Cheatography

Command Line and Terminal navigation Cheat Sheet

by Kyngo (Kyngo) via cheatography.com/131267/cs/26366/

Basics

cd [path]	change your current directory to the specified one
cd ~	go to your home folder
cd -	go the the folder you were before
ls	list the contents of the directory
ls -lh	list the contents of the directory in a human-friendly format
cp [origin] [destination]	copies the given file wherever you want to
mv [origin] [destination]	moves or renames the given file
pwd	get the current directory you're in
mkdir [name]	create a folder
mkdir -p [name]	create a folder and all its parents, if needed
chmod 755 [name]	change a file's permissions - Allows the user to read, write and execute, and anyone else to just read and execute
chmod 400 [name]	change a file's permissions - Only the owner will be able to read the file
chown user:group [name]	changes the owners of a given file or folder
chown -R user:group [name]	changes the owners of a given file or folder, and all of its contents
touch [name]	creates a file with the given name
file [name]	reports the file type

Basics (cont)

rm [file]	removes a file
rm -rf [file]	removes a folder and all of its contents
cat [file]	prints a file's contents
tac [file]	prints a file's contents from bottom to top
sed	allows replacing of contents in files with regular expressions
grep [pattern] [file]	prints the contents of a given file that match the given pattern
tr -s [pattern]	replaces all concurrent duplicates of a given pattern
tr [pattern] [replacement]	replaces the given pattern with the given replacement string
tr -d [pattern]	removes the given pattern from a string
head [file]	prints the first lines of a given file
tail [file]	prints the last lines of a given file
cut -f [field] -d [separator]	allows you to print specific fields from an origin that have a given separator
uname	gets the OS kernel's name (Linux, Darwin...)
uname -m	gets the machine's architecture (if not within an emulator)
uname -r	gets the kernel version
uname -a	shows all the details of your UNIX OS

Basics (cont)

less [file]	prints a file using pagination
more [file]	same as `less [file]`
ln -s [source] [destination]	makes a symbolic link of a given source at the given destination
cal	prints a calendar on the terminal
date	reports the current date and time

Write (or append to) a file without an editor

```
cat > [file] << EOF
hello world
this is a file's content
blah blah blah
hello again
bye for now
EOF
```

In order to append to a file instead of replacing all of its contents, add two output cones instead of only one (>>).

Command pipeline concatenation example

```
curl https://developer.android.com/studio#downloads | grep ".dmg" | grep href | head -n1 | cut -f2 -d"=" | tr -d '"'
```

This command will:
- download the downlaods page for Android Studio
- find for the lines that contain ".dmg" within them
- filter again to get only those that contain "-href"
- filter again to get only the first occurrence
- split the result to get only the second field using = as a separator
- remove any double quotation marks on the string

By **Kyngo** (Kyngo)
cheatography.com/kyngo/
arnaumart.in

Published 29th January, 2021.
Last updated 22nd March, 2021.

Command pipeline concatenation example (cont)

The result should be a link that, when opened, will download the macOS installer for Android Studio.
Please note, if the website changes, this command may not work as is.

Manuals

Almost all programs on any Unix OS will have what's called a "manpage". This is an instruction manual with details on how to use a program.
In order to read the manual for a specific application, just type `man [application]` and you will be able to read how it works.
Press "Q" to close the manual when you're done.

sed examples

The `sed` command uses a string as parameter to determine what to oeprate, and can receive several more parameters to configure the behavior.
`sed -i 's/hello/hi/' file.txt` will replace the first instance of "hello" that the script can find at each line, and write the result at the same given file. To avoid overwriting, you can just remove the `-i` argument.
`sed -i 's/hello/hi/g' file.txt` will replace every instance of "hello" that exist in the file.
To apply the patterns from a file, use the `-f`parameter with a path to a file.
If you want to make a backup of the file, add a suffix for said file after the `-i` parameter. For example:
`sed -i".bkp" 's/hello/hi/g' file.txt` will generate a file named `file.txt.bkp` with the original contents.
Regular expressions can be applied to the pattern given to sed, as maybe you want to replace something that may not be an exact string, but rather a pattern of it.

Networks

ifconfig	Shows a general network brief
ip addr show	Same as `ifconfig` but on Linux
nmap [ip]/32	Scans the ports of the given IP
ping [host]	Send a sequence of ICMP packets to a host
whois [host]	Tells you information about the domain
dig [domain]	Tells you how a specific domain resolves
nslookup [domain]	The same as `dig [domain]`
host [domain]	Reports several types of records for a given domain
wget [url] -O [file]	Downloads the given URL to the specified file
curl [url] -o [file]	Downloads the given URL to the specified file
iftop	Allows you to monitor the network throughput (Linux)
netstat -tulpn	Shows which applications are using what ports (Linux)
sudo lsof -i -n -P	Shows which applications are using what ports (macOS)

Pipelines and operators

[command] > [file]	outputs the result of a command to a file
[command] >> [file]	outputs the result of a command to the end of a file
[command] < [file]	gets a file and prints its content as if it were you entering it

Pipelines and operators (cont)

[command] << [file]	appends a file's contents into the program
[command1] && [command2]	if command1 succeeds, command2 will be executed
[command1] || [command2]	if command1 fails, command2 will be executed
&	the process will be run in the background
!!	the last executed command
$?	the last command's exit code
[command1] | [command2]	sends the output of command1 to command2's input
[command] \	allows you to make a line break without executing the command
[command] 2>&1	redirects the command's stderr to stdout
`[command]`	runs the given command, and then runs the result as a command itself

Remote hosts

ssh [server]	connects to a server via SSH
ssh [server] -p [port]	
ssh [server] -i [certificate]	
scp [user]@[server]:[path] [local path]	copies a file from a remote server to your machine
telnet [host] [port]	makes a raw tcp connection to a given host and port
w	reports who's connected at the machine
who	same as `w`

Cheatography

Command Line and Terminal navigation Cheat Sheet

by Kyngo (Kyngo) via cheatography.com/131267/cs/26366/

Remote hosts (cont)

whoami	tells you your username

For SCP, you can upload from your machine to a remote server by changing the order of the commands. You can also use SSH's parameters with SCP (for port, you must use `-P` (capital)).

Environment variables

PATH	the directories where the shell will find for the command binaries
HOME	your home directory
UID	the user ID
EUID	the user ID running the command
SHELL	your current shell
PS1	your prompt's style
PWD	your current directory
RANDOM	random number
HOST	the host's name
LANG	the shell's language
TTY	your current session's TTY

Loops and decision taking

"For" loop

```
for i in {1..10}
do
echo $i
done
```

"While" loop (example of an infinite loop)

```
while [ true ]
do
echo hello
done
```

"Until" loop (do while)

```
until [ $IDX -eq 5 ]
do
echo $IDX
((IDX++))
```

Loops and decision taking (cont)

```
done
```

"If-else if-else" operator

```
if [ $UID -eq 0 ]
then
echo You are root
elif [ $UID -eq 1 ]
echo You are user with ID 1
else
echo You are NOT root
fi
```

Permission bits

0	---	Do nothing
1	--x	Execution
2	-w-	Write
3	-wx	Execute and write
4	r--	Read
5	r-x	Read and execute
6	rw-	Read and write
7	rwx	Read, write and execute

Here "r" stand for "read", "w" stands for "-write", and "x" stands for "execute". It may be useless to have permissions below 4, as you won't be able to read the file. A 0 permission is useful to fully restrict access to any other user.

Permissions are usually represented by three digits, and their meaning is the following: the first one represents the owner user of the file, the second number represents the owner group's permissions, and the last one represents everybody else's permissions.

Package Managers

apt	Debian, Ubuntu
yum	Amazon Linux, Red Hat
dnf	Red Hat, Fedora
pacman	Arch Linux
emerge	Gentoo
brew	macOS (Homebrew)

Package Managers (cont)

choco	Windows (Chocolatey)

Searching

find [path] -name [name pattern]	finds anything within a given path with a given pattern on its name
whereis [name]	tells you all the locations for a given binary name
which [name]	tells you the given binary name's path that will be run according to your PATH
locate [name]	tells you the location of any kind of file within your machine

Monitoring the OS

ps aux	prints a snapshot of all system processes
top	shows the processes running on the machine
htop	shows the processes running on the machine with some graphs for memory usage
df -h	prints the system's storage devices' usage
du -hs [path]	prints a folder or file size
free -m	reports used and free ram and swap
kill [pid]	kills a process with the given pid
kill -9 [pid]	forcefully kills a process with the given pid

By **Kyngo** (Kyngo)
cheatography.com/kyngo/
arnaumart.in

Published 29th January, 2021.
Last updated 22nd March, 2021.

Cheatography

Command Line and Terminal navigation Cheat Sheet

by Kyngo (Kyngo) via cheatography.com/131267/cs/26366/

Monitoring the OS (cont)

kill -l	lists the termination codes available to send the processes
pkill [process name]	kills any processes with the name provided (also works with -9 and other numeric signals)
xkill	kills a graphical process (Xorg)
lsblk	lists the drives, its partitions, and space (Linux)
blkid	shows the mount details of each volume (Linux)
lspci	lists PCI devices
lsusb	lists USB devices

Compression

tar xf [file]	extracts a tar file at the current path
tar cf [filename] [content]	creates a tar file with the given name from the given content
tar zcf [filename] [content]	creates a gzipped tar file with the given name from the given content
unzip [file]	unzips a .zip file
zip [filename] [content]	creates a .zip file with the given name from the given content

ls parameters

-l	detailed list
-h	human-readable file size, used with -l
-r	reversed
-d	list directories themselves
-a	include dotfiles (hidden files)
--si	list using base 1000 instead of 1024

ls parameters (cont)

-m	list with commas instead of tabs
-t	sort by newest to oldest

grep parameters

-i	case insensitive
-v	hide all matches
-r	recursive search
-e	regular expression pattern
-x	match entire line
-w	match entire word
-f [file]	use patterns from file
-I	do not search inside binary files
-R	recursive, even with symlinks

screen parameters

screen	creates a new screen session
screen -ls	lists the existing screen sessions
screen -r [name]	resume a given screen
`CTRL + A`	activates commands for the active screen session
`CTRL + A, D`	disconnects from the screen

sort parameters

-n	numeric
-r	reverse
-k [number]	specific field
-f	case insensitive

`ls -l | sort -n -k5` will list a folder's contents by its size, from the least to the most sized.

Niceness

Niceness is they way Unix OSes give priority to the applications running on the machine. A niceness of 19 means it's got the **least** priority, whereas a -20 priority means it's got the **most** priority.

Niceness (cont)

`renice 19 [pid]` will make the process with given PID have the least priority within the CPU. This means that, when the OS is running low on CPU resources, this process will be more ignored than one with a lower niceness number.

`renice -20 [pid]` will make this process have the most CPU availability even when resources are scarce.

S3 Commands (aws s3)

`ls s3://bucket/-file`	Lets you know if a file exists or not
`cp s3://bucket/file /path/-on/machine`	Downloads a file into your computer
`cp --recursive s3://bucket/-folder /path/on/-machine`	Downloads a folder and its content into your computer
`rm s3://bucket/-file`	Removes a remote file

All commands must begin by `aws s3`. Paths can be specified in both ways: from local to remote, or from remote to local. They can also work from remote to remote, but will use your device as a bridge.

By **Kyngo** (Kyngo)
cheatography.com/kyngo/
arnaumart.in

Published 29th January, 2021.
Last updated 22nd March, 2021.