

# ETH ROBOTICS SUMMER SCHOOL

## LINUX & ROS CHEAT SHEET

AUTHOR: YIZE WANG  
LAST UPDATED: MAY 5, 2020

## File Commands

ls	list contents of files and directories
ls -a	list hidden files and directories
cd \$DIR	change working directory to \$DIR
cd	change working directory to home
mkdir \$DIR	create a directory named \$DIR
pwd	print working directory
rm \$FILE	remove \$FILE
rm -r \$DIR	remove \$DIR
rm -f \$FILE	force remove \$FILE
rm -rf \$DIR	force remove \$DIR
cp \$FILE1 \$FILE2	copy \$FILE1 to \$FILE2
cp -r \$DIR1 \$DIR2	copy \$DIR1 to \$DIR2
mv \$FILE1 \$FILE2	move \$FILE1 to \$FILE2
ln -s \$FILE \$LINK	create symbolic link \$LINK to \$FILE
touch \$FILE	create \$FILE
cat \$FILE	view content of \$FILE
cat > \$FILE	write input into \$FILE
more \$FILE	print contents of \$FILE
head \$FILE	print the first 10 lines of \$FILE
tail \$FILE	print the last 10 lines of \$FILE

## System Info

env	print environment variables
date	print system date and time
cal	print current month calendar
uptime	print system uptime
w	print online users
whoami	print current user
finger \$USER	print information about \$USER
uname -a	print kernel information
cat /proc/cpuinfo	print cpu information
cat /proc/meminfo	print memory information
man \$COMMAND	print user manual of \$COMMAND
df	print disk usage
du	print directory space usage
free	print memory and swap usage
whereis \$APP	print locations of \$APP
which \$APP	print print executable file of \$APP

## Compression

tar cf \$FILE.tar \$FILES	convert \$FILES into \$FILE.tar
tar xf \$FILE.tar	extract files from \$FILE.tar
tar czf \$FILE.tar.gz \$FILES	compress \$FILES into \$FILE.tar.gz using Gzip
tar xzf \$FILE.tar.gz	extract files from \$FILE.tar.gz using Gzip
gzip \$FILE	compress \$FILE and rename it as \$FILE.gz
gzip -d \$FILE.gz	decompress \$FILE.gz back to \$FILE

## Network

ip address	print all internet protocol addresses
ping \$HOST	ping \$HOST and print results
whois \$DOMAIN	print information about \$DOMAIN
dig \$DOMAIN	print DNS of \$DOMAIN
dig -x \$HOST	reverse lookup \$HOST
wget \$FILE	download \$FILE

## Terminator

Ctrl+Alt+T	launch a new terminal
Ctrl+C	kill the current process
Ctrl+Z	suspend the current process
fg	resume the suspended process in foreground
bg	resume the suspended process in background
Ctrl+D	log out of the current session
Ctrl+W	erase one word in the current line
Ctrl+U	erase the whole current line
Ctrl+R	reverse search in the previous commands
!!	execute the last command
exit	log out of the current session
Ctrl+Shift+E	split the window vertically vertically
Ctrl+Shift+O	split the window horizontally

## Package

apt-get update	synchronize package index files from sources
apt-get upgrade	install latest versions of installed packages
apt-get install \$PACKAGE	install \$PACKAGE
dpkg -i \$PACKAGE.deb	install a Debian package \$PACKAGE.deb
/configure	configure building settings
make	build the program from source code
make install	install the program

## Secure Shell (SSH)

ssh \$USER @ \$HOST	connect to \$HOST as \$USER
ssh \$IP_ADDRESS	connect to \$IP_ADDRESS
ssh -p \$PORT \$USER @ \$HOST	connect to \$HOST on \$PORT as \$USER
ssh-copy-id \$USER @ \$HOST	add the key to \$HOST as \$USER

## Searching

grep \$PATTERN \$FILE	search for \$PATTERN in \$FILE
grep -r \$PATTERN \$DIR	recursively search for \$PATTERN in \$DIR
\$COMMAND   grep \$PATTERN	search for \$PATTERN in \$COMMAND's output
locate \$FILE_NAME	find all files whose name contain \$FILE_NAME

## Git

git clone \$URL	clone the repository from \$URL
git status	print current branch status
git branch \$BRANCH	create a new branch named \$BRANCH
git checkout \$BRANCH	switch to the branch named \$BRANCH
git merge \$BRANCH	combine \$BRANCH into the current one
git fetch	download all history from GitHub
git merge	combine remote branches into local branch
git push	upload all local branch commits to GitHub
git pull	update local branch from GitHub
git log	list version history for current branch
git log --follow \$FILE	list version history for \$FILE
git show \$COMMIT	output content changes of \$COMMIT
git add \$FILE	stage \$FILE
git commit -m "\$MESSAGE"	commit staged file with \$MESSAGE
git reset \$FILE	reset \$FILE
git reset --hard	reset all uncommitted changes
git clean -fd	recursively force remove unstaged files

## .bashrc

## ROS Run

roscore	invoke the core of ros
roslaunch \$PACKAGE \$EXECUTABLE	run \$EXECUTABLE in \$PACKAGE
roslaunch \$PACKAGE \$LAUNCHFILE	launch \$LAUNCHFILE in \$PACKAGE

## ROS Node

<b>rostopic ping</b> <b>\$NODE</b>	test connectivity to <b>\$NODE</b>
<b>rostopic list</b>	list active nodes
<b>rostopic info</b> <b>\$NODE</b>	print information about <b>\$NODE</b>
<b>rostopic machine</b>	list nodes running on the machine
<b>rostopic kill</b> <b>\$NODE</b>	kill a running node

## ROS Topic

<b>rostopic list</b>	print information about active topics
<b>rostopic bw</b> <b>\$TOPIC</b>	display bandwidth used by <b>\$TOPIC</b>
<b>rostopic echo</b> <b>\$TOPIC</b>	print messages from <b>\$TOPIC</b>
<b>rostopic find</b> <b>\$TYPE</b>	find topics with <b>\$TYPE</b>
<b>rostopic hz</b> <b>\$TOPIC</b>	display publishing rate of <b>\$TOPIC</b>
<b>rostopic info</b> <b>\$TOPIC</b>	print information about <b>\$TOPIC</b>
<b>rostopic pub</b> <b>\$TOPIC</b>	publish data to <b>\$TOPIC</b>
<b>rostopic type</b> <b>\$TOPIC</b>	print type of <b>\$TOPIC</b>

## ROS Service

<b>rosservice list</b>	list active services
<b>rosservice call</b> <b>\$SERVICE</b> <b>\$ARGS</b>	call <b>\$SERVICE</b> with <b>\$ARGS</b>
<b>rosservice find</b> <b>\$TYPE</b>	find services with <b>\$TYPE</b>
<b>rosservice info</b> <b>\$SERVICE</b>	print information about <b>\$SERVICE</b>
<b>rosservice type</b> <b>\$SERVICE</b>	print type of <b>\$SERVICE</b>
<b>rosservice uri</b> <b>\$SERVICE</b>	print uri of <b>\$SERVICE</b>

## ROS Parameter

<b>rosparam list</b>	list all parameter names
<b>rosparam set</b> <b>\$PARAM</b> <b>\$VAL</b>	set <b>\$PARAM</b> to <b>\$VAL</b>
<b>rosparam get</b> <b>\$PARAM</b>	print value of <b>\$PARAM</b>
<b>rosparam load</b> <b>\$YAML</b>	load parameters from <b>\$YAML</b>
<b>rosparam dump</b> <b>\$YAML</b>	dump parameters to <b>\$YAML</b>
<b>rosparam delete</b> <b>\$PARAM</b>	delete <b>\$PARAM</b>

## ROS Bag

<b>rosbag record</b> <b>\$TOPIC</b>	record <b>\$TOPIC</b> into bag
<b>rosbag info</b> <b>\$BAG</b>	print content summary of <b>\$BAG</b>
<b>rosbag play</b> <b>\$BAG</b>	play back content of <b>\$BAG</b>
<b>rosbag check</b> <b>\$BAG</b>	check play-ability of <b>\$BAG</b> in current system
<b>rosbag compress</b> <b>\$BAG</b>	compress <b>\$BAG</b> using BZ2
<b>rosbag decompress</b> <b>\$BAG</b>	decompress <b>\$BAG</b> using BZ2

## ROS Catkin Workspace

<b>mkdir -p</b> <b>~/catkin_ws/src</b>	create a new catkin workspace
<b>cd</b> <b>~/catkin_ws</b>	
<b>catkin_make</b>	
<b>catkin build</b>	build the whole workspace
<b>catkin build</b> <b>\$PACKAGE</b>	build <b>\$PACKAGE</b>
<b>catkin clean</b>	clean the whole workspace
<b>catkin clean</b> <b>\$PACKAGE</b>	clear <b>\$PACKAGE</b>
remember to source <b>~/catkin_ws/devel/setup.bash</b>	

## ROS Package Structure

<b>package.xml</b>	manifest, dependencies and plugins
<b>CMakeLists.txt</b>	description of compilation procedure
<b>src/</b>	C and C++ source codes
<b>include/</b>	C and C++ header files
<b>scripts/</b>	Python and bash scripts
<b>config/</b>	ymal configuration files
<b>cfg/</b>	dynamics reconfigure scripts
<b>launch/</b>	launch files

## ROS Visualization

## TODO

ubuntu mono font for commands

tab auto-completion

tar explanation

is git reset unstage or reset?

simtime true

copyright

rostopic machine

column split

Cmakelist

ros launchfile

**Variable Declaration** VARIABLE stands for a variable whose name is VARIABLE. For example, FILE means a file named FILE.  
**sudo** sudo means super user do, elevated right granted

## ROS Visualization