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

Basics (cont)

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
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

Basics (cont)

head [file]	prints
tail [file]	prints
cut -f [field] -d [separator]	allow from sepa
uname	gets Darw
uname -m	gets not w
uname -r	gets
uname -a	show OS
less [file]	prints
more [file]	same
ln -s [source] [destination]	make sourc
cal	prints
date	repor

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 -s "https://developer.android.c-
om/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

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 "man-page". 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.

sed examples (c

Regular expressio
given to sed, as m
something that ma
rather a pattern of

Networks

ifconfig	Sho
ip addr show	Sar
nmap [ip]/32	Sca
ping [host]	Ser to a
whois [host]	Tell dor
dig [domain]	Tell res
nslookup [domain]	The
host [domain]	Rep a gi
wget [url] -O [file]	Dov spe
curl [url] -o [file]	Dov spe
iftop	Allo thro

Command Line and Terminal navigation Cheat Sheet

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

Networks (cont)

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
[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

Pipelines and operators (cont)

[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`
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)).

Env

PAT
HOI
UID
EUII
SHE
PS1
PWI
RAN
HOS
LAN
TTY

Loo

"For

```
for
do
ech
don
```

"Wh
loop

```
whi
do
ech
don
```

"Unt

```
unt
do
ech
((I
```

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)
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

Monitc

htop
df -h
du -hs [path]
free -m
kill [pid
kill -9 [pid]
kill -l
pkill [proces name]
xkill
lsblk
blkid
lspci
lsusb

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
-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

grep parameters (cont)

-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 w
priority within the CPU. This means that, whe
resources, this process will be more ignored
number.
`renice -20 [pid]` will make this process
even when resources are scarce.

S3 Commands (aws s3)

`ls s3://bucket/file`	
`cp s3://bucket/file /path/on/machine`	
`cp --recursive s3://bucket/folder /path/on/machine`	
`rm s3://bucket/file`	

All commands must begin by `aws s3`.
Paths can be specified in both ways: from loc
local. They can also work from remote to rem
a bridge.