NAME

atop — AT Computing's System & Process Monitor

SYNOPSIS

Interactive usage:

atop [-g|-m|-d|-n|-u|-p|-s|-c|-v] [-C|-M|-D|-N|-A] [-af1x] [*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] [-C|-M|-D|-N|-A] [-f1x]

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- and network load is only shown per process if a kernel patch 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 curses library) and behaves interactively; in the second case it produces flat ASCII-output.

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 the latter case *atop* will switch 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). The flags used are the same as the keys which 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

When *atop* is started, it switches on the process accounting mechanism in the kernel. This forces the kernel to write a record with accounting information to the accounting file whenever a process ends. Apart from the kernel administration related to the running processes, *atop* also interprets the accounting records on disk with every interval; in this way *atop* can also show the activity of a process during the interval in which it is finished.
Whenever the last incarnation of *atop* stops (either by pressing 'q' or by 'kill -15'), it switches off the process accounting mechanism again. 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.

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 noticable 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 noticable 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) 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’.
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 section CONFIGURATION FILE).

When a resource exceeded its critical occupation percentage, the entire screen line is colored red.
When a resource exceeded (default) 80% of its critical percentage (so it is almost critical), the entire screen line is colored cyan. This ‘almost critical percentage’ (one value for all resources) can be modified in the configuration file (see section CONFIGURATION FILE).

With the key ’x’ (or flag -x), line coloring can be suppressed.

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 list.

g Show generic output (default).

Per process the following fields are shown: 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 contain the username, number of threads in the thread group, the status and exit code. However if the kernel patch ‘cnt’ has been installed, the number of read- and write transfers, and the number of received and sent network packets are shown.
The last columns contain the state, the occupation percentage for the choosen resource (default: cpu) and the process name.

m Show memory related output.

Per process the following fields are shown: 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.

d Show disk-related output.

Per process the following fields are shown: process-id, number of physical disk reads, average size per read (bytes), total size for read transfers, physical disk writes, average size per write (bytes), total size for write transfers, disk occupation percentage and process name.
This information can only be shown when kernel patch ‘cnt’ is installed.

n Show network related output.

Per process the following fields are shown: process-id, 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), and received and send raw packets (e.g. ICMP) in one column, the network occupation percentage and process name.
This information can only be shown when kernel patch ‘cnt’ is installed.

s Show scheduling characteristics.

Per process/thread the following fields are shown: process-id, thread group id, number of threads in thread group, scheduling policy (normal timesharing, realtime round-robin, realtime fifo), nice value, priority, realtime priority, current processor, status, state, the occupation percentage for the choosen resource and the process name.

v Show various process characteristics.

Per process the following fields are shown: 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 choosen resource and the process name.

c Show the command line of the process.

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

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 the kernel patch ‘cnt’ has been installed, the accumulated number of read- and write transfers on disk, and the number of received and sent network packets are shown. When the kernel patch is not installed, these counters are zero.
The last columns contain the accumulated occupation percentage for the choosen 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 the kernel patch ‘cnt’ has been installed, the accumulated number of read- and write transfers on disk, and the number of received and sent network packets are shown. When the kernel patch is not installed, these counters are zero.
The last columns contain the accumulated occupation percentage for the choosen 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 packets received/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 or network is only possible when kernel patch ‘cnt’ is installed.

Miscellaneous interactive commands:

? Request for help information (also the key ’h’ can be pressed).

V Request for version information (version number and date).

x Use colors to highlight critical resources (toggle).

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.

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.

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.

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.

f Fixate the number of lines for system resources (toggle). By default only the lines are shown about system resources (cpu, paging, disk, network) that really have been active during the last interval.

With this key you can force *atop* to show lines of inactive resources as well.

1 Show relevant counters as an average per second (in the format '..../s') instead of as a total during the interval (toggle).

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 cpu's, disks and network interfaces which have been active during the last interval. Limiting these lines can be useful on systems with huge number cpu's, 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 (aka kill a process).

q Quit the monitor program.

ˆF Show the next page of the process list (forward).

ˆB Show the previous page of the process list (backward).

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 just 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/opt/UNItools/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/opt/UNItools/log/atop"/atop_***YYYYMMDD* is opened.
The samples from the file can be viewed interactively by using the key 't' to show the next sample and the key 'T' to show the previous sample. When output is redirected to a file or pipe, **atop** prints all samples in plain ASCII.
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, two scripts are stored in the */etc/atop* directory. Each of these scripts take care that **atop** is activated every day to write compressed binary data to the file **/var/opt/UNItools/log/atop"/atop_***YYYYMMDD* with an interval of 10 minutes.
Furthermore the script removes all raw files which are older than four weeks.

Only one of these scripts should be used for automatic storage of the system- and process level information:

atop.daily This script should be used for systems on which process accounting is *not* activated via **logrotate** (i.e. the file **/etc/logrotate.d/psacct** is not present). In that case the script **atop.daily** can be activated every day (at midnight) via the **cron** daemon by creating the file */etc/cron.d/atop* with the contents

```
1 0 * * * root /etc/atop/atop.daily
```

atop.24hours For systems on which process accounting is daily restarted via **logrotate** the script **atop.24hours** should be used:

The section ’postrotate’ in the file **/etc/logrotate.d/psacct** should be extended by calling the script **/etc/atop/atop.24hours** (without arguments) *after* reactivating process accounting with the **accton** command.

OUTPUT DESCRIPTION

The first sample shows the system level activity since boot (the elapsed time in the header shows the number of seconds 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 notation 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). For other values, a kind of exponent notation is used (value 123456789 shown in a column of 5 positions gives 123e6).

The system level information consists of the following output lines:

PRC

Process 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 number of zombie processes (‘#zombie’) and the number of processes that ended during the interval (‘#exit’, which shows ‘?’ if process accounting could not be switched on).

CPU

CPU utilization.
One line is shown for the total occupation of all CPU’s 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 cpu’s 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.

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’) and the number of serviced interrupts (‘intr’) are 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 (‘cache’), the amount of memory

used for filesystem meta data ('buff') and the amount of memory being used for kernel malloc's ('slab' - always 0 for kernel 2.4).

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.

DSK

Disk utilization.
Per active disk one line is produced, sorted on disk activity. Such line shows the name of the disk (e.g. hda or sda), the busy percentage i.e. the portion of time that the disk was busy handling requests ('busy'), the number of read requests issued ('read'), the number of write requests issued ('write') and the average number of milliseconds needed by a request ('avio') for seek, latency and data transfer.

The number of lines showing the disk occupation can be limited.

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') and the number of UDP datagrams transmitted ('udpo'). These counters are related to IPv4 and IPv6.
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') and the number of IP datagrams which were delivered to local higher-layer protocols ('deliv'). These counters are related to IPv4 and IPv6.
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 loopback interface), '---' is shown instead of the percentage.
Furthermore the number of received packets ('pcki'), the number of transmitted packets ('pcko'), the effective amount of bits received per second ('si') and the effective amount of bits transmitted per second ('so').

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

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:

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 list are shown.

COMMAND-LINE
The full command line of the process (including arguments), which is limited to the length of the screen line. Th command line can be surrounded by "less/greater than" signs (‘<line>’) which means that the process has finished during the last interval.
Behind the verb ‘COMMAND-LINE’ in the header line, the current page number and the total number of pages of the process list are shown.

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

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 can only be shown when kernel patch ‘cnt’ is installed.

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).

GROUP The real primary group identity under which the process runs.

MAJFLT
The number of page faults issued by this process.

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

MINFLT The number of page reclaims issued by this process.

NET The occupation percentage of this process related to the total load that is produced by all processes (i.e. network packets transferred by all processes during the last interval).
This information can only be shown when kernel patch ‘cnt’ is installed.

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. However when the kernel patch ‘acct’ is installed, this value will be shown.

POLICY Policy ’normal’ (SCHED_OTHER) refers to a timesharing process, ’fifo’ (SCHED_FIFO) and ’roundr’ (SCHED_RR) to a realtime process.

PRIO 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).

RAWRS The number of raw datagrams received and sent by this process. This information can only be shown when kernel patch ‘cnt’ is installed.
If a process has finished during the last interval, no value is shown since network counters are not registered in the standard process accounting record. However when the kernel patch ‘acct’ is installed, this value will be shown.

RDDSK The number of read accesses issued physically on disk (so reading from the disk cache is not accounted for). This information can only be shown when kernel patch ‘cnt’ is installed.

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. However when the kernel patch ‘acct’ is installed, this value will be shown.

RNET The number of TCP- and UDP packets received by this process. This information can only be shown when kernel patch ‘cnt’ 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. However when the kernel patch ‘acct’ is installed, this value will be shown.

RSIZE The total resident memory usage consumed by this process (or user).
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. However when the kernel patch ‘acct’ is installed, this value will be shown.

S The current state of the process: ‘R’ for running (currently processing or in the run queue), ‘S’ for sleeping interruptable (wait for an event to occur), ‘D’ for sleeping non-interruptable, ‘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.

SNET The number of TCP- and UDP packets transmitted by this process. This information can only be shown when kernel patch ‘cnt’ 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. However when the kernel patch ‘acct’ is installed, this value will be shown.

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 in the column ‘EXC’.

STDATE The start date of the process.

STTIME The start time of the process.

SYSCPU CPU time consumption of this process in system mode (kernel mode), usually due to system call handling.

TCPRCV
The number of receive requests issued by this process for TCP sockets, and the average size per transfer in bytes. This information can only be shown when kernel patch ‘cnt’ is installed.
If a process has finished during the last interval, no value is shown since network counters are not registered in the standard process accounting record. However when the kernel patch ‘acct’ is installed, this value will be shown.

TCPSND The number of send requests issued by this process for TCP sockets, and the average size per transfer in bytes. This information can only be shown when kernel patch ‘cnt’ is installed.
If a process has finished during the last interval, no value is shown since network counters are not registered in the standard process accounting record. However when the kernel patch ‘acct’ is installed, this value will be shown.

THR A multithreaded application consists of various threads. All related threads are contained in a thread group, represented by *atop* as one line.

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.

UDPRCV
The number of UDP datagrams received by this process, and the average size per transfer in bytes. This information can only be shown when kernel patch ‘cnt’ is installed.
If a process has finished during the last interval, no value is shown since network counters are not registered in the standard process accounting record. However when the kernel patch ‘acct’ is installed, this value will be shown.

UDPSND
The number of UDP datagrams transmitted by this process, and the average size per transfer in bytes. This information can only be shown when kernel patch ‘cnt’ is installed.
If a process has finished during the last interval, no value is shown since network counters are not registered in the standard process accounting record. However when the kernel patch ‘acct’ is installed, this value will be shown.

USERNAME
The real user identity under which the process runs.

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

VGROW The amount of virtual memory that the process has grown during the last interval. A virtual growth can be caused by e.g. issueing a malloc() or attaching a shared memory segment. Note that a virtual growth can also be negative by e.g. issueing 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. However when the kernel patch ‘acct’ is installed, this value will be shown.

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. However when the kernel patch ‘acct’

is installed, this value will be shown.

VSTEXT
The virtual memory size used by the shared text of this process.

WRDSK The number of write accesses issued physically on disk (so writing to the disk cache is not accounted for). Usually application processes just transfer their data to the cache, while the physical write accesses are done later on by kernel daemons. This information can only be shown when kernel patch ‘cnt’ is installed.
Note that the number read- and write accesses are not separately maintained in the standard process accounting record. This means that only one value is given for read’s and write’s in case a process has finished during the last interval. However when the kernel patch ‘acct’ is installed, these values will be shown separately.

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:

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

CONFIGURATION FILE

The default values used by **atop** can be overruled by a personal configuration file. This file, called **˜/.atoprc** contains a keyword-value pair one every line (blank lines and lines starting with a #-sign are skipped). The following keywords can be specified:

flags A list of default flags can be defined here. The flags which are allowed are ’g’, ’m’, ’d’, ’n’, ’u’, ’p’, ’s’, ’c’, ’v’, ’C’, ’M’, ’D’, ’N’, ’A’, ’a’, ’f’, ’1’ and ’x’.

interval The default interval value in seconds.

username
The default regular expression for the users for which active processes will be shown.

procname
The default regular expression for the process names to be shown.

maxlinecpu
The maximum number of active CPU’s which will be shown.

maxlinedisk
The maximum number of active disks which will be shown.

maxlineintf
The maximum number of active network interfaces which will be shown.

cpucritperc

The busy percentage considered critical for a processor (see section COLORS). This percentage is used to determine a weighted percentage for line coloring and sorting of active processes. When this value is zero, no line coloring or automatic sorting is performed for this resource.

dskcritperc

The busy percentage considered critical for a disk (see section COLORS). This percentage is used to determine a weighted percentage for line coloring and sorting of active processes. When this value is zero, no line coloring or automatic sorting is performed for this resource.

netcritperc

The busy percentage considered critical for a network interface (see section COLORS). This percentage is used to determine a weighted percentage for line coloring and sorting of active processes. When this value is zero, no line coloring or automatic sorting is performed for this resource.

memcritperc

The percentage considered critical for memory utilization (see section COLORS). This percentage is used to determine a weighted percentage for line coloring and sorting of active processes. When this value is zero, no line coloring or automatic sorting is performed for this resource.

swpcritperc

The occupation percentage considered critical for swap space (see section COLORS). This percentage is used to determine a weighted percentage for line coloring and sorting of active processes. When this value is zero, no line coloring or automatic sorting is performed for this resource.

almostcrit

A percentage of the critical percentage to determine if the resource is almost critical (see section COLORS). When this value is zero, no line coloring for ‘almost critical’ is performed.

An example of the **˜/.atoprc** file:

```
flags        af
interval     5
username
procname
maxlinecpu    4
maxlinedisk   10
maxlineintf   5
cpucritperc   80
almostcrit    90
```

FILES

/tmp/atop.d/atop.acct

File in which the kernel writes the accounting records if the standard accounting to the file **/var/log/pacct** or **/var/account/pacct** is not used.

˜/.atoprc

Configuration file containing personal default values.

/var/opt/UNItools/log/atop"/atop_*YYYYMMDD*

Raw file, where *YYYYMMDD* are digits representing the current date. This name is used by the scripts **atop.daily** and **atop.24hours** 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.

SEE ALSO

atsar, logrotate
http://www.ATComputing.nl/atop, http://www.ATConsultancy.nl/atop

AUTHOR

Gerlof Langeveld, AT Computing (gerlof@ATComputing.nl)