



ShellScripters ToolBox

The toolbox for every shell-scripter or everyone just starting scripting at the linux/unix command-line.

user interaction	2
file operations	3
datastream operations	5
searching	6
users & groups	7
permissions	8
process-management	9
network operations	10
misc tools	11

handcrafted by robert wohlfahrt :-)

user interaction

echo print out a message to the console

```
echo "Hello Shellscripter!"  
echo -e "a\ttab\tlimited\tlist"
```

cat generate a multiline-output

```
cat << END  
Hello Shellscripter  
This is a multiline output :-)  
END
```

printf generate a more formatted output

```
printf "pricetag: %05.2f$\n" "9.5"
```

read read user-input into a variable

```
read -p "What's your name again? " NAME
```

dialog read user-input with a nicer user-interface

```
dialog --inputbox "What's your name again?" 0 0  
dialog --calendar "What's your birthday?" 0 0
```

*perhaps you need to install the package "dialog"

select generate a text-only selection-menu

```
select FILE in *.sh ; do  
echo "you have selected $FILE"  
done
```

file operations 1/2

touch generate a new file or update the timestamp of an existing one

```
touch /tmp/newfile.txt
```

echo generate a new file with content or append to an existing one

```
echo "new file content" > newfile.txt
```

```
echo "content to append" >> newfile.txt
```

mktemp generate a new file with a unique, hard to predict filename

```
mktemp /tmp/mydata_XXXXXX
```

mkdir create an empty directory

```
mkdir /tmp/pictures
```

```
mkdir -p ~/pictures/2019/summer
```

ln create links

```
ln -s /etc/passwd ~/passwd
```

cat print out the content of a file

```
cat /etc/passwd
```

head print out only the first lines of a file

```
head /etc/passwd
```

tail print out only the last lines of a file

```
tail /etc/passwd
```

dd print out the content of a file, byte-wise or block-wise

```
dd if=/etc/passwd bs=1 count=128
```

```
dd if=/dev/sda of=/tmp/mbr.tmp bs=512 count=1
```

wc count the lines, words and characters within a file

```
wc -l /etc/passwd
```

test check a file for existence and type, check for permissions

```
test -f /etc/passwd
```

```
test -w /tmp
```

file operations 2/2

file	get the type of a file <code>file /boot/initrd</code> <code>file -s /dev/sda1</code>
stat	get the meta-information of a file, print them out formatted <code>stat --printf "last-modified: %z\n" /etc/passwd</code>
mv	move or rename a file <code>mv myscript.sh /usr/local/bin</code> <code>mv image.jpg image.jpeg</code>
rm	remove a file or directory <code>rm -i ~/old_file.sh</code> <code>rm -r ~/tmp/</code>
rmdir	remove a directory, empty directories only <code>rmdir ~/pictures/2019/summer</code>
cut	print out only parts of the lines of a file <code>cut -d : -f 1 /etc/passwd</code>
ls	list and sort files within a directory <code>ls -ltr /data</code>
grep	search for lines within a file containing a search-pattern <code>grep root /etc/passwd</code>
find	search for files recursively by a given criteria <code>find /etc -mmin -60</code>
sed	search and replace patterns within a file <code>sed -e 's/NAME/Jon Smith/g' address.txt > address_new.txt</code> <code>sed -i -e 's/DOMAIN/example.com/g' website.conf</code>
md5sum	generate a md5-hash as a fingerprint for the content of a file <code>md5sum /etc/passwd</code>
sha1sum	generate a sha1-hash as a fingerprint for the content of a file <code>sha1sum /etc/passwd</code>

datastream operations 1/2

cat	print the input-datastream while marking special characters <code>ls -l cat -A</code>
tac	print the input-datastream while reversing the lines <code>ls -l tac</code>
cut	cut the datastream into parts using a tab or character as a delimiter <code>echo "Hello Shellscripter" cut -d " " -f 1</code>
head	print out only the first lines of a datastream <code>ls -S head -n 1</code>
tail	print out only the last lines of a datastream <code>df -h tail -n +2</code>
read	read one single line from the input-datastream into a variable <code>getent passwd (read FIRSTLINE; echo "line: \$FIRSTLINE")</code>
sort	sort the input datastream <code>getent passwd sort</code> <code>du -sh * sort -h</code>
uniq	remove or find duplicates on the sorted input datastream <code>sort data.txt uniq</code> <code>sort /etc/profile uniq -D</code>
tr	substitute or delete single characters on the input-datastream <code>echo "Hello Shellscripter" tr " " "_"</code> <code>echo "Hello Shellscripter" tr -d eio</code>
wc	count the lines, words or characters of the input datastream <code>echo "Hello Shellscripter" wc -w</code> <code>ls -l wc -l</code>
rev	reverse the characters in each input line <code>echo "Hello Shellscripter" rev</code>

datastream operations 2/2

tee	print out the input datastream while logging it into a file <code>find -name "*.sh" tee filenames.txt sort</code>
xargs	convert the input datastream to commandline parameters <code>find -name "*.sh" xargs -i cp {} /tmp/backup/</code>
sed	print out the edited input-datastream, e.g. do substitutions <code>getent passwd sed 's/bash/ksh/g'</code>
awk	print out certain fields from the input datastream <code>df -h tail -n +2 awk '{print \$4 " free on " \$6}'</code>

searching

which	search for an executable file in the search path \$PATH <code>which sudo</code>
grep	search a file or datastream for lines matching a given pattern <code>grep -lR root /etc</code> <code>ps ax grep apache2</code>
find	search for files based on external criteria <code>find /etc -name "pass*"</code> <code>find /usr/bin -type f -perm -4001</code>

users & groups

getent	get all known users or groups on a system <code>getent passwd</code> <code>getent group</code>
useradd	create a new user <code>useradd -m exampleuser</code>
usermod	modify an existing user <code>usermod -s /bin/bash exampleuser</code>
userdel	remove a user from the system <code>userdel -r exampleuser</code>
passwd	set or change the password of a user <code>passwd exampleuser</code>
groupadd	create a new group <code>groupadd examplegroup</code>
groupmod	modify an existing group <code>groupmod -n newgroupname examplegroup</code>
groupdel	remove a group from the system <code>groupdel examplegroup</code>
w	show all currently logged-in users <code>w</code>
who	show all currently logged-in users <code>who</code>

permissions

test	test permissions of the current user for a file <pre>test -r /etc/passwd; echo \$? test -w /dev/sda; echo \$? test -x ~/myscript.sh; echo \$?</pre>
chown	change the owner and the group of a file <pre>chown exampleuser /tmp/newfile.txt chown -R www-data:www-data /srv/www/htdocs/</pre>
chgrp	change the ownergroup of a file <pre>chgrp admingroup /var/lib/somefile chgrp -R www-data /srv/www/htdocs</pre>
chmod	change the permissions for a file <pre>chmod +x /tmp/myscript.sh chmod u=rw,g=r,o= ~/list.txt chmod 640 ~/list.txt</pre>
setfacl	modify acls of a file <pre>setfacl -m user:www-data:r /srv/www/htdocs/index.html setfacl -R -m user:www-data:rx /srv/www/htdocs setfacl -x user:www-data /srv/www/htdocs/index.html</pre>
getfacl	get the acls of a file <pre>getfacl /srv/www/htdocs/index.html getfacl -R /data > saved_acls.txt</pre>

process-management

ps	get a list of running processes <code>ps ax</code>
kill	send a signal to a process or terminate a process identified by its PID <code>kill 1234</code> <code>kill -1 1234</code>
killall	send a signal to a process or terminate a process identified by its command <code>killall httpd</code> <code>killall -1 httpd</code>
wait	wait for a background-process to terminate <code>sleep 100 &</code> <code>echo "start waiting ..."</code> <code>wait \$! && echo "waiting is over :)"</code>
nice	start a process with a modified priority <code>nice -n 19 some_heavy_command</code>
renice	change the priority of a running process <code>renice 19 1234</code>
service	control services on most linux distributions <code>service apache2 restart</code>
systemctl	control services on modern linux-distributions with systemd <code>systemctl restart apache2</code>

network-operations

ping	send icmp echo-requests to a system to check network availability <code>ping -c 3 8.8.8.8</code>
netstat	list all active or listening ports on the local system <code>netstat -nltup</code> <code>netstat -natp</code>
nmap	scan for network-availability or open ports of a remote system <code>nmap 192.168.100.10</code>
nc	send a datatream over the network <code>echo "hello remote-system" nc 192.168.100.1 1234</code> <code>nc -l -p 1234 > received_data</code>
ssh	execute commands on a remote system, connecting via ssh <code>ssh user@remote-host "df -h"</code> <code>cat /etc/passwd ssh user@remote-host "cat > passwd_copy"</code>
ip	get and set the ip- and routing-configuration <code>ip addr show dev eth0</code> <code>ip route show</code>
arp	read or modify the arp-cache <code>arp -n</code>
host	resolve a hostname via dns and do other dns-queries <code>host www.google.de</code>
dig	resolve a hostname via dns and do other dns-queries <code>dig -t MX google.de @8.8.8.8</code>

misc tools

date print the current date and time, calculate with date and time

```
date +%Y_%m_%d"  
date -d "now + 2 hours"
```

shuf randomize a given list

```
echo -e "hello\nworld\nof\nscripting" | shuf  
shuf -i 10-20
```

sudo run a command as a different user

```
sudo -u www-data /usr/local/bin/cleanup.sh
```

This ToolBox was created after receiving a great number of the same request from my students: Again and again I was asked for a “list of common commands”. So I digged through an enormous number of shell-scripts I have written during the last ... wait more than 15 years (*uh* - really this long?). The I extracted all the tools used frequently within many of these scripts and compiled them into this tiny eBook.

So thanks for your interest and thank you for reading.

If you enjoy this ToolBox as much as I enjoyed creating it, feel free to let me know via robert@robert-wohlfahrt.de.

If you have any suggestions about further improvements - please also let me know via robert@robert-wohlfahrt.de. I appreciate every feedback.

happy scripting

robert