

data manipulation

awk	awk -F ":" 'BEGIN {OFS="\t"} /r..t/ { \$2 = "" ; print "Line #", \$NR, \$0 }' /etc/passwd -F Field separator (or variable IFS) \$0 The current line, after substitutions	
cut	cut -d: -f6 /etc/passwd top cut -c8-12,66-	Set the delimiter to ':' and display 6 th field Display pid and user time for each process
sed	sed '4,7 d' test.txt sed '/^#/ d' test.txt sed '/pattern/q' file sed '1,/pattern/d' file sed -re 's/\t/ /g'	Delete lines 4 to 7 Delete lines which begin by # Dump file until pattern Dump file after pattern Substitute tabs by 4 spaces
tr	tr -d '\r' < dos.txt tr "[a-z]" "[A-Z]" < fic1	A simple unix2dos command Display fic1 in uppercase
head / tail	head -n 10 /etc/passwd tail -f file.log	Show the 10 first lines Continuously display file.log
sort	sort -t':' -k2n -t Field separator -k Key position and type (d lexicographic, n num.)	-f Case insensitive -b Ignore blanks
read	-r do not allow backslashes to escape any characters -d delimiter	
uniq	-d Display multiple occurrences -c Count multiple occurrences	
grep	grep -line_buffered -e pattern1 -e pattern2 -C 3 -r * -A Display n lines before matching string -B Display n lines after matching string -C Display n lines around matching string -e Allow multiple regexp -E Extended regexp -f File which contains patterns	-i Case insensitive search -l Display only file name -o Show only the matching part -r Look a directory -v Filter out lines that match regexp -w Look for whole word

loop control

for	for i in "a" "b" "c"; do echo \$i ; done for i in \$(ls) ; do echo \$i ; done for i in 1 2 4 9 do echo \$((\$i * \$i)) done	Iterate on a list Take output of external program as list Numerical multi-line loop with operation
while	while read -r filename do echo "Print: \$filename" cat "\$filename" done < filenames.txt	Display a list of files
xargs	xargs --arg-file filenames.txt -I % sh -c "echo %; cat %" -n Number of args to be sent each time -I Substitution pattern -a File to be used	

redirections

> 2> &>	ls > /tmp/stdout.txt ls > /dev/tcp/host/port (echo "my directory" ; ls) > ls.txt { echo "my directory" ; ls } > ls.txt grep "pattern" * > search.txt 2> /dev/null head * 2>&1 > /dev/null grep ydk python -m SimpleHTTPServer 80 &> /dev/null &	Redirect stdout to file /tmp/stdout.txt Redirect to tcp flow Redirect multiple commands using a sub-shell Redirect multiple commands without a sub-shell Suppress stderr Pipe stderr only (order matters) Redirect stderr and stdout to /dev/null (bash only)
<	cat </dev/tcp/time.nist.gov/13 while read ln; do echo \$ln ; done < in.txt while read -r proc do echo "Proc: \$proc" done < <(ps cut -c25-)	Read current time from daytime protocol A kind of cat <(..) group input process
netcat	-l (-L) Listen mode once (persistent) -p port -e Program to be executed	