

ssh

sshfs [USER@HOST]:[remot- e_directory] mountpoint [options]	mount remote drive in mountpoint
ssh -o ServerAliveInte- rval=X -o ServerAliveCoun- tMax=Y [HOST]	Maintaing ssh conection for X seconds and Y tries
ssh-copy-id -i ~/.ssh/id_rsa [USER@HOST]	add autentication via ssh, not .pub file
autossh -N -f -L localh- ost:[PORTU]:localhost:[P- ORTR] [USER@HOST]	map local port PORTU to PORTR on USER@HOST with autoreconnect

operations on files & drives

cp -r \$(<file_list- ed.txt) outdir	copies files & directories listed in file_listed.txt to outdir
xargs -a file_list- ed.txt cp -r -t outdir	same as above
tar -v)czf file.t- ar.gz /path/to_compres	create -cf archive -xf for extract
unzip file.zip -d file	unzip file.zip to file directory (must exists before)
mkdir -p dir	make parent directories as needed, no error
rm -r !(*.ext)	delete everything except files with extension .ext
du -hs dir	check size of dir use -h for GB or -m for MB
basename path/basenam- e.ext .ext	get basename from path without ext
grep -i -r -E 'patte- rn1 pattern2 pattern3'	multiple patterns with grep -i case insensitive -r recursive
find -i)name "pattern" -type d/f	list f files or d directories with matching pattern

Statements

while read line; do [Command] done < input.file	Loop over file content
for file in /*.extention; do [Command] done	loop over files in directory
if test -f file.txt; then [Command] fi	check if file exists -f, - d for directory

Statements (cont)

if [-f file.txt]; then [Command] fi	same different syntax
if [-z "\$S1"]; then fi	checks if arg \$1 is empty
function function_name { <commands> }	function declaration, \$1 \$2 - argument names
"\$?"	return exit status from last command
\$(expr)	evaluate expr and store it's result

system

cat /proc/cpuinfo #or lscpu	cpu informations
uname -a	OS informations
ss -tpln	show forwarded ports
lsblk -S	list block devices -S for minimal view
dd if=/path/to/image.iso of=/drive/location bs=4M && sync	create bootable drive from image.iso
sudo mount [drive] [mountpoint]	mounts drive at mountpoint
kill -9 \$(pgrep -f [pattern])	terminate all processes matching pattern

git

git rm -r -- cached <path/fil- e.ext>	remove untracked files
git clean -d - (n/f)	remove undone changes add: -n to show files to remove or -f to remove them
git fetch --all [branch_name]	copies all changes from last pull/clone on current branch or given branch name
git log --not -- remotes	all commits on all branches that aren't pushed yet
git checkout -b <new_branch>	change & create new branch
git switch -c <new_branch>	same as above