Command	**Description**
• grep . /proc/sys/net/ipv4/*	List the contents of flag files
• set | grep $USER	Search current environment
• tr '\0' '\n' < /proc/$$/environ	Display the *startup* environment for any process
• echo $PATH | tr : '\n'	Display the $PATH one per line
• kill -0 $$ && echo process exists and can accept signals	Check for the existence of a process (pid)
• find /etc -readable | xargs less -K -p'*ntp' -j $((${LINES:-25}/2))	Search paths and data with full context. Use **n** to iterate
Low impact admin	
# apt-get install "package" -o Acquire::http::Dl-Limit=42 \ -o Acquire::Queue-mode=access	Rate limit apt-get to 42KB/s
echo 'wget url' | at 01:00	Download url at 1AM to current dir
# apache2ctl configtest && apache2ctl graceful	Restart apache if config is OK
• nice openssl speed sha1	Run a low priority command (openssl benchmark)
• renice 19 -p $$; ionice -c3 -p $$	Make shell (script) low priority. Use for non interactive tasks
Interactive monitoring	
• htop -d 5	Better top (scrollable, tree view, lsof/strace integration, ...)
• iotop	What's doing I/O
# watch -d -n30 "nice ps_mem.py | tail -n $((${LINES:-12}-2))"	What's using RAM
# iftop	What's using the network. See also iptraf
# mtr www.pixelbeat.org	ping and traceroute combined
Useful utilities	
• pv < /dev/zero > /dev/null	Progress Viewer for data copying from files and pipes
• wkhtml2pdf http://.../linux_commands.html linux_commands.pdf	Make a pdf of a web page
• timeout 1 sleep 3	run a command with bounded time. See also timeout
Networking	
• python -m SimpleHTTPServer	Serve current directory tree at http://$HOSTNAME:8000/
• openssl s_client -connect www.google.com:443 </dev/null 2>&0 | openssl x509 -dates -noout	Display the date range for a site's certs
• curl -I www.pixelbeat.org	Display the server headers for a web site
# lsof -i tcp:80	What's using port 80
# httpd -S	Display a list of apache virtual hosts
• vim scp://user@remote//path/to/file	Edit a remote file directly in vim
• curl -s http://www.pixelbeat.org/pixelbeat.asc | gpg --import	Import a gpg key from the web
• tc qdisc add dev lo root handle 1:0 netem delay 20msec	Add 20ms latency to loopback device (for testing)
• tc qdisc del dev lo root	Remove latency added above
Notification	
• echo "DISPLAY=$DISPLAY xmessage cooker" | at "NOW +30min"	Popup reminder
• notify-send "subject" "message"	Display a gnome popup notification
echo "mail -s 'go home' P@draigBrady.com < /dev/null" | at 17:30	Email reminder
uuencode file name | mail -s subject P@draigBrady.com	Send a file via email
ansi2html.sh | mail -a "Content-Type: text/html" P@draigBrady.com	Send/Generate HTML email
Better default settings (useful in your .bashrc)	
# tail -s.1 -f /var/log/messages	Display file additions more responsively
• seq 100 | tail -n $((${LINES:-12}-2))	Display as many lines as possible without scrolling
# tcpdump -s0	Capture full network packets
Useful functions/aliases (useful in your .bashrc)	
• md () { mkdir -p "$1" && cd "$1"; }	Change to a new directory
• strerror() { python -c "import os; print os.strerror($1)"; }	Display the meaning of an errno
• plot() { { echo 'plot "-"' "$@"; cat; } | gnuplot -persist; }	Plot stdin. (e.g: • seq 1000 | sed 's/.*/s(&)/' | bc -l | plot)
• hili() { e="$1"; shift; grep --col=always -Eih "$e|$" "$@"; }	highlight occurences of expr. (e.g: • env | hili $USER)
• alias hd='od -Ax -tx1z -v'	Hexdump. (usage e.g.: • hd /proc/self/cmdline | less)
• alias realpath='readlink -f'	Canonicalize path. (usage e.g.: • realpath ~/../$USER)
Multimedia	
• DISPLAY=:0.0 import -window root orig.png	Take a (remote) screenshot
• convert -filter catrom -resize '600x>' orig.png 600px_wide.png	Shrink to width, computer gen images or screenshots
mplayer -ao pcm -vo null -vc dummy /tmp/Flash*	Extract audio from flash video to audiodump.wav
ffmpeg -i filename.avi	Display info about multimedia file
• ffmpeg -f x11grab -s xga -r 25 -i :0 -sameq demo.mpg	Capture video of an X display
DVD	
for i in $(seq 9); do ffmpeg -i $i.avi -target pal-dvd $i.mpg; done	Convert video to the correct encoding and aspect for DVD
dvdauthor -odvd -t -v "pal,4:3,720xfull" *.mpg;dvdauthor -odvd -T	Build DVD file system. Use 16:9 for widescreen input
growisofs -dvd-compat -Z /dev/dvd -dvd-video dvd	Burn DVD file system to disc

Unicode		
•	python -c "import unicodedata as u; print u.name(unichr(0x2028))"	Lookup a unicode character
•	uconv -f utf8 -t utf8 -x nfc	Normalize combining characters
•	printf '\300\200' | iconv -futf8 -tutf8 >/dev/null	Validate UTF-8
•	printf 'ŨTF8\n' | LANG=C grep --color=always '[^ -~]\+'	Highlight non printable ASCII chars in UTF-8
•	fc-match -s "sans:lang=zh"	List font match order for language and style
Development		
•	gcc -march=native -E -v -</dev/null 2>&1|sed -n 's/.*-mar/-mar/p'	Show autodetected gcc tuning params. See also gcccpuopt
•	for i in $(seq 4); do { [$i = 1] && wget http://url.ie/6lko -qO-|| ./a.out; } | tee /dev/tty | gcc -xc - 2>/dev/null; done	Compile and execute C code from stdin
•	cpp -dM /dev/null	Show all predefined macros
•	echo "#include <features.h>" | cpp -dN | grep "#define __USE_"	Show all glibc feature macros
	gdb -tui	Debug showing source code context in separate windows
Extended Attributes (Note you may need to (re)mount with "acl" or "user_xattr" options)		
•	getfacl **.**	Show ACLs for file
•	setfacl -m u:nobody:r **.**	Allow a specific user to read file
•	setfacl -x u:nobody **.**	Delete a specific user's rights to file
	setfacl --default -m group:users:rw- dir/	Set umask for a for a specific dir
	getcap file	Show capabilities for a program
	setcap cap_net_raw+ep your_gtk_prog	Allow gtk program raw access to network
•	stat -c%C **.**	Show SELinux context for file
	chcon ... file	Set SELinux context for file (see also restorecon)
•	getfattr -m- -d **.**	Show all extended attributes (includes selinux,acls,...)
•	setfattr -n "user.foo" -v "bar" **.**	Set arbitrary user attributes
BASH specific		
•	echo 123 | tee >(tr 1 a) | tr 1 b	Split data to 2 commands (using process substitution)
	meld local_file <(ssh host cat remote_file)	Compare a local and remote file (using process substitution)
Multicore		
•	taskset -c 0 nproc	Restrict a command to certain processors
•	find -type f -print0 | xargs -r0 -P$(nproc) -n10 md5sum	Process files in parallel over available processors
	sort -m <(sort data1) <(sort data2) >data.sorted	Sort separate data files over 2 processors