Linux ShortCuts

Written by Zack MIlls
Thursday, 10 September 2009 22:29 -

pwd

Print working directory, i.e., display the name of my current directory on the screen.
hostname

Print the name of the local host (the machine on which I am working). Use netconf (as root) to change the name of the machine.

whoami

Print my login name.

id username

Print user id (uid) and his/her group id (gid), effective id (if different than the real id) and the supplementary groups.

date
Print the operating system current date, time and timezone. For an ISO standard format, I have to use: date -Iseconds
I can change the date and time to 2000-12-31 23:57 using this command: date 123123572000
or using these two commands (easier to remember):
date --set 2000-12-31
date --set 23:57:00

To set the hardware (BIOS) clock from the system (Linux) clock, I can use the command (as root): setclock

The international (ISO 8601) standard format for all-numeric date/time has the form: 2001-01-31 (as in Linux default "C" localization). You can be more precise if you wish using, for example: 2001-01-31 23:59:59.999-05:00 (representing I millisecond before February 2001, in a timezone which is 5 hours behind the Universal Coordinated Time (UTC)) . The most "kosher" representation of the same point in time could be: 2001013172359S9.999-0500. See the standard at ftp://ftp.qsl.net/pub/g1smd/8601v03.pdf.

time

Determine the amount of time that it takes for a process to complete + other process accounting. Don't confuse it with the date command (see previous entry). E.g. I can find out how long it takes to display a directory content using: time ls. Or I can test the time function with time sleep 10 (time the commands the does nothing for 10 seconds).
clock
hwclock

(two commands, use either). Obtain date/time from the computer hardware (real time, battery-powered) clock. You can also use one of this commands to set the hardware clock, but setclock may be simplier (see 2 commands above). Example: hwclock --systohc --utc sets the hardware clock (in UTC) from the system clock.

who

Determine the users logged on the machine.

w

Determine who is logged on the system, find out what they are doing, their processor ussage, etc. Handy security command.

rwho -a

(=remote who) Determine users logged on other computers on your network. The rwho service must be enabled for this command to run. If it isn't, run setup (RedHat specific) as root to enable "rwho".

finger user_name

System info about a user. Try: finger root . One can use finger with any networked computer that exposes the finger service to the world, e.g., I can do (try): finger @finger.kernel.org

last

Show listing of users last logged-in on your system. Really good idea to check it from time to time as a security measure on your system.

lastb

(="last bad") Show the last bad (unsuccessful) login attempts on my system. It did not work on my system, so got it started with: touch /var/log/btmp

"There's a good reason why /var/log/btmp isn't available on any sane set-up - it's a world-readable file containing login mistakes. Since one of the most common login mistakes is to type the password instead of the username, /var/log/btmp is a gift to crackers." (Thanks to Bruce Richardson). It appears the problem can be solved by changing the file permissions so only root can use "lastb":

chmod o+r /var/log/btmp

history | more

Show the last (1000 or so) commands executed from the command line on the current account. The "| more" causes the display to stop after each screenful. To see what another user was doing on your system, login as "root" and inspect his/her "history". The history is kept in the file .bash_history in the user home directory (so yes, it can be modified or erased).
uptime

Show the amount of time since the last reboot.

ps

(="print status" or "process status") List the processes currently run by the current user.

ps axu | more

List all the processes currently running, even those without the controlling terminal, together with the name of the user that owns each process.

top

Keep listing the currently running processes on my computer, sorted by cpu usage (top processes first). Press c when done.

PID = process identification.

USER=name of the user who owns (started!) the process.
PRI=priority of the process (the higher the number, the lower the priority, normal 0, highest priority is -20, lowest 20.
NI=niceness level (i.e., if the process tries to be nice by adjusting the priority by the number given). The higher the number, the higher the niceness of the process (i.e., its priority is lower).
SIZE=Kilobytes of program+data+stack taken by the process in memory.
RSS=Kilobytes of physical RAM taken by the process.
SHARE=Kilobytes of memory shared with other processes.
STAT=state of the process: S-sleeping, R-running, T-stopped or traced, D-uninterruptable sleep, Z=zombie.
%CPU=share of the CPU usage (since last screen update).
%MEM=share of physical memory.
TIME=total CPU time used by the process (since it was started).

COMMAND=command line used to start the task (careful with passwords, etc., on command line, all permitted to run "top" may see them!

gtop
ktop

(in X terminal) Two GUI choices for top. My favourite is gtop (comes with gnome). In KDE, ktop is also available from the "K"menu under "System"-"Task Manager".
uname -a

(= "Unix name" with option "all") Info on your (local) server. I can also use guname (in X-window terminal) to display the info more nicely.

XFree86 -version

Show me the version of X windows I have on my system.

cat /etc/issue

Check what distribution you are using. You can put your own message in this text file--it's displayed on login. It is more common to put your site-specific login message to the file /etc/motd ("motd"= "message of the day").

free

Memory info (in kilobytes). "Shared" memory is the memory that can be shared between processes (e.g., executable code is "shared"). "Buffered" and "cached" memory is the part that keeps parts of recently accessed files--it can be shrunk if more memory is needed by processes.

df -h

(=disk free) Print disk info about all the filesystems (in human-readable form).

du / -bh | more

(=disk usage) Print detailed disk usage for each subdirectory starting at the "/" (root) directory (in human legible form).

cat /proc/cpuinfo

Cpu info--it shows the content of the file cpuinfo. Note that the files in the /proc/directory are not real files--they are hooks to look at information available to the kernel.

cat /proc/interrupts

List the interrupts in use. May need to find out before setting up new hardware.

cat /proc/version

Linux version and other info.

cat /proc/filesystems

Show the types of filesystems currently in use.

cat /etc/printcap |more

Show the setup of printers.

lsmod

(= "lst modules". As root. Use /sbin/lsmod to execute this command when you are a non-root user.) Show the kernel modules currently loaded.

set|more

Show the current user environment (in full). Normally too much to bother.

echo $PATH

Show the content of the environment variable "PATH". This command can be used to show other environment variables as well. Use set to see the full environment (see the previous command).

dmesg | less

Print kernel messages (the content of the so-called kernel ring buffer). Press "q" to quit "less". Use less /var/log/dmesg to see what "dmesg" dumped into this file right after the last system bootup.

chage -l my_login_name

See my password expiry information.

quota

See my disk quota (the limits of disk usage).

sysctl -a |more

Display all the configurable Linux kernel parameters.

runlevel

Print the previous and current runlevel. The output "N5" means: "no previous runlevel" and "5 is the current runlevel". To change the runlevel, use "init", e.g., /init 1switches the system to a single user mode.

Runlevel is the mode of operation of Linux. Runlevel can be switched "on the fly" using the command /init. For example, /init 3 (as root) will switch me to runlevel 3. The following runlevels are standard:

- *0 - /halt (Do NOT set initdefault to this)*
- *1 - Single user mode*
- *2 - Multiuser, without NFS (The same as 3, if you do not have networking)*
- *3 - Full multiuser mode*
- *4 - unused*
- *5 - X11*
- *6 - reboot (Do NOT set initdefault to this)*

The system default runlevel is set in the file /etc/initab .

sar

View information extracted the system activity log file (/var/log/sarxx where xx is the current day number). sar can extract many kinds of system statistics including CPU load averages, i/o statistics, and network traffic statistics for the current day and (usually) several days backs.