

Command	Description
• <code>apropos whatis</code>	Show commands pertinent to string. See also <a href="#">threadsafe</a>
• <code>man -t ascii   ps2pdf - &gt; ascii.pdf</code>	make a pdf of a manual page
<code>which command</code>	Show full path name of command
<code>time command</code>	See how long a command takes
• <code>time cat</code>	Start stopwatch. Ctrl-d to stop. See also <a href="#">sw</a>
dir navigation	
• <code>cd -</code>	Go to previous directory
• <code>cd</code>	Go to \$HOME directory
<code>(cd dir &amp;&amp; command)</code>	Go to dir, execute command and return to current dir
• <code>pushd .</code>	Put current dir on stack so you can popd back to it
file searching	
• <code>alias l='ls -l --color=auto'</code>	quick dir listing
• <code>ls -lrt</code>	List files by date. See also <a href="#">newest</a> and <a href="#">find_mm_yyyy</a>
• <code>ls /usr/bin   pr -T9 -W\$COLUMNS</code>	Print in 9 columns to width of terminal
<code>find -name '*.ch'   xargs grep -E 'expr'</code>	Search 'expr' in this dir and below. See also <a href="#">findrepo</a>
<code>find -type f -print0   xargs -r0 grep -F 'example'</code>	Search all regular files for 'example' in this dir and below
<code>find -maxdepth 1 -type f   xargs grep -F 'example'</code>	Search all regular files for 'example' in this dir
<code>find -maxdepth 1 -type d   while read dir; do echo \$dir; echo cmd2; done</code>	Process each item with multiple commands (in while loop)
• <code>find -type f ! -perm -444</code>	Find files not readable by all (useful for web site)
• <code>find -type d ! -perm -111</code>	Find dirs not accessible by all (useful for web site)
• <code>locate -r 'file[^/]*\.txt'</code>	Search cached index for names. This re is like glob *file*.txt
• look reference	Quickly search (sorted) dictionary for prefix
• <code>grep --color reference /usr/share/dict/words</code>	Highlight occurrences of regular expression in dictionary
archives and compression	
<code>gpg -c file</code>	Encrypt file
<code>gpg file.gpg</code>	Decrypt file
<code>tar -c dir/   bzip2 &gt; dir.tar.bz2</code>	Make compressed archive of dir/
<code>bzip2 -dc dir.tar.bz2   tar -x</code>	Extract archive (use gzip instead of bzip2 for tar.gz files)

<code>tar -c dir/   gzip   gpg -c   ssh user@remote 'dd of=dir.tar.gz.gpg'</code>	Make encrypted archive of dir/ on remote machine
<code>find dir/ -name '*.txt'   tar -c --files-from=-   bzip2 &gt; dir_txt.tar.bz2</code>	Make archive of subset of dir/ and below
<code>find dir/ -name '*.txt'   xargs cp -a --target-directory=dir_txt/ --parents</code>	Make copy of subset of dir/ and below
<code>( tar -c /dir/to/copy )   ( cd /where/to/ &amp;&amp; tar -x -p )</code>	Copy (with permissions) copy/ dir to /where/to/ dir
<code>( cd /dir/to/copy &amp;&amp; tar -c . )   ( cd /where/to/ &amp;&amp; tar -x -p )</code>	Copy (with permissions) contents of copy/ dir to /where/to/
<code>( tar -c /dir/to/copy )   ssh -C user@remote 'cd /where/to/ &amp;&amp; tar -x -p'</code>	Copy (with permissions) copy/ dir to remote:/where/to/ dir
<code>dd bs=1M if=/dev/sda   gzip   ssh user@remote 'dd of=sda.gz'</code>	Backup harddisk to remote machine
<b>rsync (Network efficient file copier: Use the --dry-run option for testing)</b>	
<code>rsync -P rsync://rsync.server.com/path/to/file file</code>	Only get diffs. Do multiple times for troublesome downloads
<code>rsync --bwlimit=1000 fromfile tofile</code>	Locally copy with rate limit. It's like nice for I/O
<code>rsync -az -e ssh --delete ~/public_html/ remote.com:'~/public_html'</code>	Mirror web site (using compression and encryption)
<code>rsync -auz -e ssh remote:/dir/ . &amp;&amp; rsync -auz -e ssh . remote:/dir/</code>	Synchronize current directory with remote one
<b>ssh (Secure SHell)</b>	
<code>ssh \$USER@\$HOST command</code>	Run command on \$HOST as \$USER (default command=shell)
• <code>ssh -f -Y \$USER@\$HOSTNAME xeyes</code>	Run GUI command on \$HOSTNAME as \$USER
<code>scp -p -r \$USER@\$HOST: file dir/</code>	Copy with permissions to \$USER's home directory on \$HOST
<code>scp -c arcfour \$USER@\$LANHOST: bigfile</code>	Use faster crypto for local LAN. This might saturate GigE
<code>ssh -g -L 8080:localhost:80 root@\$HOST</code>	Forward connections to \$HOSTNAME:8080 out to \$HOST:80
<code>ssh -R 1434:imap:143 root@\$HOST</code>	Forward connections from \$HOST:1434 in to imap:143
<code>ssh-copy-id \$USER@\$HOST</code>	Install public key for \$USER@\$HOST for password-less log in
<b>wget (multi purpose download tool)</b>	
• <code>(cd dir/ &amp;&amp; wget -nd -pHEKk http://www.pixelbeat.org/cmdline.html)</code>	Store local browsable version of a page to the current dir
<code>wget -c http://www.example.com/large.file</code>	Continue downloading a partially downloaded file
<code>wget -r -nd -np -l1 -A '*.jpg' http://www.example.com/dir/</code>	Download a set of files to the current directory
<code>wget ftp://remote/file[1-9].iso/</code>	FTP supports globbing directly

• wget -q -O- http://www.pixelbeat.org/timeline.html   grep 'a href'   head	Process output directly
echo 'wget url'   at 01:00	Download url at 1AM to current dir
wget --limit-rate=20k url	Do a low priority download (limit to 20KB/s in this case)
wget -nv --spider --force-html -i bookmarks.html	Check links in a file
wget --mirror http://www.example.com/	Efficiently update a local copy of a site (handy from cron)
networking (Note ifconfig, route, mii-tool, nslookup commands are <a href="#">obsolete</a> )	
ethtool eth0	Show status of ethernet interface eth0
ethtool --change eth0 autoneg off speed 100 duplex full	Manually set ethernet interface speed
iw dev wlan0 link	Show link status of wireless interface wlan0
iw dev wlan0 set bitrates legacy-2.4 1	Manually set wireless interface speed
• iw dev wlan0 scan	List wireless networks in range
• ip link show	List network interfaces
ip link set dev eth0 name wan	Rename interface eth0 to wan
ip link set dev eth0 up	Bring interface eth0 up (or down)
• ip addr show	List addresses for interfaces
ip addr add 1.2.3.4/24 brd + dev eth0	Add (or del) ip and mask (255.255.255.0)
• ip route show	List routing table
ip route add default via 1.2.3.254	Set default gateway to 1.2.3.254
• ss -tupl	List internet services on a system
• ss -tup	List active connections to/from system
• host pixelbeat.org	Lookup DNS ip address for name or vice versa
• hostname -i	Lookup local ip address (equivalent to host `hostname`)
• whois pixelbeat.org	Lookup whois info for hostname or ip address
windows networking (Note samba is the package that provides all this windows specific networking support)	
• smbtree	Find windows machines. See also findsmb
nmblookup -A 1.2.3.4	Find the windows (netbios) name associated with ip address
smbclient -L windows_box	List shares on windows machine or samba server
mount -t smbfs -o fmask=666,guest //windows_box/share /mnt/share	Mount a windows share
echo 'message'   smbclient -M windows_box	Send popup to windows machine (off by default in XP sp2)
text manipulation (Note sed uses stdin and stdout. Newer versions support inplace editing with the -i option)	

<code>sed 's/string1/string2/g'</code>	Replace string1 with string2
<code>sed 's/(.*)1/12/g'</code>	Modify anystring1 to anystring2
<code>sed '/^ *#/d; /^ *\$/d'</code>	Remove comments and blank lines
<code>sed ':a; /\\$/N; s/\\n//; ta'</code>	Concatenate lines with trailing \
<code>sed 's/[ \t]*\$//'</code>	Remove trailing spaces from lines
<code>sed 's/([`"\$\]\\)/\\1/g'</code>	Escape shell metacharacters active within double quotes
• <code>seq 10   sed "s/^/ /; s/ *(\{7,\})/1/"</code>	Right align numbers
• <code>seq 10   sed p   paste - -</code>	Duplicate a column
<code>sed -n '1000{p;q}'</code>	Print 1000th line
<code>sed -n '10,20p;20q'</code>	Print lines 10 to 20
<code>sed -n 's/.*&lt;title&gt;\(.*\)&lt;/title&gt;.*\1/ip;T;q'</code>	Extract title from HTML web page
<code>sed -i 42d ~/.ssh/known_hosts</code>	Delete a particular line
<code>sort -t. -k1,1n -k2,2n -k3,3n -k4,4n</code>	Sort IPV4 ip addresses
• <code>echo 'Test'   tr '[:lower:]' '[:upper:]'</code>	Case conversion
• <code>tr -dc '[:print:]' &lt; /dev/urandom</code>	Filter non printable characters
• <code>tr -s '[:blank:]' '\t' &lt;/proc/diskstats   cut -f4</code>	cut fields separated by blanks
• <code>history   wc -l</code>	Count lines
• <code>seq 10   paste -s -d ' '</code>	Concatenate and separate line items to a single line
set operations (Note you can <a href="#">export LANG=C</a> for speed. Also these assume no duplicate lines within a file)	
<code>sort file1 file2   uniq</code>	<u>Union</u> of unsorted files
<code>sort file1 file2   uniq -d</code>	<u>Intersection</u> of unsorted files
<code>sort file1 file1 file2   uniq -u</code>	<u>Difference</u> of unsorted files
<code>sort file1 file2   uniq -u</code>	<u>Symmetric Difference</u> of unsorted files
<code>join -t'\0' -a1 -a2 file1 file2</code>	Union of sorted files
<code>join -t'\0' file1 file2</code>	Intersection of sorted files
<code>join -t'\0' -v2 file1 file2</code>	Difference of sorted files
<code>join -t'\0' -v1 -v2 file1 file2</code>	Symmetric Difference of sorted files
math	
• <code>echo '(1 + sqrt(5))/2'   bc -l</code>	Quick math (Calculate $\Phi$ ). See also <a href="#">bc</a>
• <code>seq -f '4/%g' 1 2 99999   paste -sd+   bc -l</code>	Calculate $\pi$ the unix way
• <code>echo 'pad=20; min=64; (100*10^6)/((pad+min)*8)'   bc</code>	More complex (int) e.g. This shows max FastE packet rate
• <code>echo 'pad=20; min=64; print (100E6)/((pad+min)*8)'   python</code>	Python handles scientific notation
• <code>echo 'pad=20; plot [64:1518] (100*10**6)/((pad+x)*8)'   gnuplot -persist</code>	Plot FastE packet rate vs packet size
• <code>echo 'obase=16; ibase=10; 64206'   bc</code>	Base conversion (decimal to hexadecimal)
• <code>echo \$((0x2dec))</code>	Base conversion (hex to dec) ((shell arithmetic expansion))

• <code>units -t '100m/9.58s' 'miles/hour'</code>	Unit conversion (metric to imperial)
• <code>units -t '500GB' 'GiB'</code>	Unit conversion (SI to IEC prefixes)
• <code>units -t '1 googol'</code>	Definition lookup
• <code>seq 100   paste -s -d+   bc</code>	Add a column of numbers. See also <a href="#">add</a> and <a href="#">funcpy</a>
calendar	
• <code>cal -3</code>	Display a calendar
• <code>cal 9 1752</code>	Display a calendar for a particular month year
• <code>date -d fri</code>	What date is it this friday. See also <a href="#">day</a>
• <code>[ \$(date -d '12:00 +1 day' +%d) = '01' ]    exit</code>	exit a script unless it's the last day of the month
• <code>date --date='25 Dec' +%A</code>	What day does xmas fall on, this year
• <code>date --date='@2147483647'</code>	Convert seconds since the epoch (1970-01-01 UTC) to date
• <code>TZ='America/Los_Angeles' date</code>	What time is it on west coast of US (use <code>tzselect</code> to find TZ)
• <code>date --date='TZ="America/Los_Angeles" 09:00 next Fri'</code>	What's the local time for 9AM next Friday on west coast US
locales	
• <code>printf "%'d\n" 1234</code>	Print number with thousands grouping appropriate to locale
• <code>BLOCK_SIZE='\1 ls -l</code>	Use locale thousands grouping in <code>ls</code> . See also <a href="#">l</a>
• <code>echo "I live in `locale territory`"</code>	Extract info from locale database
• <code>LANG=en_IE.utf8 locale int_prefix</code>	Lookup locale info for specific country. See also <a href="#">ccodes</a>
• <code>locale -kc \$(locale   sed -n 's/\(LC_.*\)=.*\$/1/p')   less</code>	List fields available in locale database
recode (Obsoletes <code>iconv</code> , <code>dos2unix</code> , <code>unix2dos</code> )	
• <code>recode -l   less</code>	Show available conversions (aliases on each line)
<code>recode windows-1252.. file_to_change.txt</code>	Windows "ansi" to local charset (auto does CRLF conversion)
<code>recode utf-8/CRLF.. file_to_change.txt</code>	Windows utf8 to local charset
<code>recode iso-8859-15..utf8 file_to_change.txt</code>	Latin9 (western europe) to utf8
<code>recode ../b64 &lt; file.txt &gt; file.b64</code>	Base64 encode
<code>recode /qp.. &lt; file.qp &gt; file.txt</code>	Quoted printable decode
<code>recode ../HTML &lt; file.txt &gt; file.html</code>	Text to HTML
• <code>recode -lf windows-1252   grep euro</code>	Lookup <a href="#">table of characters</a>
• <code>echo -n 0x80   recode latin-9/x1..dump</code>	Show what a code represents in latin-9 charmap
• <code>echo -n 0x20AC   recode ucs-2/x2..latin-9/x</code>	Show latin-9 encoding
• <code>echo -n 0x20AC   recode ucs-2/x2..utf-8/x</code>	Show utf-8 encoding
CDs	

<code>gzip &lt; /dev/cdrom &gt; cdrom.iso.gz</code>	Save copy of data cdrom
<code>mkisofs -V LABEL -r dir   gzip &gt; cdrom.iso.gz</code>	Create cdrom image from contents of dir
<code>mount -o loop cdrom.iso /mnt/dir</code>	Mount the cdrom image at /mnt/dir (read only)
<code>cdrecord -v dev=/dev/cdrom blank=fast</code>	Clear a CDRW
<code>gzip -dc cdrom.iso.gz   cdrecord -v dev=/dev/cdrom -</code>	Burn cdrom image (use dev=ATAPI -scanbus to confirm dev)
<code>cdparanoia -B</code>	Rip audio tracks from CD to wav files in current dir
<code>cdrecord -v dev=/dev/cdrom -audio -pad *.wav</code>	Make audio CD from all wavs in current dir (see also <a href="#">cdrdao</a> )
<code>oggenc --tracknum=\$track track.cdda.wav -o track.ogg</code>	Make ogg file from wav file
disk space (See also <a href="#">FSlint</a> )	
• <code>ls -lSr</code>	Show files by size, biggest last
• <code>du -s *   sort -k1,1rn   head</code>	Show top disk users in current dir. See also <a href="#">dutop</a>
• <code>du -hs /home/*   sort -k1,1h</code>	Sort paths by easy to interpret disk usage
• <code>df -h</code>	Show free space on mounted filesystems
• <code>df -i</code>	Show free inodes on mounted filesystems
• <code>fdisk -l</code>	Show disks partitions sizes and types (run as root)
• <code>rpm -q -a --qf '%10{SIZE}\t%{NAME}\n'   sort -k1,1n</code>	List all <a href="#">packages</a> by installed size (Bytes) on rpm distros
• <code>dpkg-query -W -f='\${Installed-Size;10}\t%{Package}\n'   sort -k1,1n</code>	List all <a href="#">packages</a> by installed size (KBytes) on deb distros
• <code>dd bs=1 seek=2TB if=/dev/null of=ext3.test</code>	Create a large test file (taking no space). See also <a href="#">truncate</a>
• <code>&gt; file</code>	truncate data of file or create an empty file
monitoring/debugging	
• <code>tail -f /var/log/messages</code>	<a href="#">Monitor messages</a> in a log file
• <code>strace -c ls &gt;/dev/null</code>	Summarise/profile system calls made by command
• <code>strace -f -e open ls &gt;/dev/null</code>	List system calls made by command
• <code>strace -f -e trace=write -e write=1,2 ls &gt;/dev/null</code>	Monitor what's written to stdout and stderr
• <code>ltrace -f -e getenv ls &gt;/dev/null</code>	List library calls made by command
• <code>lsof -p \$\$</code>	List paths that process id has open
• <code>lsof ~</code>	List processes that have specified path open
• <code>tcpdump not port 22</code>	Show network traffic except ssh. See also

	<a href="#">tcpdump_not_me</a>
• <code>ps -e -o pid,args --forest</code>	List processes in a hierarchy
• <code>ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu   sed '/^ 0.0 /d'</code>	List processes by % cpu usage
• <code>ps -e -orss=,args=   sort -b -kl,ln   pr -TW\$COLUMNS</code>	List processes by mem (KB) usage. See also <a href="#">ps_mem.py</a>
• <code>ps -C firefox-bin -L -o pid,tid,pcpu,state</code>	List all threads for a particular process
• <code>ps -p 1,\$\$ -o etime=</code>	List elapsed wall time for particular process IDs
• <code>last reboot</code>	Show system reboot history
• <code>free -m</code>	Show amount of (remaining) RAM (-m displays in MB)
• <code>watch -n.1 'cat /proc/interrupts'</code>	Watch changeable data continuously
• <code>udevadm monitor</code>	Monitor udev events to help configure rules
system information (see also <a href="#">sysinfo</a> ) ('#' means root access is required)	
• <code>uname -a</code>	Show kernel version and system architecture
• <code>head -nl /etc/issue</code>	Show name and version of distribution
• <code>cat /proc/partitions</code>	Show all partitions registered on the system
• <code>grep MemTotal /proc/meminfo</code>	Show RAM total seen by the system
• <code>grep "model name" /proc/cpuinfo</code>	Show CPU(s) info
• <code>lspci -tv</code>	Show PCI info
• <code>lsusb -tv</code>	Show USB info
• <code>mount   column -t</code>	List mounted filesystems on the system (and align output)
• <code>grep -F capacity: /proc/acpi/battery/BAT0/info</code>	Show state of cells in laptop battery
# <code>dmidecode -q   less</code>	Display SMBIOS/DMI information
# <code>smartctl -A /dev/sda   grep Power_On_Hours</code>	How long has this disk (system) been powered on in total
# <code>hdparm -i /dev/sda</code>	Show info about disk sda
# <code>hdparm -tT /dev/sda</code>	Do a read speed test on disk sda
# <code>badblocks -s /dev/sda</code>	Test for unreadable blocks on disk sda
interactive (see also <a href="#">linux keyboard shortcuts</a> )	
• <a href="#">readline</a>	Line editor used by bash, python, bc, gnuplot, ...
• <a href="#">screen</a>	Virtual terminals with detach capability, ...
• <a href="#">mc</a>	Powerful file manager that can browse rpm, tar, ftp, ssh, ...
• <a href="#">gnuplot</a>	Interactive/scriptable graphing
• <a href="#">links</a>	Web browser
	open a file or url

- `xdg-open .`

with the registered  
desktop application

© Jan 7 2008