community.linuxmint.com

The website for all Linux Mint users

- Home
-
- Community
    - Ideas
    - Tutorials
    - Hardware
    - Software
    - Countries
    - Users
    - Moderation
    - Chat room
-
- Testing
    - ISO Images
    - Teams

Login
Username

Password

□

Remember me
Forgot password
Login Register
Back

Written by: Cipi **Score:** 40
votes: 44
Format: Article

Simple and advanced Shell (Terminal) tutorial.

System:
Running kernel and system information:

```
# uname -a                       # Get the kernel version (and BSD version)
# lsb_release -a                 # Full release info of any LSB distribution
# cat /etc/debian_version        # Get Debian version
```

Use /etc/DISTR-release with DISTR= lsb (Ubuntu) /etc/issue.

```
# uptime                            # Show how long the system has been running + load
# hostname                          # system's host name
# hostname -i                       # Display the IP address of the host.
# man hier                          # Description of the file system hierarchy
# last reboot                       # Show system reboot history
```

Hardware Informations:

Kernel detected hardware:

```
# dmesg                          # Detected hardware and boot messages
# lsdev                          # information about installed hardware
# dd if=/dev/mem bs=1k skip=768 count=256 2>/dev/null | strings -n 8 # Read BIOS

# cat /proc/cpuinfo                   # CPU model
# cat /proc/meminfo                   # Hardware memory
# grep MemTotal /proc/meminfo       # Display the physical memory
# watch -n1 'cat /proc/interrupts'  # Watch changeable interrupts continuously
# free -m                             # Used and free memory (-m for MB)
# cat /proc/devices                   # Configured devices
# lspci -tv                  # Show PCI devices
# lsusb -tv                  # Show USB devices
# lshal                      # Show a list of all devices with their properties
# dmidecode                 # Show DMI/SMBIOS: hw info from the BIOS
```

Load, statistics and messages:

The following commands are useful to find out what is going on on the system.

```
# top                                 # display and update the top cpu processes
# mpstat 1                           # display processors related statistics
# vmstat 2                           # display virtual memory statistics
# iostat 2                           # display I/O statistics (2 s intervals)
# systat -vmstat 1                  # BSD summary of system statistics (1 s intervals)
# systat -tcp 1                     # BSD tcp connections (try also -ip)
# systat -netstat 1                # BSD active network connections
# systat -ifstat 1                  # BSD network traffic through active interfaces
# systat -iostat 1                  # BSD CPU and and disk throughput
# tail -n 500 /var/log/messages   # Last 500 kernel/syslog messages
# tail /var/log/warn                # System warnings messages see syslog.conf
```

Users:

```
# id                                         # Show the active user id with login and group
# last                                       # Show last logins on the system
# who                                        # Show who is logged on the system
# groupadd admin                           # Add group "admin" and user colin
# useradd -c "Colin Barschel" -g admin -m colin
# usermod -a -G                             # Add existing user to group (Debian)
# userdel colin                             # Delete user colin
# pw groupmod admin -m newmembe r       # Add a new member to a group
# pw useradd colin -c "Colin Barschel" -g admin -m -s /bin/tcsh
# pw userdel colin; pw groupdel admin
```

Kernel modules:

```
# lsmod                           # List all modules loaded in the kernel
# modprobe isdn                 # To load a module (here isdn)
```

Compile Kernel

```
# cd /usr/src/linux
# make mrproper                  # Clean everything, including config files
# make oldconfig                  # Reuse the old .config if existent
# make menuconfig               # or xconfig (Qt) or gconfig (GTK)
# make                            # Create a compressed kernel image
# make modules                 # Compile the modules
# make modules_install       # Install the modules
# make install                    # Install the kernel
# reboot
```

Repair grub:

So you broke grub? Boot from a live cd, [find your linux partition under /dev and use fdisk to find the linux partion] mount the linux partition, add /proc and /dev and use grub-install /dev/xyz. Suppose linux lies on /dev/sda4:

```
# mount /dev/sda6 /mnt               # mount the linux partition on /mnt
# mount --bind /proc /mnt/proc       # mount the proc subsystem into /mnt
# mount --bind /dev /mnt/dev          # mount the devices into /mnt
# chroot /mnt                            # change root to the linux partition
# grub-install /dev/sda                # reinstall grub with your old settings
```

Listing and PIDs:

Each process has a unique number, the PID. A list of all running process is retrieved with ps.

```
# ps -auxefw                       # Extensive list of all running process
```

However more typical usage is with a pipe or with pgrep:

```
# ps axww | grep cron
  586  ??  Is     0:01.48 /usr/sbin/cron -s
# ps axjf                           # All processes in a tree format
# ps aux | grep 'ss[h]'           # Find all ssh pids without the grep pid
# pgrep -l sshd                   # Find the PIDs of processes by (part of) name
# echo $$                          # The PID of your shell
# fuser -va 22/tcp                # List processes using port 22 (Linux)
# pmap PID                         # Memory map of process (hunt memory leaks) (Linux)
# fuser -va /home                # List processes accessing the /home partition
# strace df                         # Trace system calls and signals
# truss df                          # same as above
```

Signals/Kill:

Terminate or send a signal with kill or killall.

```
# kill -s TERM 4712              # same as kill -15 4712
# killall -1 httpd                 # Kill HUP processes by exact name
# pkill -9 http                     # Kill TERM processes by (part of) name
# pkill -TERM -u www            # Kill TERM processes owned by www
# fuser -k -TERM -m /home     # Kill every process accessing /home (to umount)
```

Important signals are:

1	HUP (hang up)
2	INT (interrupt)
3	QUIT (quit)
9	KILL (non-catchable, non-ignorable kill)
15	TERM (software termination signal)

Permissions:

Change permission and ownership with chmod and chown. The default umask can be changed for all users in /etc/profile for Linux. The default umask is usually 022. The umask is subtracted from 777, thus umask 022 results in a permission 0f 755.

```
1 --x execute                    # Mode 764 = exec/read/write | read/write | read
2 -w- write                      # For:      |--  Owner  --|   |- Group-|   |Oth|
4 r-- read
  ugo=a                          u=user, g=group, o=others, a=everyone
# chmod [OPTION] MODE[,MODE] FILE    # MODE is of the form [ugoa]*([-+=]([rwxXst]))
# chmod 640 /var/log/maillog              # Restrict the log -rw-r-----
# chmod u=rw,g=r,o= /var/log/maillog      # Same as above
# chmod -R o-r /home/*                    # Recursive remove other readable for all users
# chmod u+s /path/to/prog                 # Set SUID bit on executable (know what you do!)
# find / -perm -u+s -print                # Find all programs with the SUID bit
# chown user:group /path/to/file          # Change the user and group ownership of a file
# chgrp group /path/to/file               # Change the group ownership of a file
# chmod 640 `find ./ -type f -print`      # Change permissions to 640 for all files
# chmod 751 `find ./ -type d -print`      # Change permissions to 751 for all directories
```

Disk information:

```
# hdparm -I /dev/sda            # information about the IDE/ATA disk (Linux)
# fdisk /dev/ad2                # Display and manipulate the partition table
# smartctl -a /dev/ad2          # Display the disk SMART info
```

System mount points/Disk usage

```
# mount | column -t             # Show mounted file-systems on the system
# df                            # display free disk space and mounted devices
# cat /proc/partitions          # Show all registered partitions

# du -sh *                      # Directory sizes as listing
# du -csh                       # Total directory size of the current directory
# du -ks * | sort -n -r         # Sort everything by size in kilobytes
```

Who has which files opened:

This is useful to find out which file is blocking a partition which has to be unmounted and gives a typical error of:

```
# umount /home/
umount: unmount of /home            # umount impossible because a file is locking home
   failed: Device busy
# ls -lSr                             # Show files, biggest last
```

Find opened files on a mount point with fuser or lsof:

```
# fuser -m /home                # List processes accessing /home
# lsof /home

COMMAND   PID    USER   FD   TYPE DEVICE    SIZE     NODE NAME
tcsh    29029 eedcoba  cwd    DIR   0,18   12288  1048587 /home/cipi (cipi:/home)
lsof    29140 eedcoba  cwd    DIR   0,18   12288  1048587 /home/cipi (cipi:/home)
```

About an application:

```
ps ax | grep Xorg | awk '{print $1}'
3324
# lsof -p 3324
COMMAND   PID    USER   FD   TYPE DEVICE    SIZE    NODE NAME
Xorg    3324 root    0w   REG        8,6   56296      12492 /var/log/Xorg.0.log
```

About a single file:

```
# lsof /var/log/Xorg.0.log
COMMAND  PID USER   FD   TYPE DEVICE  SIZE  NODE NAME
Xorg    3324 root    0w   REG    8,6 56296 12492 /var/log/Xorg.0.log
```

Mount/remount a file system

For example the cdrom. If listed in /etc/fstab:

```
# mount /cdrom
# mount -t auto /dev/cdrom /mnt/cdrom           # typical cdrom mount command
# mount /dev/hdc -t iso9660 -r /cdrom           # typical IDE
# mount /dev/scd0 -t iso9660 -r /cdrom          # typical SCSI cdrom
# mount /dev/sdc0 -t ntfs-3g /windows           # typical SCSI
```

Entry in /etc/fstab:

```
/dev/cdrom   /media/cdrom  subfs noauto,fs=cdfss,ro,procuid,nosuid,nodev,exec 0 0
```

Add swap on-the-fly

Suppose you need more swap (right now), say a 2GB file /swap2gb .

```
# dd if=/dev/zero of=/swap2gb bs=1024k count=2000
# mkswap /swap2gb                               # create the swap area
# swapon /swap2gb                               # activate the swap. It now in use
# swapoff /swap2gb                              # when done deactivate the swap
# rm /swap2gb
```

Mount an SMB share

Suppose we want to access the SMB share myshare on the computer smbserver, the address as typed on a Windows PC is \\smbserver\myshare\. We mount on /mnt/smbshare. Warning> cifs wants an IP or DNS name, not a Windows name.

```
# smbclient -U user -I 192.168.16.229 -L //smbshare/       # List the shares
# mount -t smbfs -o username=winuser //smbserver/myshare /mnt/smbshare
# mount -t cifs -o username=winuser,password=winpwd //192.168.16.229/myshare /mnt/share
```

Additionally with the package mount.cifs it is possible to store the credentials in a file, for example /home/user/.smb:

```
username=winuser
password=winpwd
```

And mount as follow:

```
# mount -t cifs -o credentials=/home/user/.smb //192.168.16.229/myshare /mnt/smbshare
```

Mount an image:

```
# mount -t iso9660 -o loop file.iso /mnt            # Mount a CD image
# mount -t ext3 -o loop file.img /mnt                # Mount an image with ext3 fs
```

Create a memory file system:

A memory based file system is very fast for heavy IO application. How to create a 64 MB partition mounted on /memdisk:

```
# mount -t tmpfs -osize=64m tmpfs /memdisk
```

Disk performance:

Read and write a 1 GB file on partition ad4s3c (/home)

```
# time dd if=/dev/ad4s3c of=/dev/null bs=1024k count=1000
# time dd if=/dev/zero bs=1024k count=1000 of=/home/1Gb.file
# hdparm -tT /dev/hda      # Linux only
```

Networking:

```
# ethtool eth0                                  # Show the ethernet status (replaces mii-diag)
# ethtool -s eth0 speed 100 duplex full # Force 100Mbit Full duplex
# ethtool -s eth0 autoneg off # Disable auto negotiation
# ethtool -p eth1                               # Blink the ethernet led - very useful when supported
# ip link show                                   # Display all interfaces on Linux (similar to ifconfig)
# ip link set eth0 up                           # Bring device up (or down). Same as "ifconfig eth0 up"
# ip addr show                                  # Display all IP addresses on Linux (similar to ifconfig)
# ip neigh show                                 # Similar to arp -a
```

Ports in use:

Listening open ports:

```
# netstat -an | grep LISTEN
# lsof -i                              # List all Internet connections
# socklist                             # Display list of open sockets
# netstat -anp --udp --tcp | grep LISTEN
# netstat -tup                        # List active connections to/from system
# netstat -tupl                       # List listening ports from system
```

Firewall

Check if a firewall is running (typical configuration only):

```
# iptables -L -n -v                           # For status Open the iptables firewall
# iptables -P INPUT       ACCEPT     # Open everything
# iptables -P FORWARD     ACCEPT
# iptables -P OUTPUT      ACCEPT
# iptables -Z                                  # Zero the packet and byte counters in all chains
# iptables -F                                  # Flush all chains
# iptables -X                                  # Delete all chains
```

IP Forward for routing

Check and then enable IP forward with :

```
# cat /proc/sys/net/ipv4/ip_forward  # Check IP forward 0=off, 1=on
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

or edit /etc/sysctl.conf with:

```
net.ipv4.ip_forward = 1
```

Network Address Translation

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE    # to activate NAT
# iptables -t nat -A PREROUTING -p tcp -d 78.31.70.238 --dport 20022 -j DNAT \
--to 192.168.16.44:22          # Port forward 20022 to internal IP port ssh
# iptables -t nat -A PREROUTING -p tcp -d 78.31.70.238 --dport 993:995 -j DNAT \
--to 192.168.16.254:993-995     # Port forward of range 993-995
# ip route flush cache
# iptables -L -t nat           # Check NAT status
```

DNS

The DNS entries are valid for all interfaces and are stored in /etc/resolv.conf. The domain to which the host belongs is also stored in this file. A minimal configuration is:

```
nameserver 66.63.128.84
search cipi.net intern.lab
domain cipi.org
```

Check the system domain name with:

```
# hostname -d                # Same as dnsdomainname
```

DHCP

```
# dhcpcd -n eth0          # Trigger a renew (does not always work)
# dhcpcd -k eth0          # release and shutdown
```

The lease with the full information is stored in:
/var/lib/dhcpcd/dhcpcd-eth0.info

tar

The command tar (tape archive) creates and extracts archives of file and directories. The archive .tar is uncompressed, a compressed archive has the extension .tgz or .tar.gz (zip) or .tbz (bzip2). Do not use absolute path when creating an archive, you probably want to unpack it somewhere else. Some typical commands are:

Create

```
# cd /
# tar -cf home.tar home/        # archive the whole /home directory (c for create)
# tar -czf home.tgz home/      # same with zip compression
# tar -cjf home.tbz home/       # same with bzip2 compression
```

Only include one (or two) directories from a tree, but keep the relative structure. For example archive /usr/local/etc and /usr/local/www and the first directory in the archive should be local/.

```
# tar -C /usr -czf local.tgz local/etc local/www
# tar -C /usr -xzf local.tgz      # To untar the local dir into /usr
# cd /usr; tar -xzf local.tgz     # Is the same as above
```

Extract

```
# tar -tzf home.tgz              # look inside the archive without extracting (list)
# tar -xf home.tar               # extract the archive here (x for extract)
# tar -xzf home.tgz             # same with zip compression (-xjf for bzip2 compression)
                          # remove leading path gallery2 and extract into gallery
# tar --strip-components 1 -zxvf gallery2.tgz -C gallery/
# tar -xjf home.tbz home/colin/file.txt    # Restore a single file
```

More advanced

```
# tar c dir/ | gzip | ssh user@remote 'dd of=dir.tgz' # arch dir/ and store remotely.
# tar cvf - `find . -print` > backup.tar                # arch the current directory.
# tar -cf - -C /etc . | tar xpf - -C /backup/etc      # Copy directories
# tar -cf - -C /etc . | ssh user@remote tar xpf - -C /backup/etc      # Remote copy.
# tar -czf home.tgz --exclude '*.o' --exclude 'tmp/' home/
```

Find

Some important options:

```
-x (on BSD) -xdev (on Linux)      Stay on the same file system (dev in fstab).
-exec cmd {} \;      Execute the command and replace {} with the full path
-iname      Like -name but is case insensitive
-ls      Display information about the file (like ls -la)
-size n      n is +-n (k M G T P)
-cmin n      File's status was last changed n minutes ago.
# find . -type f ! -perm -444       # Find files not readable by all
# find . -type d ! -perm -111       # Find dirs not accessible by all
# find /home/user/ -cmin 10 -print  # Files created or modified in the last 10 min.
# find . -name '*.[ch]' | xargs grep -E 'expr' # Search 'expr' in this dir and below.
# find / -name "*.core" | xargs rm  # Find core dumps and delete them (also try core.*)
# find / -name "*.core" -print -exec rm {} \;  # Other syntax
# Find images and create an archive, iname is not case sensitive. -r for append
# find . \( -iname "*.png" -o -iname "*.jpg" \) -print -exec tar -rf images.tar {} \;
# find . -type f -name "*.txt" ! -name README.txt -print  # Exclude README.txt files
# find /var/ -size +10M -exec ls -lh {} \;     # Find large files > 10 MB
# find /var/ -size +10M -ls          # This is simpler
# find . -size +10M -size -50M -print
# find /usr/ports/ -name work -type d -print -exec rm -rf {} \;  # Clean the ports
# Find files with SUID; those file are vulnerable and must be kept secure
# find / -type f -user root -perm -4000 -exec ls -l {} \;
```

Miscellaneous

```
# which command                     # Show full path name of command
# time command                      # See how long a command takes to execute
# time cat                           # Use time as stopwatch. Ctrl-c to stop
# set | grep $USER                 # List the current environment
# cal -3                             # Display a three month calendar
# date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
# date 10022155                    # Set date and time
# whatis grep                       # Display a short info on the command or word
# whereis java                      # Search path and standard directories for word
# setenv varname value           # Set env. variable varname to value (csh/tcsh)
# export varname="value"        # set env. variable varname to value (sh/ksh/bash)
# pwd                        # Print working directory
# mkdir -p /path/to/dir               # no error if existing, make parent dirs as needed
# mkdir -p project/{bin,src,obj,doc/{html,man,pdf},debug/some/more/dirs}
# rmdir /path/to/dir                  # Remove directory
# rm -rf /path/to/dir                 # Remove directory and its content (force)
# rm -- -badchar.txt                 # Remove file whitch starts with a dash (-)
# cp -la /dir1 /dir2              # Archive and hard link files instead of copy
# cp -lpR /dir1 /dir2             #
# cp unixtoolbox.xhtml{,.bak}  # Short way to copy the file with a new extension
# mv /dir1 /dir2                  # Rename a directory
# ls -1                            # list one file per line
# history | tail -50             # Display the last 50 used commands
# cd -                             # cd to previous ($OLDPWD) directory
```

Add/Remove software

Debian/Ubuntu/Mint

```
# apt-get update              # First update the package lists
# apt-get install emacs      # Install the package emacs
# dpkg --remove emacs        # Remove the package emacs
# dpkg -S file                   # find what package a file belongs to
```

Tags: *Simple and advanced Shell tutorial for advances and newbie users.*
Created: 5 years ago.
Last edited: 5 years ago.
Reviewed: 5 years ago.
Read 907 times.

Comments

8 months ago	musicinhills	Thank you, invaluable to a lot of us new learners, young and old and older lol
1 year ago	MagicMint	I'd call it a collection of sample Linux commands rather than a "simple and advanced tutorial" — it cannot be both at the same time.
1 year ago	lib2know	it filled some gaps, thank you !!!
1 year ago	dgbutterworth	This is great! Thanks.
2 years ago	blue_bullet	I found the userdel command here. I had created a bogus user name lm17_kdex w/o a home directory and could not figure out how to get MDM to stop displaying the user. sudo userdel lm17_kdex from Konsole took care of it. I had googled all over before I found this list. Copying it to my linux_tips_and_tricks file in nixnotes. Thank you.
2 years ago	cappy	Not much I can say that has not already been said: a very good pocket reference.
2 years ago	jahid_0903014	nice useful tutorial....
2 years ago	Shakeyacres	I'm just poking around as I've never used Terminal. Looks like a great reference.

2 years ago	curtvaughan	The command is "ls -l", not "ls -1", ruinouspalms. This is a very nice pocket reference.
2 years ago	philsoft	Very nice helps newbies thanks
4 years ago	ruinouspalms	The "ls -1" command does not work for me. The directory just lists as it would for a standard "ls" command. Any thoughts on why this would be the case?
4 years ago	roht	Excellent reference!
5 years ago	Labby	Very nice! This will come in handy for sure. Promoted and sbscribed!
5 years ago	quaie	you forgot ! in front of one of the commands in history to re-run it or something like that ...
5 years ago	loukoumas	excellent!
5 years ago	efthialex	Nice job! (y)
5 years ago	Jac978	Perfect! Since now it has been added to my bookmarks.
5 years ago	Frank	Muy bueno, me sirvió de mucho! Completo Saludos!

Other tutorials from Cipi

No other tutorials.

Linux Mint | Blog | Forums