

ETH ROBOTICS SUMMER SCHOOL LINUX & ROS CHEAT SHEET

AUTHOR: YIZE WANG
LAST UPDATED: MAY 7, 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
\$ gedit \$FILE	edit \$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

Linux Basics

path symbolic links	.	current directory
	..	parent directory
	~	home directory
	\	root directory
output direction	>	to a file (rewrite)
	>>	to a file (append)
		pipe the output of first command to the second

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

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

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

Searching

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

Git

\$ git clone \$URL	clone the repository from \$URL
\$ git status	print current branch status \$BRANCH
\$ 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

Miscellaneous

Hitting **Tab** while typing a command, file name, and option will auto-complete it.

sudo (superuser do) runs command with elevated privilege.

tar (tape archive) deal with tape drives backup.

Appending **--help** after a command will print command usage

When simulating in ROS, remember \$ **set use_sim_time true** and to append **--clock**.

ROS Catkin Workspace

\$ roscd \$PACKAGE	change directory to \$PACKAGE
\$ mkdir -p ~/catkin_ws/src	create a new catkin workspace
\$ cd ~/catkin_ws	
\$ catkin_make	
\$ catkin build	build the whole workspace
\$ catkin build \$PACKAGE	build \$PACKAGE
\$ catkin clean	clean the whole workspace
\$ catkin clean \$PACKAGE	clear \$PACKAGE

ROS Run

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

ROS Node

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

ROS Service

\$ rosservice list	list active services
\$ rosservice call \$SERVICE \$ARGS	call \$SERVICE with \$ARGS
\$ rosservice find \$TYPE	find services with \$TYPE
\$ rosservice info \$SERVICE	print information about \$SERVICE
\$ rosservice type \$SERVICE	print type of \$SERVICE
\$ rosservice uri \$SERVICE	print uri of \$SERVICE
\$ rossrv show \$TYPE	print structure of \$TYPE

ROS Topic

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

ROS Parameter

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

ROS Bag

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

ROS Package Structure

package.xml	manifest, dependencies and plugins
CMakeLists.txt	description of compilation procedure
src/	C and C++ source codes
build/	generated makefiles and support files
devel/	compiled binaries, libraries, headers
include/	C and C++ header files
scripts/	Python and bash scripts
config/	ymal configuration files
cfg/	dynamics reconfigure scripts
launch/	launch files

ROS Visualization Tools

\$ rviz	3D visualization of data and models
\$ gzclient	Gazebo GUI
\$ rqt	powerful GUI tool
\$ rqt_plot	simple and lightweight plotting
\$ rqt_bag	visualize content of a bag
\$ rqt_image_view	visualize camera images

TODO

use ubuntu font for commands
is git reset unstaged or reset?
rostopic machine
CMakeList
Eigen
ROS programming