed, the script **atop.daily** is stored in the */etc/atop* directory. This script takes care that **atop** is activated every day at midnight to write compressed binary data to the file */var/log/atop/atop_YYYYMMDD* with an interval of 10 minutes.

Furthermore the script removes all raw files which are older than four weeks.

The script is activated via the **cron** daemon using the file */etc/cron.d/atop* with the contents

```
0 0 * * * root /etc/atop/atop.daily
```

When the package **psacct** is installed, the process accounting is automatically restarted via the **logrotate** mechanism. The file */etc/logrotate.d/psacct_atop* takes care that **atop** is finished just before the rotation of the process accounting file and the file */etc/logrotate.d/psacctu_atop* takes care that **atop** is restarted again after the rotation. When the package **psacct** is not installed, these logrotate-files have no effect.

OUTPUT DESCRIPTION

The first sample shows the system level activity since boot (the elapsed time in the header shows the time since boot). Note that particular counters could have reached their maximum value (several times) and started by zero again, so do not rely on these figures.

For every sample *atop* first shows the lines related to system level activity. If a particular system resource has not been used during the interval, the entire line related to this resource is suppressed. So the number of system level lines may vary for each sample.

After that a list is shown of processes which have been active during the last interval. This list is by default sorted on cpu consumption, but this order can be changed by the keys which are previously described.

If values have to be shown by *atop* which do not fit in the column width, another format is used. If e.g. a cpu-consumption of 233216 milliseconds should be shown in a column width of 4 positions, it is shown as '233s' (in seconds). For large memory figures, another unit is chosen if the value does not fit (Mb instead of Kb, Gb instead of Mb, Tb instead of Gb, ...). For other values, a kind of exponent notation is used (value 123456789 shown in a column of 5 positions gives 123e6).

OUTPUT DESCRIPTION - SYSTEM LEVEL

The system level information consists of the following output lines:

PRC

Process and thread level totals.

This line contains the total cpu time consumed in system mode ('sys') and in user mode ('user'), the total number of processes present at this moment ('#proc'), the total number of threads present at this moment in state 'running' ('#trun'), 'sleeping interruptible' ('#tslpi') and 'sleeping uninterruptible' ('#tslpu'), the number of zombie processes ('#zombie'), the number of clone system calls ('clones'), and the number of processes that ended during the interval ('#exit') when process accounting is used. Instead of '#exit' the last column may indicate that process accounting could not be activated ('no procacct').

If the screen-width does not allow all of these counters, only a relevant subset is shown.

CPU

CPU utilization.

At least one line is shown for the total occupation of all CPUs together.

In case of a multi-processor system, an additional line is shown for every individual processor (with 'cpu' in lower case), sorted on activity. Inactive CPUs will not be shown by default. The lines showing the per-cpu occupation contain the cpu number in the last field.

Every line contains the percentage of cpu time spent in kernel mode by all active processes ('sys'), the percentage of cpu time consumed in user mode ('user') for all active processes (including processes running with a nice value larger than zero), the percentage of cpu time spent for interrupt handling ('irq') including softirq, the percentage of unused cpu time while no processes were waiting for

disk-I/O ('idle'), and the percentage of unused cpu time while at least one process was waiting for disk-I/O ('wait').

In case of per-cpu occupation, the last column shows the cpu number and the wait percentage ('w') for that cpu. The number of lines showing the per-cpu occupation can be limited.

For virtual machines the steal-percentage is shown ('steal'), reflecting the percentage of cpu time stolen by other virtual machines running on the same hardware.

For physical machines hosting one or more virtual machines, the guest-percentage is shown ('guest'), reflecting the percentage of cpu time used by the virtual machines. Notice that this percentage overlaps the user-percentage.

In case of frequency-scaling, all previously mentioned CPU-percentages are relative to the used scaling of the CPU during the interval. If a CPU has been active for e.g. 50% in user mode during the interval while the frequency-scaling of that CPU was 40%, only 20% of the full capacity of the CPU has been used in user mode.

In case that the kernel module 'cpufreq_stats' is active (after issueing 'modprobe cpufreq_stats'), the *average* frequency ('avgf') and the *average* scaling percentage ('avgscal') is shown. Otherwise the *current* frequency ('curf') and the *current* scaling percentage ('curscal') is shown at the moment that the sample is taken.

If the screen-width does not allow all of these counters, only a relevant subset is shown.

CPL CPU load information.

This line contains the load average figures reflecting the number of threads that are available to run on a CPU (i.e. part of the runqueue) or that are waiting for disk I/O. These figures are averaged over 1 ('avg1'), 5 ('avg5') and 15 ('avg15') minutes.

Furthermore the number of context switches ('csw'), the number of serviced interrupts ('intr') and the number of available CPUs are shown.

If the screen-width does not allow all of these counters, only a relevant subset is shown.

MEM

Memory occupation.

This line contains the total amount of physical memory ('tot'), the amount of memory which is currently free ('free'), the amount of memory in use as page cache including the total resident shared memory ('cache'), the amount of memory within the page cache that has to be flushed to disk ('dirty'), the amount of memory used for filesystem meta data ('buff'), the amount of memory being used for kernel mallocs ('slab'), the amount of slab memory that is reclaimable ('slrec'), the resident size of shared memory including tmpfs ('shmem'), the resident size of shared memory ('shrss') the amount of shared memory that is currently swapped ('shswp'), the amount of memory that is currently claimed by vmware's balloon driver ('vmbal'), the amount of memory that is claimed for huge pages ('hptot'), and the amount of huge page memory that is really in use ('hpuse').

If the screen-width does not allow all of these counters, only a relevant subset is shown.

SWP

Swap occupation and overcommit info.

This line contains the total amount of swap space on disk ('tot') and the amount of free swap space ('free').

Furthermore the committed virtual memory space ('vmcom') and the maximum limit of the committed space ('vmlim', which is by default swap size plus 50% of memory size) is shown. The committed space is the reserved virtual space for all allocations of private memory space for processes. The kernel only verifies whether the committed space exceeds the limit if strict overcommit handling is configured (vm.overcommit_memory is 2).

PAG Paging frequency.

This line contains the number of scanned pages ('scan') due to the fact that free memory drops below

a particular threshold and the number times that the kernel tries to reclaim pages due to an urgent need ('stall').

Also the number of memory pages the system read from swap space ('swin') and the number of memory pages the system wrote to swap space ('swout') are shown.

LVM/MDD/DSK

Logical volume/multiple device/disk utilization.

Per active unit one line is produced, sorted on unit activity. Such line shows the name (e.g. Vol-Group00-lvtmp for a logical volume or sda for a hard disk), the busy percentage i.e. the portion of time that the unit was busy handling requests ('busy'), the number of read requests issued ('read'), the number of write requests issued ('write'), the number of KiBytes per read ('KiB/r'), the number of KiBytes per write ('KiB/w'), the number of MiBytes per second throughput for reads ('MBr/s'), the number of MiBytes per second throughput for writes ('MBw/s'), the average queue depth ('avq') and the average number of milliseconds needed by a request ('avio') for seek, latency and data transfer.

If the screen-width does not allow all of these counters, only a relevant subset is shown.

The number of lines showing the units can be limited per class (LVM, MDD or DSK) with the 'l' key or statically (see separate man-page of atoprc). By specifying the value 0 for a particular class, no lines will be shown any more for that class.

NFM

Network Filesystem (NFS) mount at the client side.

For each NFS-mounted filesystem, a line is shown that contains the mounted server directory, the name of the server ('srv'), the total number of bytes physically read from the server ('read') and the total number of bytes physically written to the server ('write'). Data transfer is subdivided in the number of bytes read via normal read() system calls ('nread'), the number of bytes written via normal read() system calls ('nwrit'), the number of bytes read via direct I/O ('dread'), the number of bytes written via direct I/O ('dwrit'), the number of bytes read via memory mapped I/O pages ('mread'), and the number of bytes written via memory mapped I/O pages ('mwrit').

NFC

Network Filesystem (NFS) client side counters.

This line contains the number of RPC calls issues by local processes ('rpc'), the number of read RPC calls ('read') and write RPC calls ('rpwrite') issued to the NFS server, the number of RPC calls being retransmitted ('retxmit') and the number of authorization refreshes ('autref').

NFS Network Filesystem (NFS) server side counters.

This line contains the number of RPC calls received from NFS clients ('rpc'), the number of read RPC calls received ('cread'), the number of write RPC calls received ('cwrit'), the number of network requests handled via TCP ('nettcp'), the number of network requests handled via UDP ('netudp'), the number of Megabytes/second returned to read requests by clients ('MBcr/s'), the number of Megabytes/second passed in write requests by clients ('MBcw/s'), the number of reply cache hits ('rchits'), the number of reply cache misses ('rcmiss') and the number of uncached requests ('rcnoca'). Furthermore some error counters indicating the number of requests with a bad format ('badfmt') or a bad authorization ('badaut'), and a counter indicating the number of bad clients ('badcln'). and the number of authorization refreshes ('autref').

NET

Network utilization (TCP/IP).

One line is shown for activity of the transport layer (TCP and UDP), one line for the IP layer and one line per active interface.

For the transport layer, counters are shown concerning the number of received TCP segments including those received in error ('tcpi'), the number of transmitted TCP segments excluding those containing only retransmitted octets ('tcpo'), the number of UDP datagrams received ('udpi'), the number of UDP datagrams transmitted ('udpo'), the number of active TCP opens ('tcpao'), the number of passive TCP opens ('tcppo'), the number of TCP output retransmissions ('tcprs'), the number of TCP

input errors ('tcpie'), the number of TCP output resets ('tcpor'), the number of UDP no ports ('udpnp'), and the number of UDP input errors ('udpie').

If the screen-width does not allow all of these counters, only a relevant subset is shown.

These counters are related to IPv4 and IPv6 combined.

For the IP layer, counters are shown concerning the number of IP datagrams received from interfaces, including those received in error ('ipi'), the number of IP datagrams that local higher-layer protocols offered for transmission ('ipo'), the number of received IP datagrams which were forwarded to other interfaces ('ipfrw'), the number of IP datagrams which were delivered to local higher-layer protocols ('deliv'), the number of received ICMP datagrams ('icmpi'), and the number of transmitted ICMP datagrams ('icmpo').

If the screen-width does not allow all of these counters, only a relevant subset is shown.

These counters are related to IPv4 and IPv6 combined.

For every active network interface one line is shown, sorted on the interface activity. Such line shows the name of the interface and its busy percentage in the first column. The busy percentage for half duplex is determined by comparing the interface speed with the number of bits transmitted and received per second; for full duplex the interface speed is compared with the highest of either the transmitted or the received bits. When the interface speed can not be determined (e.g. for the loop-back interface), '---' is shown instead of the percentage.

Furthermore the number of received packets ('pcki'), the number of transmitted packets ('pcko'), the line speed of the interface ('sp'), the effective amount of bits received per second ('si'), the effective amount of bits transmitted per second ('so'), the number of collisions ('coll'), the number of received multicast packets ('mkti'), the number of errors while receiving a packet ('erri'), the number of errors while transmitting a packet ('erro'), the number of received packets dropped ('drpi'), and the number of transmitted packets dropped ('drpo').

If the screen-width does not allow all of these counters, only a relevant subset is shown.

The number of lines showing the network interfaces can be limited.

OUTPUT DESCRIPTION - PROCESS LEVEL

Following the system level information, the processes are shown from which the resource utilization has changed during the last interval. These processes might have used cpu time or issued disk or network requests. However a process is also shown if part of it has been paged out due to lack of memory (while the process itself was in sleep state).

Per process the following fields may be shown (in alphabetical order), depending on the current output mode as described in the section INTERACTIVE COMMANDS and depending on the current width of your window:

AVGRSZ The average size of one read-action on disk.

AVGWSZ

The average size of one write-action on disk.

BANDWI

Total bandwidth for received TCP and UDP packets consumed by this process (bits-per-second). This value can be compared with the value 'si' on interface level (used bandwidth per interface). This information will only be shown when the kernel module 'netatop' is loaded.

BANDWO

Total bandwidth for sent TCP and UDP packets consumed by this process (bits-per-second). This value can be compared with the value 'so' on interface level (used bandwidth per interface). This information will only be shown when the kernel module 'netatop' is loaded.

CMD

The name of the process. This name can be surrounded by "less/greater than" signs ('<name>') which means that the process has finished during the last interval. Behind the abbreviation 'CMD' in the header line, the current page number and the total number of pages of the process/thread list are shown.

COMMAND-LINE

The full command line of the process (including arguments). If the length of the command line exceeds the length of the screen line, the arrow keys -> and <- can be used for horizontal scroll. Behind the verb 'COMMAND-LINE' in the header line, the current page number and the total number of pages of the process/thread list are shown.

CPU The occupation percentage of this process related to the available capacity for this resource on system level.

CPUNR The identification of the CPU the (main) thread is running on or has recently been running on.

CTID Container-id. If a process has been started and finished during the last interval, a '?' is shown because the container-id is not part of the standard process accounting record.

DSK The occupation percentage of this process related to the total load that is produced by all processes (i.e. total disk accesses by all processes during the last interval). This information is shown when per process "storage accounting" is active in the kernel.

EGID Effective group-id under which this process executes.

ENDATE

Date that the process has been finished. If the process is still running, this field shows 'active'.

ENTIME

Time that the process has been finished. If the process is still running, this field shows 'active'.

ENVID Virtual environment identified (OpenVZ only).

EUID Effective user-id under which this process executes.

EXC The exit code of a terminated process (second position of column 'ST' is E) or the fatal signal number (second position of column 'ST' is S or C).

FSGID Filesystem group-id under which this process executes.

FSUID Filesystem user-id under which this process executes.

MAJFLT

The number of page faults issued by this process that have been solved by creating/loading the requested memory page.

MEM The occupation percentage of this process related to the available capacity for this resource on system level.

MINFLT The number of page faults issued by this process that have been solved by reclaiming the requested memory page from the free list of pages.

NET The occupation percentage of this process related to the total load that is produced by all processes (i.e. consumed network bandwidth of all processes during the last interval). This information will only be shown when kernel module 'netatop' is loaded.

NICE The more or less static priority that can be given to a process on a scale from -20 (high priority) to +19 (low priority).

NPROCS

The number of active and terminated processes accumulated for this user or program.

PID Process-id. If a process has been started and finished during the last interval, a '?' is shown because the process-id is not part of the standard process accounting record.

POLI The policies 'norm' (normal, which is SCHED_OTHER), 'btch' (batch) and 'idle' refer to time-sharing processes. The policies 'fifo' (SCHED_FIFO) and 'rr' (round robin, which is SCHED_RR) refer to realtime processes.

PPID Parent process-id. If a process has been started and finished during the last interval, value 0 is shown because the parent process-id is not part of the standard process accounting record.

- PRI** The process' priority ranges from 0 (highest priority) to 139 (lowest priority). Priority 0 to 99 are used for realtime processes (fixed priority independent of their behavior) and priority 100 to 139 for timesharing processes (variable priority depending on their recent CPU consumption and the nice value).
- PSIZE** The proportional memory size of this process (or user).
Every process shares resident memory with other processes. E.g. when a particular program is started several times, the code pages (text) are only loaded once in memory and shared by all incarnations. Also the code of shared libraries is shared by all processes using that shared library, as well as shared memory and memory-mapped files. For the PSIZE calculation of a process, the resident memory of a process that is shared with other processes is divided by the number of sharers. This means, that every process is accounted for a proportional part of that memory. Accumulating the PSIZE values of all processes in the system gives a reliable impression of the total resident memory consumed by all processes.
Since gathering of all values that are needed to calculate the PSIZE is a relatively time-consuming task, the 'R' key (or '-R' flag) should be active. Gathering these values also requires superuser privileges (otherwise '?K' is shown in the output).
If a process has finished during the last interval, no value is shown since the proportional memory size is not part of the standard process accounting record.
- RDDSK** When the kernel maintains standard io statistics ($\geq 2.6.20$):
The read data transfer issued physically on disk (so reading from the disk cache is not accounted for).
Unfortunately, the kernel aggregates the data transfer of a process to the data transfer of its parent process when terminating, so you might see transfers for (parent) processes like cron, bash or init, that are not really issued by them.
- RGID** The real group-id under which the process executes.
- RGROW** The amount of resident memory that the process has grown during the last interval. A resident growth can be caused by touching memory pages which were not physically created/loaded before (load-on-demand). Note that a resident growth can also be negative e.g. when part of the process is paged out due to lack of memory or when the process frees dynamically allocated memory. For a process which started during the last interval, the resident growth reflects the total resident size of the process at that moment.
If a process has finished during the last interval, no value is shown since resident memory occupation is not part of the standard process accounting record.
- RNET** The number of TCP- and UDP packets received by this process. This information will only be shown when kernel module 'netatop' is installed.
If a process has finished during the last interval, no value is shown since network counters are not part of the standard process accounting record.
- RSIZE** The total resident memory usage consumed by this process (or user). Notice that the RSIZE of a process includes all resident memory used by that process, even if certain memory parts are shared with other processes (see also the explanation of PSIZE).
If a process has finished during the last interval, no value is shown since resident memory occupation is not part of the standard process accounting record.
- RTPR** Realtime priority according the POSIX standard. Value can be 0 for a timesharing process (policy 'norm', 'btch' or 'idle') or ranges from 1 (lowest) till 99 (highest) for a realtime process (policy 'rr' or 'fifo').
- RUID** The real user-id under which the process executes.
- S** The current state of the (main) thread: 'R' for running (currently processing or in the runqueue), 'S' for sleeping interruptible (wait for an event to occur), 'D' for sleeping non-interruptible, 'Z' for zombie (waiting to be synchronized with its parent process), 'T' for stopped (suspended or traced), 'W' for swapping, and 'E' (exit) for processes which have finished during the last interval.

- SGID** The saved group-id of the process.
- SNET** The number of TCP and UDP packets transmitted by this process. This information will only be shown when the kernel module 'netatop' is loaded.
- ST** The status of a process.
The first position indicates if the process has been started during the last interval (the value *N* means 'new process').

The second position indicates if the process has been finished during the last interval.
The value *E* means 'exit' on the process' own initiative; the exit code is displayed in the column 'EXC'.
The value *S* means that the process has been terminated unvoluntarily by a signal; the signal number is displayed in the in the column 'EXC'.
The value *C* means that the process has been terminated unvoluntarily by a signal, producing a core dump in its current directory; the signal number is displayed in the column 'EXC'.
- STDATE** The start date of the process.
- STTIME** The start time of the process.
- SUID** The saved user-id of the process.
- SWAPSZ** The swap space consumed by this process (or user).
- SYSCPU** CPU time consumption of this process in system mode (kernel mode), usually due to system call handling.
- TCPRASZ**
The average size of a received TCP buffer in bytes. This information will only be shown when the kernel module 'netatop' is loaded.
- TCPRCV**
The number of TCP packets received for this process. This information will only be shown when the kernel module 'netatop' is loaded.
- TCPSASZ**
The average size of a transmitted TCP buffer in bytes. This information will only be shown when the kernel module 'netatop' is loaded.
- TCPSND** The number of TCP packets transmitted for this process. This information will only be shown when the kernel module 'netatop' is loaded.
- THR** Total number of threads within this process. All related threads are contained in a thread group, represented by *atop* as one line or as a separate line when the 'y' key (or -y flag) is active.

On Linux 2.4 systems it is hardly possible to determine which threads (i.e. processes) are related to the same thread group. Every thread is represented by *atop* as a separate line.
- TID** Thread-id. All threads within a process run with the same PID but with a different TID. This value is shown for individual threads in multi-threaded processes (when using the key 'y').
- TRUN** Number of threads within this process that are in the state 'running' (R).
- TSLPI** Number of threads within this process that are in the state 'interruptible sleeping' (S).
- TSLPU** Number of threads within this process that are in the state 'uninterruptible sleeping' (D).
- UDPRASZ**
The average size of a received UDP packet in bytes. This information will only be shown when the kernel module 'netatop' is loaded.
- UDPRCV**
The number of UDP packets received by this process. This information will only be shown when the kernel module 'netatop' is loaded.

UDPSASZ

The average size of a transmitted UDP packets in bytes. This information will only be shown when the kernel module 'netatop' is loaded.

UDPSND

The number of UDP packets transmitted by this process. This information will only be shown when the kernel module 'netatop' is loaded.

USRCPU

CPU time consumption of this process in user mode, due to processing the own program text.

VDATA

The virtual memory size of the private data used by this process (including heap and shared library data).

VGROW

The amount of virtual memory that the process has grown during the last interval. A virtual growth can be caused by e.g. issuing a malloc() or attaching a shared memory segment. Note that a virtual growth can also be negative by e.g. issuing a free() or detaching a shared memory segment. For a process which started during the last interval, the virtual growth reflects the total virtual size of the process at that moment.

If a process has finished during the last interval, no value is shown since virtual memory occupation is not part of the standard process accounting record.

VPID

Virtual process-id (within a container). If a process has been started and finished during the last interval, a '?' is shown because the virtual process-id is not part of the standard process accounting record.

VSIZE

The total virtual memory usage consumed by this process (or user).

If a process has finished during the last interval, no value is shown since virtual memory occupation is not part of the standard process accounting record.

VSLIBS

The virtual memory size of the (shared) text of all shared libraries used by this process.

VSTACK

The virtual memory size of the (private) stack used by this process

VSTEXT

The virtual memory size of the (shared) text of the executable program.

WRDSK

When the kernel maintains standard io statistics ($\geq 2.6.20$):

The write data transfer issued physically on disk (so writing to the disk cache is not accounted for). This counter is maintained for the application process that writes its data to the cache (assuming that this data is physically transferred to disk later on). Notice that disk I/O needed for swapping is not taken into account.

Unfortunately, the kernel aggregates the data transfer of a process to the data transfer of its parent process when terminating, so you might see transfers for (parent) processes like cron, bash or init, that are not really issued by them.

WCANCL

When the kernel maintains standard io statistics ($\geq 2.6.20$):

The write data transfer previously accounted for this process or another process that has been cancelled. Suppose that a process writes new data to a file and that data is removed again before the cache buffers have been flushed to disk. Then the original process shows the written data as WRDSK, while the process that removes/truncates the file shows the unflushed removed data as WCANCL.

PARSEABLE OUTPUT

With the flag **-P** followed by a list of one or more labels (comma-separated), parseable output is produced for each sample. The labels that can be specified for system-level statistics correspond to the labels (first verb of each line) that can be found in the interactive output: "CPU", "cpu" "CPL" "MEM", "SWP", "PAG", "LVM", "MDD", "DSK", "NFM", "NFC", "NFS" and "NET".

For process-level statistics special labels are introduced: "PRG" (general), "PRC" (cpu), "PRM" (memory), "PRD" (disk, only if "storage accounting" is active) and "PRN" (network, only if the kernel module

'netatop' has been installed).

With the label "ALL", all system and process level statistics are shown.

For every interval all requested lines are shown whereafter **atop** shows a line just containing the label "SEP" as a separator before the lines for the next sample are generated.

When a sample contains the values since boot, **atop** shows a line just containing the label "RESET" before the lines for this sample are generated.

The first part of each output-line consists of the following six fields: **label** (the name of the label), **host** (the name of this machine), **epoch** (the time of this interval as number of seconds since 1-1-1970), **date** (date of this interval in format YYYY/MM/DD), **time** (time of this interval in format HH:MM:SS), and **interval** (number of seconds elapsed for this interval).

The subsequent fields of each output-line depend on the label:

CPU Subsequent fields: total number of clock-ticks per second for this machine, number of processors, consumption for all CPUs in system mode (clock-ticks), consumption for all CPUs in user mode (clock-ticks), consumption for all CPUs in user mode for niced processes (clock-ticks), consumption for all CPUs in idle mode (clock-ticks), consumption for all CPUs in wait mode (clock-ticks), consumption for all CPUs in irq mode (clock-ticks), consumption for all CPUs in softirq mode (clock-ticks), consumption for all CPUs in steal mode (clock-ticks), consumption for all CPUs in guest mode (clock-ticks) overlapping user mode, frequency of all CPUs and frequency percentage of all CPUs.

cpu Subsequent fields: total number of clock-ticks per second for this machine, processor-number, consumption for this CPU in system mode (clock-ticks), consumption for this CPU in user mode (clock-ticks), consumption for this CPU in user mode for niced processes (clock-ticks), consumption for this CPU in idle mode (clock-ticks), consumption for this CPU in wait mode (clock-ticks), consumption for this CPU in irq mode (clock-ticks), consumption for this CPU in softirq mode (clock-ticks), consumption for this CPU in steal mode (clock-ticks), consumption for this CPU in guest mode (clock-ticks) overlapping user mode, frequency of this CPU and frequency percentage of this CPU.

CPL Subsequent fields: number of processors, load average for last minute, load average for last five minutes, load average for last fifteen minutes, number of context-switches, and number of device interrupts.

MEM Subsequent fields: page size for this machine (in bytes), size of physical memory (pages), size of free memory (pages), size of page cache (pages), size of buffer cache (pages), size of slab (pages), dirty pages in cache (pages), reclaimable part of slab (pages), size of vmware's balloon pages (pages), total size of shared memory (pages), size of resident shared memory (pages), size of swapped shared memory (pages), huge page size (in bytes), total size of huge pages (huge pages), and size of free huge pages (huge pages).

SWP Subsequent fields: page size for this machine (in bytes), size of swap (pages), size of free swap (pages), 0 (future use), size of committed space (pages), and limit for committed space (pages).

PAG Subsequent fields: page size for this machine (in bytes), number of page scans, number of alloc-stalls, 0 (future use), number of swapins, and number of swapouts.

LVM/MDD/DSK

For every logical volume/multiple device/hard disk one line is shown.

Subsequent fields: name, number of milliseconds spent for I/O, number of reads issued, number of sectors transferred for reads, number of writes issued, and number of sectors transferred for write.

NFM Subsequent fields: mounted NFS filesystem, total number of bytes read, total number of bytes written, number of bytes read by normal system calls, number of bytes written by normal system calls, number of bytes read by direct I/O, number of bytes written by direct I/O, number of pages read by memory-mapped I/O, and number of pages written by memory-mapped I/O.

- NFC** Subsequent fields: number of transmitted RPCs, number of transmitted read RPCs, number of transmitted write RPCs, number of RPC retransmissions, and number of authorization refreshes.
- NFS** Subsequent fields: number of handled RPCs, number of received read RPCs, number of received write RPCs, number of bytes read by clients, number of bytes written by clients, number of RPCs with bad format, number of RPCs with bad authorization, number of RPCs from bad client, total number of handled network requests, number of handled network requests via TCP, number of handled network requests via UDP, number of handled TCP connections, number of hits on reply cache, number of misses on reply cache, and number of uncached requests.
- NET** First one line is produced for the upper layers of the TCP/IP stack.
Subsequent fields: the verb "upper", number of packets received by TCP, number of packets transmitted by TCP, number of packets received by UDP, number of packets transmitted by UDP, number of packets received by IP, number of packets transmitted by IP, number of packets delivered to higher layers by IP, and number of packets forwarded by IP.
- Next one line is shown for every interface.
Subsequent fields: name of the interface, number of packets received by the interface, number of bytes received by the interface, number of packets transmitted by the interface, number of bytes transmitted by the interface, interface speed, and duplex mode (0=half, 1=full).
- PRG** For every process one line is shown.
Subsequent fields: PID (unique ID of task), name (between brackets), state, real uid, real gid, TGID (group number of related tasks/threads), total number of threads, exit code, start time (epoch), full command line (between brackets), PPID, number of threads in state 'running' (R), number of threads in state 'interruptible sleeping' (S), number of threads in state 'uninterruptible sleeping' (D), effective uid, effective gid, saved uid, saved gid, filesystem uid, filesystem gid, elapsed time (hertz), is_process (y/n), virtual pid and container id.
- PRC** For every process one line is shown.
Subsequent fields: PID, name (between brackets), state, total number of clock-ticks per second for this machine, CPU-consumption in user mode (clockticks), CPU-consumption in system mode (clockticks), nice value, priority, realtime priority, scheduling policy, current CPU, sleep average, TGID (group number of related tasks/threads) and is_process (y/n).
- PRM** For every process one line is shown.
Subsequent fields: PID, name (between brackets), state, page size for this machine (in bytes), virtual memory size (Kbytes), resident memory size (Kbytes), shared text memory size (Kbytes), virtual memory growth (Kbytes), resident memory growth (Kbytes), number of minor page faults, number of major page faults, virtual library exec size (Kbytes), virtual data size (Kbytes), virtual stack size (Kbytes), swap space used (Kbytes), TGID (group number of related tasks/threads), is_process (y/n) and proportional set size (Kbytes) if in 'R' option is specified.
- PRD** For every process one line is shown.
Subsequent fields: PID, name (between brackets), state, obsoleted kernel patch installed ('n'), standard io statistics used ('y' or 'n'), number of reads on disk, cumulative number of sectors read, number of writes on disk, cumulative number of sectors written, cancelled number of written sectors, TGID (group number of related tasks/threads) and is_process (y/n).
If the standard I/O statistics ($\geq 2.6.20$) are not used, the disk I/O counters per process are not relevant. The counters 'number of reads on disk' and 'number of writes on disk' are obsoleted anyhow.
- PRN** For every process one line is shown.
Subsequent fields: PID, name (between brackets), state, kernel module 'netatop' loaded ('y' or 'n'), number of TCP-packets transmitted, cumulative size of TCP-packets transmitted, number of TCP-packets received, cumulative size of TCP-packets received, number of UDP-packets transmitted, cumulative size of UDP-packets transmitted, number of UDP-packets received, cumulative size of UDP-packets transmitted, number of raw packets transmitted (obsolete, always 0), number of raw packets received (obsolete, always 0), TGID (group number of related

tasks/threads) and `is_process` (y/n).

If the kernel module is not active, the network I/O counters per process are not relevant.

EXAMPLES

To monitor the current system load interactively with an interval of 5 seconds:

```
atop 5
```

To monitor the system load and write it to a file (in plain ASCII) with an interval of one minute during half an hour with active processes sorted on memory consumption:

```
atop -M 60 30 > /log/atop.mem
```

Store information about the system and process activity in binary compressed form to a file with an interval of ten minutes during an hour:

```
atop -w /tmp/atop.raw 600 6
```

View the contents of this file interactively:

```
atop -r /tmp/atop.raw
```

View the processor and disk utilization of this file in parseable format:

```
atop -PCPU,DSK -r /tmp/atop.raw
```

View the contents of today's standard logfile interactively:

```
atop -r
```

View the contents of the standard logfile of the day before yesterday interactively:

```
atop -r yy
```

View the contents of the standard logfile of 2014, June 7 from 02:00 PM onwards interactively:

```
atop -r 20140607 -b 14:00
```

FILES

/var/run/pacct_shadow.d/

Directory containing the process accounting shadow files that are used by *atop* when the *atopacctd* daemon is active.

/tmp/atop.d/atop.acct

File in which the kernel writes the accounting records when *atop* itself has activated the process accounting mechanism.

/etc/atoprc

Configuration file containing system-wide default values. See related man-page.

~/.atoprc

Configuration file containing personal default values. See related man-page.

/var/log/atop/atop_YYYYMMDD

Raw file, where *YYYYMMDD* are digits representing the current date. This name is used by the script **atop.daily** as default name for the output file, and by **atop** as default name for the input file when using the **-r** flag.

All binary system and process level data in this file has been stored in compressed format.

/var/run/netatop.log

File that contains the netpertask structs containing the network counters of exited processes. These structs are written by the *netatopd* daemon and read by *atop* after reading the standard process accounting records.

SEE ALSO

atopsar(1), **atoprc(5)**, **atopacctd(8)**, **netatop(4)**, **netatopd(8)**, **logrotate(8)**

<http://www.atoptool.nl>

AUTHOR

Gerlof Langeveld (gerlof.langeveld@atoptool.nl)
JC van Winkel