The sys admin’s daily grind: QPS

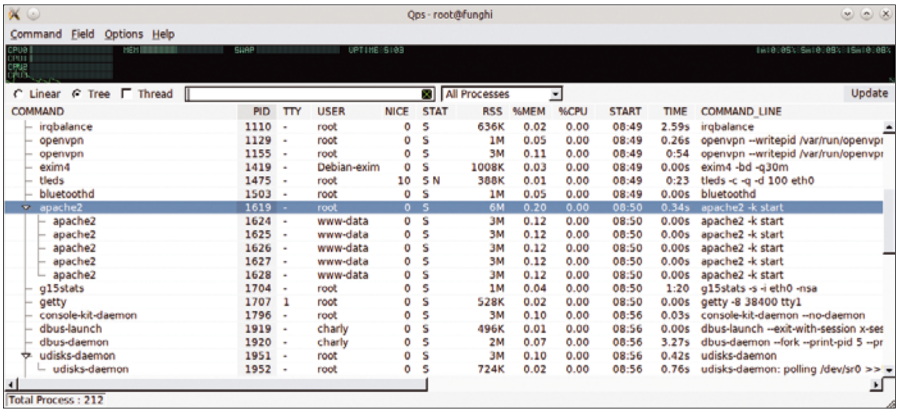
Process Monitor

The graphical tool QPS frees admins from Kafkaesque ambiguities about the cause, history, and side effects of running processes. Depending on the view, either clarity or detailed information dominate the scene. *By Charly Kühnast*

Graphical tools are something I use sparingly – with the practical and fast QPS [1] being one exception. The process monitor, programmed in Qt combines the characteristics of `top`, `PS`, and `lsof` in a single package. The main window (Figure 1) shows central system parameters, such as CPU utilization, memory, swap, and uptime. This is followed by the tabular process list, which QPS presents in linear form, or as a tree.

In the *Tree* view, I can hide or show child processes. A click on a column header changes the sort order. I like to sort processes by the UID to keep the most sensitive ones, those with root privileges, firmly in view. The flexibility of the process list is a major advantage of QPS. At the start, only the first seven or eight most important fields appear. You can use *Fields* | *Custom Fields* to choose your focus fields from a total of 48 – if want to see all the fields simultaneously, you should invest in a second display.

QPS makes it easy to find processes with the same properties: I just press Shift + Ctrl and click a field in the process table at the same time. If I click on the value *0* in the `UID` column, QPS selects all processes that belong to root. Right-clicking on a process as root gives me the option of sending signals of all kinds (`SIGHUP`, `SIGKILL`, etc.) to this process. In the same context menu, I can also change the `nice` value (i.e., shift the execution priority up or down) and even change the scheduling.



COMMAND	PID	TTY	USER	NICE	STAT	RSS	%MEM	%CPU	START	TIME	COMMAND_LINE
irqbalance	1110	-	root	0	S	636K	0.02	0.00	08:49	2.59s	irqbalance
openvpn	1129	-	root	0	S	1M	0.05	0.00	08:49	0.26s	openvpn --writepid /var/run/openvpı
openvpn	1155	-	root	0	S	3M	0.11	0.00	08:49	0:54	openvpn --writepid /var/run/openvpı
exim4	1419	-	Debian-exim	0	S	1008K	0.03	0.00	08:49	0.00s	exim4 -bd -q30m
tleds	1475	-	root	10	S N	388K	0.01	0.00	08:49	0:23	tleds -c -q -d 100 eth0
bluetoothd	1503	-	root	0	S	1M	0.05	0.00	08:49	0.00s	bluetoothd
apache2	1619	-	root	0	S	6M	0.20	0.00	08:50	0.34s	apache2 -k start
apache2	1624	-	www-data	0	S	3M	0.12	0.00	08:50	0.00s	apache2 -k start
apache2	1625	-	www-data	0	S	3M	0.12	0.00	08:50	0.00s	apache2 -k start
apache2	1626	-	www-data	0	S	3M	0.12	0.00	08:50	0.00s	apache2 -k start
apache2	1627	-	www-data	0	S	3M	0.12	0.00	08:50	0.00s	apache2 -k start
apache2	1628	-	www-data	0	S	3M	0.12	0.00	08:50	0.00s	apache2 -k start
g15stats	1704	-	root	0	S	1M	0.04	0.00	08:50	1:20	g15stats -s -i eth0 -nsa
getty	1707	1	root	0	S	528K	0.02	0.00	08:50	0.00s	getty -8 38400 tty1
console-kit-daemon	1796	-	root	0	S	3M	0.10	0.00	08:56	0.03s	console-kit-daemon --no-daemon
dbus-launch	1919	-	charly	0	S	496K	0.01	0.00	08:56	0.00s	dbus-launch --exit-with-session x-ses
dbus-daemon	1920	-	charly	0	S	2M	0.07	0.00	08:56	3.27s	dbus-daemon --fork --print-pid 5 --pr
udisks-daemon	1951	-	root	0	S	3M	0.10	0.00	08:56	0.42s	udisks-daemon
udisks-daemon	1952	-	root	0	S	724K	0.02	0.00	08:56	0.76s	udisks-daemon: polling /dev/sr0 >>

Figure 1: The main window in QPS presents a detailed process list. Just like `top`, the tool informs the user of CPU, RAM, swap, and uptime at the top of the window.

Finding the Talkers

The *View Details* item in the context menu opens a window containing additional information about the selected process. The window has several tabs; the first shows a list of all the network sockets the process is keeping open. Here, I can check quickly whether the process is just listening on the desired IP addresses and ports or whether it is being inappropriately verbose.

In the next tab, you will find a list of all the files the process is currently accessing. The question “which process is currently writing to file *XY*,” is one I need to answer quite often in my daily work. Luckily, an additional tab contains the memory-mapping information – a feature that will delight some developers. However, I am more interested in the *Environment* section, which lists all of the relevant environment variables for the respective process. The last section, *All fields*, displays a list of the 48 fields (which you will remember from the main view) with the currently assigned values.

QPS only updates its presentation every five seconds, which is why it seems to be slightly slower than its text-based colleagues. However, I can speed things up in the Options menu. Of course, if you choose a too demanding update interval, QPS clambers to the top of the process list itself. ∎∎∎

INFO

[1] QPS: *http://qps.kldp.net/projects/qps*

AUTHOR

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.