

Linux Handleiding

Paul Wondel

19 mei 2018

Inhoudsopgave

1	i3 Window Manager Controls	3
2	dwm & st	4
3	Linux Basic Commands	5
4	C++ in Linux	7
4.1	Install C/C++ compiler and related tools	7
4.2	Verify installation	7
4.3	How to Compile and Run C/C++ program on Linux	7
4.3.1	How do I compile the program on Linux?	8
4.3.2	How do I run or execute the program I made on Linux?	8
5	Mounting USB devices	9
5.1	Detecting USB Hard Drive	9
5.2	Create a mount point	9
5.3	Mount USB Drive	9
5.4	Access USB Data	10
5.5	Unmount USB	10
6	Git	11
6.1	Commandos	11
7	Package Managers Unix-like	12

1 i3 Window Manager Controls

”mod” = alt or key of choice if set in config

mod+ENTER	Open Terminal
mod+(j/←)	Focus Left
mod+(k/↓)	Focus Down
mod+(l/↑)	Focus Up
mod+(;/→)	Focus Right
mod+SHIFT+(j/←)	Move Window Left
mod+SHIFT+(k/↓)	Move Window Down
mod+SHIFT+(l/↑)	Move Window Up
mod+SHIFT+(;/→)	Move Window Right
mod+SHIFT+Q	Kill A Window
mod+SHIFT+”number”	Move Window To Workshop

Tabel 1: **Moving Around**

mod+e	Default
mod+h	Stacking
mod+w	Tabbed
mod+SHIFT+f	Global Fullscreen
mod+f	Toggle Fullscreen
mod+SHIFT+SPACE	Toggle Floating
mod+”mouse”	Drag Floating

Tabel 2: **Changing Container Modes**

mod+d	Open Application Launcher (dmenu)
mod+”number”	Switch To Workshop num
mod+SHIFT+r	Replace i3 Inplace
mod+SHIFT+e	Exit i3

Tabel 3: **Overige Commandos**

2 dwm & st

dwm Use `sudo make clean install` in `dwm` directory to compile

st Use `sudo make install` in `st` directory to compile

3 Linux Basic Commands

ctrl+l clean terminal output screen

sudo run commands with superuser rights

su log into superuser account(or root)

apt install command to install packages or programs

apt purge delete a package completely

mv to move files or rename them

rm to remove files

rm -r remove directories

rm -rf remove everything by force (**NOT TO BE USED LIGHTLY, VERY DANGEROUS WITH SUDO, NEVER USE / BEHIND IT!!!!**)

mkdir make a new directory

chmod setting permissions to files if they can be readable(**r**), writable(**w**) and/or executable(**e**), the permissions are user(**u**), group(**g**), other(**o**)
chmod <options><permissions><file-name>

ls display the files and subdirectories in the directory

ls -a display every hidden and unhidden files with subdirectories in the directory

pwd displays work directory you're currently in

cat display contents of a file

cat <newfile> create a new file

cat <oldfile> > <newfile> make a new file and copy contents over to newfile

cat <file1> <file2> > <file3> combine contents of multiple files into 1 file

cat <file> | less display a file with a lot of content partly and to navigate within

whoami display username

passwd change password for user(whom is currently in session)

login login as a different user

logout log out of current session as user

cut with the options -b, -c or -f view limited parts of the file

sort sort lines of text files

head output the first part of files

tail output the last part of files

wc print newline, word, and bytecounts for each file (can also count words)

whatis gives you info about a command

alias <new command>= '<actual command it is running>'

shutdown shuts the machine down at a given time (mostly a minute later from when the command was confirmed)

shutdown now shuts down the machine immediately

df -h display memory spaces details

fdisk -l shows all available harddrives and details

cfdisk opens linux disk partition manager

ifconfig -a shows all available networks

mkfs.ext4 build a new linux filesystem with the type of an extended drive

man *Entry out of Linux Manual*

scp sourcefile username@domain FTP file upload to a server domain ex.
'scp test.txt boogie@mydomain.com'

xmodmap displays the mod buttons

4 C++ in Linux

4.1 Install C/C++ compiler and related tools

If you are using Fedora, Red Hat, CentOS, or Scientific Linux, use the following yum command to install GNU c/c++ compiler:

```
|| # yum groupinstall 'Development Tools'
```

If you are using Debian or Ubuntu Linux, type the following apt-get command to install GNU c/c++ compiler:

```
|| $ sudo apt-get update
|| $ sudo apt-get install build-essential manpages-dev
```

4.2 Verify installation

Type the following command to display the version number and location of the compiler on Linux:

```
|| $ whereis gcc
|| $ which gcc
|| $ gcc --version
```

4.3 How to Compile and Run C/C++ program on Linux

Create a file called demo.c using a text editor such as vi, emacs or joe:

```
|| #include<stdio.h>
|| /* demo.c: My first C program on a Linux */
||
|| int main(void)
|| {
||     printf("Hello! This is a test prgoram.\n");
||     return 0;
|| }
```

4.3.1 How do I compile the program on Linux?

Use any one of the following syntax to compile the program called demo.c:

```
|| cc program-source-code.c -o executable-file-name
```

OR

```
|| gcc program-source-code.c -o executable-file-name
```

OR assuming that executable-file-name.c exists

```
|| make executable-file-name
```

4.3.2 How do I run or execute the program I made on Linux?

Simply type the the program name:

```
|| $ ./executable-file-name
```

OR

```
|| $ /path/to/executable-file-name
```


5 Mounting USB devices

5.1 Detecting USB Hard Drive

Use `f-disk -l` to detect connected devices to your computer.

```
|| # fdisk -l
```

OR

```
|| $ sudo fdisk -l
```

It should show a list of connected devices connected. For example:

```
|| Disk /dev/sdc: 7.4 GiB, 7948206080 bytes, 15523840 sectors
|| Units: sectors of 1 * 512 = 512 bytes
|| Sector size (logical/physical): 512 bytes / 512 bytes
|| I/O size (minimum/optimal): 512 bytes / 512 bytes
|| Disklabel type: dos
|| Disk identifier: 0x00000000
||
|| Device      Boot Start      End  Sectors  Size Id Type
|| /dev/sdc1   *      8192 15523839 15515648  7.4G  b W95 FAT32
```

In this case our device is called `/dev/sdc1`. On every system it could have a different name. So look carefully by the details of the devices to identify your usb drive.

5.2 Create a mount point

To access the files on the drive in your own home directory you need to create a directory where you can easily access the usb drive from. This is done by doing the command:

```
|| # mkdir /media/usb-drive
```

You can also make your own name for the directory: "mkdir /media/[directory name]"

5.3 Mount USB Drive

Now you can mount the your USB drive into the mount point. To do this you can type:

```
|| # mount /dev/sdc1 /media/usb-drive/
```

To check if the USB drive has been mounted correctly use this command:

```
|| # mount | grep sdc1
```

You should get a result similar to this:

```
|| /dev/sdc1 on /media/usb-drive type vfat
|| (rw,relatime,mask=0022,dmask=0022,codepage=437,
|| iocharset=utf8,shortname=mixed,errors=remount-ro)
```

5.4 Access USB Data

We now know our mount point that we created and where we mounted the usb device. To access the mount point we use:

```
|| # cd /media/usb-drive
```

You should be in the directory of the usb files. To check you can use the "ls -l" command to see the files.

5.5 Unmount USB

To unmount the usb drive use this command:

```
|| # umount /media/usb-drive
```

Before unmounting you should do need to make sure that no other processes are using the mount point or else you might get an error message. Like for ex.:

```
|| umount: /media/usb-drive: target is busy
```

In some cases useful info about processes that use the device is found by `lsof(8)` or `fuser(1)`.

6 Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

- link: [Git Wiki](#)

6.1 Commandos

git clone git@github.com/yourrepository - clone a git repository from a source

git init - initializes the directory to be a repository

git remote add origin git@github.com/yourrepository - add an git repository to your local repository

git pull - update the repository data

git fetch - overwrite your repository from a git repository

git add - add your files to a commit to be pushed to git (-A, *)

git commit -m "message" - describe your reason for commit

git push (-set-upstream) - push your works to the git repository

git status - shows files which haven't been pushed to the git repository yet

git checkout -b branchname - creates a new local branch

git branch - See the branches within the git repository

git checkout branchname - switch to a branch

git merge branchname/master - merge a branch with another branch or master

7 Package Managers Unix-like

Every Linux system has a package manager. The popular ones are pacman, apt, yum etc. Every package manager has it's own set of commands. Below you can see a list of few package managers on different Linux systems.

- link: [Package Manager Wiki](#)

apk-tools (apk) Alpine Package Keeper, the package manager for Alpine Linux.

APT Advanced Packaging Tool

dpkg Originally used by Debian and now by Ubuntu.

yum Yellow Dog Updater, Modified

dnf Dandified YUM

RPM Package Manager RPM is the Linux Standard Base packaging format and the base of a number of additional tools

slackpkg Slackware Package Management [Permalink](#)

pacman Used in Arch Linux, Frugalware and DeLi Linux.

abs The Arch Build System

emerge Gentoo Linux Package Management

Entropy For Sabayon Linux.

Flatpak A containerized/sandboxed packaging format previously known as xdg-app.

GNU Guix Used by the GNU System

ipkg A dpkg-inspired used on HP's webOS.

netpkg

OpenPKG Cross-platform package management system based on RPM Package Manager

opkg Fork of ipkg

PETget used by Puppy Linux

PISI Used by Pardus

slapt-get which is used by Slackware

Smart Package Manager Used by CCux Linux

Snappy cross-distribution package manager, originally developed for Ubuntu.

swaret

Zero Install (0install) available for Arch Linux, Debian, Knoppix, Mint, Ubuntu, Fedora, Gentoo, OpenSUSE, Red Hat and Slackware.

[Link to a commands of package managers](#)