



ft_linux

how__to__train__your__kernel

Louis Solofrizzo louis@ne02ptzero.me
42 Staff pedago@42.fr

Summary: Make your own linux distribution

Contents

I	Forewords	2
I.1	Linus Torvalds quotes	2
II	Introduction	3
III	Goals	4
IV	General instructions	5
	IV.0.1 The links	5
	IV.0.2 Instructions	5
V	Mandatory part	6
	V.0.1 Packages to Install	6
VI	Bonus part	9
VII	Turn-in and peer-evaluation	10

Chapter I

Forewords

I.1 Linus Torvalds quotes

- No-one has ever called me a cool dude. I'm somewhere between geek and normal.
- I'm perfectly happy complaining, because it's cathartic, and I'm perfectly happy arguing with people on the Internet because arguing is my favourite pastime - not programming.
- Hey, I'm a good software engineer, but I'm not exactly known for my fashion sense. White socks and sandals don't translate to 'good design sense'
- The Linux philosophy is 'Laugh in the face of danger'. Oops. Wrong One. 'Do it yourself'. Yes, that's it.
- Software is like sex: it's better when it's free.
- Microsoft isn't evil, they just make really crappy operating systems.
- I actually think that I'm a rather optimistic and happy person; it's just that I'm not a very positive person, if you see the difference.
- See, you not only have to be a good coder to create a system like Linux, you have to be a sneaky bastard too.
- I do get my pizzas paid for by Linux indirectly.
- To be a nemesis, you have to actively try to destroy something, don't you? Really, I'm not out to destroy Microsoft. That will just be a completely unintentional side effect.

See? Be a good Linus-like guy.

Chapter II

Introduction

Welcome to ft_linux. In this subject, you have to build a basic, but functional, linux distribution.

This subject is not about Kernel programming, but it's highly related.

This distro will be the base for all your kernel projects, because all your kernel-code will be executed here, on your distro.

Try to implement what you want/need to. This is your userspace, take care of it!

Chapter III

Goals

- Build a Linux Kernel
- Install some binaries (See the list below)
- Implement a filesystem hierarchy compliant with the [standards](#)
- Connect to the Internet

Chapter IV

General instructions

IV.0.1 The links

- [The Bible](#)
- [How to build a Kernel](#)
- [Autotools](#)

IV.0.2 Instructions

- For this subject, you must use a virtual machine, live VirtualBox or VMWare.
- Though it is not REQUIRED, you SHOULD read [this](#) and [that](#) right now. Keep those standards in mind. You won't be graded on your compliance with them, but still, it would be good practice.
- You must use a kernel version 4.x. Stable or not, as long as it's a 4.x version.
- The kernel sources must be in /usr/src/kernel-\$(version)
- You must use at least 3 different partitions. (root, /boot and a swap partition). You can of course make more partitions if you want to.
- Your distro must implement a kernel_module loader, like udev.
- The kernel version must contain your student login in it. Something like 'Linux kernel 4.1.2-<student_login>'
- The distribution hostname must be your student login
- You're free to choose between a 32 or 64-bit system.
- You must use a software for central management and configuration, like SysV or SystemD.
- Your distro must boot with a bootloader, like LILO or GRUB.
- The kernel binary located in /boot must be named like this: vmlinuz-<linux_version>-<student_login>. Adapt your bootloader configuration to that.

Chapter V

Mandatory part

V.0.1 Packages to Install



The following versions are known to work together correctly.
However, you are free to use the versions you want.



Some packages below (vim, bash, grub, udev) are examples. Feel free
to change them by any equivalent you like.

- Acl (2.2.52)
- Attr (2.4.47)
- Autoconf (2.69)
- Automake (1.15)
- Bash (4.3.30)
- Bc (1.06.95)
- Binutils (2.25.1)
- Bison (3.0.4)
- Bzip2 (1.0.6)
- Check (0.10.0)
- Coreutils (8.24)
- DejaGNU (1.5.3)
- Diffutils (3.3)
- Eudev (3.1.2)
- E2fsprogs (1.42.13)
- Expat (2.1.0)

- Expect (5.45)
- File (5.24)
- Findutils (4.4.2)
- Flex (2.5.39)
- Gawk (4.1.3)
- GCC (5.2.0)
- GDBM (1.11)
- Gettext (0.19.5.1)
- Glibc (2.22)
- GMP (6.0.0a)
- Gperf (3.0.4)
- Grep (2.21)
- Groff (1.22.3)
- GRUB (2.02 beta2)
- Gzip (1.6)
- Iana-Etc (2.30)
- Inetutils (1.9.4)
- Intltool (0.51.0)
- IPRoute2 (4.2.0)
- Kbd (2.0.3)
- Kmod (21)
- Less (458)
- Libcap (2.24)
- Libpipeline (1.4.1)
- Libtool (2.4.6)
- M4 (1.4.17)
- Make (4.1)
- Man-DB (2.7.2)
- Man-pages (4.02)
- MPC (1.0.3)
- MPFR (3.1.3)
- Ncurses (6.0)
- Patch (2.7.5)
- Perl (5.22.0)

- Pkg-config (0.28)
- Procps (3.3.11)
- Psmisc (22.21)
- Readline (6.3)
- Sed (4.2.2)
- Shadow (4.2.1)
- Sysklogd (1.5.1)
- Sysvinit (2.88dsf)
- Tar (1.28)
- Tcl (8.6.4)
- Texinfo (6.0)
- Time Zone Data (2015f)
- Udev-lfs Tarball (udev-lfs-20140408)
- Util-linux (2.27)
- Vim (7.4)
- XML::Parser (2.44)
- Xz Utils (5.2.1)
- Zlib (1.2.8)

Chapter VI

Bonus part

You have a stable system ? Nice. Now let's have some fun ! Install whatever you want.
Any software, GUI, ANYTHING.
Make this system yours, with your touch.
Special points for an X Server, and window managers / desktop environments, like
GNOME / LXDE / KDE / i3 / dwm ...

Chapter VII

Turn-in and peer-evaluation

Turn your work in using your **GiT** repository, as usual. Only work present on your repository will be graded in defense.

For obvious reasons, you will not push your entire virtual machine but a checksum of your disk image instead.

That can be done with something like:

```
shasum < disk.vdi
```

Keep your disk image somewhere for the peer-evaluation.