

Command	Description
• <code>apropos whatis</code>	Show commands pertinent to string. See also threadsafe
• <code>man -t ascii ps2pdf -> 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 sw
dir navigation	
• <code>cd -</code>	Go to previous directory
• <code>cd</code>	Go to \$HOME directory
<code>(cd dir && 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. See also l
• <code>ls -ltr</code>	List files by date. See also newest and find_mm_yyyy
• <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 findrepo
<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 <code>*file*.txt</code>
• <code>look reference</code>	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 > 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 > 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/ && tar -x -p)</code>	Copy (with permissions) copy/ dir to /where/to/ dir
<code>(cd /dir/to/copy && tar -c .) (cd /where/to/ && tar -x -p)</code>	Copy (with permissions) contents of copy/ dir to /where/to/ dir
<code>(tar -c /dir/to/copy) ssh -C user@remote 'cd /where/to/ && 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
rsync (Network efficient file copier: Use the --dry-run option for testing)	
<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/ . && rsync -auz -e ssh . remote:/dir/</code>	Synchronize current directory with remote one
ssh (Secure SHell)	
<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
wget (multi purpose download tool)	
• <code>(cd dir/ && 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
• <code>wget -q -O- http://www.pixelbeat.org/timeline.html grep 'a href' head</code>	Process output directly
<code>echo 'wget url' at 01:00</code>	Download url at 1AM to current dir
<code>wget --limit-rate=20k url</code>	Do a low priority download (limit to 20KB/s in this case)
<code>wget -nv --spider --force-html -i bookmarks.html</code>	Check links in a file
<code>wget --mirror http://www.example.com/</code>	Efficiently update a local copy of a site (handy from cron)
networking (Note ifconfig, route, mii-tool, nslookup commands are obsolete)	
<code>ethtool eth0</code>	Show status of ethernet interface eth0
<code>ethtool --change eth0 autoneg off speed 100 duplex full</code>	Manually set ethernet interface speed
<code>iw dev wlan0 link</code>	Show link status of wireless interface wlan0
<code>iw dev wlan0 set bitrates legacy-2.4 1</code>	Manually set wireless interface speed
• <code>iw dev wlan0 scan</code>	List wireless networks in range
• <code>ip link show</code>	List network interfaces
<code>ip link set dev eth0 name wan</code>	Rename interface eth0 to wan
<code>ip link set dev eth0 up</code>	Bring interface eth0 up (or down)
• <code>ip addr show</code>	List addresses for interfaces
<code>ip addr add 1.2.3.4/24 brd + dev eth0</code>	Add (or del) ip and mask (255.255.255.0)
• <code>ip route show</code>	List routing table
<code>ip route add default via 1.2.3.254</code>	Set default gateway to 1.2.3.254
• <code>ss -tupl</code>	List internet services on a system
• <code>ss -tup</code>	List active connections to/from system
• <code>host pixelbeat.org</code>	Lookup DNS ip address for name or vice versa
• <code>hostname -i</code>	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 find smb
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)	
sed 's/string1/string2/g'	Replace string1 with string2
sed 's/(.*)/1/12/g'	Modify anystring1 to anystring2
sed '/^ *#/d; /^ *\$/d'	Remove comments and blank lines
sed ':a; \\\\$/N; s/\\n//; ta'	Concatenate lines with trailing \
sed 's/[\t]*\$/'	Remove trailing spaces from lines
sed 's/\([^\\$]\)/\\1/g'	Escape shell metacharacters active within double quotes
• seq 10 sed "s/^ / /; s/^\(.\{7,\}\)/\1"	Right align numbers
• seq 10 sed p paste - -	Duplicate a column
sed -n '1000{p;q}'	Print 1000th line
sed -n '10,20p;20q'	Print lines 10 to 20
sed -n 's/.*<title>(.*)</title>.*\1/ip;T;q'	Extract title from HTML web page
sed -i 42d ~/.ssh/known_hosts	Delete a particular line
sort -t. -k1,1n -k2,2n -k3,3n -k4,4n	Sort IPV4 ip addresses
• echo 'Test' tr '[:lower:]' '[:upper:]'	Case conversion
• tr -dc '[:print:]' < /dev/urandom	Filter non printable characters
• tr -s '[:blank:]' '\t' </proc/diskstats cut -f4	cut fields separated by blanks
• history wc -l	Count lines
• seq 10 paste -s -d ' '	Concatenate and separate line items to a single line
set operations (Note you can export LANG=C for speed. Also these assume no duplicate lines within a file)	
sort -u file1 file2	Union of unsorted files
sort file1 file2 uniq -d	Intersection of unsorted files
sort file1 file1 file2 uniq -u	Difference of unsorted files
sort file1 file2 uniq -u	Symmetric Difference of unsorted files
join -t'\0' -a1 -a2 file1 file2	Union of sorted files
join -t'\0' file1 file2	Intersection of sorted files
join -t'\0' -v2 file1 file2	Difference of sorted files
join -t'\0' -v1 -v2 file1 file2	Symmetric Difference of sorted files
math	
• echo '(1 + sqrt(5))/2' bc -l	Quick math (Calculate φ). See also bc
• seq -f '4/%g' 1 2 99999 paste -sd-+ bc -l	Calculate π the unix way
• echo 'pad=20; min=64; (100*10^6)/((pad+min)*8)' bc	More complex (int) e.g. This shows max FastE packet rate
• echo 'pad=20; min=64; print (100E6)/((pad+min)*8)' python	Python handles scientific notation
• echo 'pad=20; plot [64:1518] (100*10**6)/((pad+x)*8)' gnuplot -persist	Plot FastE packet rate vs packet size
• echo 'obase=16; ibase=10; 64206' bc	Base conversion (decimal to hexadecimal)
• echo \$(0x2dec)	Base conversion (hex to dec) ((shell arithmetic expansion))
• units -t '100m/9.58s' 'miles/hour'	Unit conversion (metric to imperial)
• units -t '500GB' 'GiB'	Unit conversion (SI to IEC prefixes). See also numfmt
• units -t '1 googol'	Definition lookup
• seq 100 paste -s -d+ bc	Add a column of numbers. See also add and funcpy
calendar	
• cal -3	Display a calendar
• cal 9 1752	Display a calendar for a particular month year
• date -d fri	What date is it this friday. See also day
• [\$(date -d '12:00 today +1 day' +%d) = '01'] exit	exit a script unless it's the last day of the month
• date --date='25 Dec' +%A	What day does xmas fall on, this year
• date --date='@2147483647'	Convert seconds since the epoch (1970-01-01 UTC) to date
• TZ='America/Los_Angeles' date	What time is it on west coast of US (use tzselect to find TZ)
• date --date='TZ="America/Los_Angeles" 09:00 next Fri'	What's the local time for 9AM next Friday on west coast US
locales	
• printf "%d\n" 1234	Print number with thousands grouping appropriate to locale
• BLOCK_SIZE=\1 ls -l	Use locale thousands grouping in ls. See also l
• echo "I live in `locale territory`"	Extract info from locale database
• LANG=en_IE.utf8 locale int_prefix	Lookup locale info for specific country. See also ccodes
• locale -kc \$(locale sed -n 's/(LC_.\{4,\})=.*\1/p') less	List fields available in locale database
recode (Obsoletes iconv, dos2unix, unix2dos)	
• recode -l less	Show available conversions (aliases on each line)
recode windows-1252.. file_to_change.txt	Windows "ansi" to local charset (auto does CRLF conversion)
recode utf-8/CRLF.. file_to_change.txt	Windows utf8 to local charset
recode iso-8859-15..utf8 file_to_change.txt	Latin9 (western europe) to utf8
recode ../b64 < file.txt > file.b64	Base64 encode
recode /qp.. < file.qp > file.txt	Quoted printable decode
recode ../HTML < file.txt > file.html	Text to HTML
• recode -lf windows-1252 grep euro	Lookup table of characters
• echo -n 0x80 recode latin-9/x1..dump	Show what a code represents in latin-9 charmap
• echo -n 0x20AC recode ucs-2/x2..latin-9/x	Show latin-9 encoding
• echo -n 0x20AC recode ucs-2/x2..utf-8/x	Show utf-8 encoding
CDs	
gzip < /dev/cdrom > cdrom.iso.gz	Save copy of data cdrom
mkisofs -V LABEL -r dir gzip > cdrom.iso.gz	Create cdrom image from contents of dir
mount -o loop cdrom.iso /mnt/dir	Mount the cdrom image at /mnt/dir (read only)

wodim dev=/dev/cdrom blank=fast	Clear a CDRW
gzip -dc cdrom.iso.gz wodim -tao dev=/dev/cdrom -v -data -	Burn cdrom image (use --prcap to confirm dev)
cdparanoia -B	Rip audio tracks from CD to wav files in current dir
wodim -v dev=/dev/cdrom -audio -pad *.wav	Make audio CD from all wavs in current dir (see also cdrdao)
oggenc --tracknum=\$track track.cdda.wav -o track.ogg	Make ogg file from wav file
disk space (See also FSlint)	
• ls -lSr	Show files by size, biggest last
• du -s * sort -k1,1rn head	Show top disk users in current dir. See also dutup
• du -hs /home/* sort -k1,1h	Sort paths by easy to interpret disk usage
• df -h	Show free space on mounted filesystems
• df -i	Show free inodes on mounted filesystems
• fdisk -l	Show disks partitions sizes and types (run as root)
• rpm -q -a --qf '%10{SIZE}\t%{NAME}\n' sort -k1,1n	List all packages by installed size (Bytes) on rpm distros
• dpkg-query -W -f='\${Installed-Size;10}\t\${Package}\n' sort -k1,1n	List all packages by installed size (KBytes) on deb distros
• dd bs=1 seek=2TB if=/dev/null of=ext3.test	Create a large test file (taking no space). See also truncate
• > file	truncate data of file or create an empty file
monitoring/debugging	
• tail -f /var/log/messages	Monitor messages in a log file
• strace -c ls >/dev/null	Summarise/profile system calls made by command
• strace -f -e open ls >/dev/null	List system calls made by command
• strace -f -e trace=write -e write=1,2 ls >/dev/null	Monitor what's written to stdout and stderr
• ltrace -f -e getenv ls >/dev/null	List library calls made by command
• lsof -p \$\$	List paths that process id has open
• lsof ~	List processes that have specified path open
• tcpdump not port 22	Show network traffic except ssh. See also tcpdump_not_me
• ps -e -o pid,args --forest	List processes in a hierarchy
• ps -e -o pcpu,cpu,nice,state,cputime,args --sort pcpu sed '/^ 0.0 /d'	List processes by % cpu usage
• ps -e -orss=,args= sort -b -k1,1n pr -TW\$COLUMNS	List processes by mem (KB) usage. See also ps_mem.py
• ps -C firefox-bin -L -o pid,tid,pcpu,state	List all threads for a particular process
• ps -p 1,\$\$ -o etime=	List elapsed wall time for particular process IDs
• watch -n.1 pstree -Uacp \$\$	Display a changing process subtree
• last reboot	Show system reboot history
• free -m	Show amount of (remaining) RAM (-m displays in MB)
• watch -n.1 'cat /proc/interrupts'	Watch changeable data continuously
• udevadm monitor	Monitor udev events to help configure rules
system information (see also sysinfo) ('#' means root access is required)	
• uname -a	Show kernel version and system architecture
• head -n1 /etc/issue	Show name and version of distribution
• cat /proc/partitions	Show all partitions registered on the system
• grep MemTotal /proc/meminfo	Show RAM total seen by the system
• grep "model name" /proc/cpuinfo	Show CPU(s) info
• lspci -tv	Show PCI info
• lsusb -tv	Show USB info
• mount column -t	List mounted filesystems on the system (and align output)
• grep -F capacity: /proc/acpi/battery/BAT0/info	Show state of cells in laptop battery
# dmidecode -q less	Display SMBIOS/DMI information
# smartctl -A /dev/sda grep Power_On_Hours	How long has this disk (system) been powered on in total
# hdparm -i /dev/sda	Show info about disk sda
# hdparm -tT /dev/sda	Do a read speed test on disk sda
# badblocks -s /dev/sda	Test for unreadable blocks on disk sda
interactive (see also linux keyboard shortcuts)	
• readline	Line editor used by bash, python, bc, gnuplot, ...
• screen	Virtual terminals with detach capability, ...
• mc	Powerful file manager that can browse rpm, tar, ftp, ssh, ...
• gnuplot	Interactive/scriptable graphing
• links	Web browser
• xdg-open .	open a file or url with the registered desktop application