Linux From Scratch

从零开始构建 Linux

Xi Ruoyao (xry111)

School of Aerospace Science and Technology
XIDIAN UNIVERSITY

Sept 15, 2018

What is Linux From Scratch?

Linux From Scratch (LFS) is a project that provides you with step-by-step instructions for building your own customized Linux system entirely from source.

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Early History of Linux

- Aug 25, 1991 - Linux kernel released (Linus Torvalds)
- Late 1991 - Boot/Root floppies (H. J. Lu)

Distributions

- Feb, 1992 - MCC Interim (Owen Le Blanc)
- May, 1992 - Softlanding Linux System (Peter MacDonald)
  - Slackware, SuSE, etc.
- Aug 16, 1993 - Debian (Ian Murdock)
  - Ubuntu, Raspbian, etc.
- July 29, 1994 - Red Hat
  - RHEL, Fedora, etc.

Why LFS?

- You want to understanding how the components work together in a GNU/Linux system

Why LFS?

- You want to understanding how the components work together in a GNU/Linux system
- You don't like the distributions
- You want a lightweight system
- You want a non-standard system
- You want to create a new distribution

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Why LFS?

- You want to understanding how the components work together in a GNU/Linux system
- You don't like the distributions
- You want a lightweight system
- You want a non-standard system
- You want to create a new distribution
- You want to follow the upstream
- You want to hack source code

Linux From Scratch
YOUR DISTRO. YOUR RULES.

The Beginning of LFS

Gerard Beekmans started Linux From Scratch in 1999 and released it on Dec 16, 1999 as an LDP HOWTO.

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Early Building Technique

- LFS-HOWTO-1.0
  - Build a statically linked temporary bootable system
  - Build Glibc and GCC in host, then reboot into the i temporary system and install them
  - Install other packages
- LFS-1.3 - rebuild GCC after reboot
- LFS-2.0 - use DESTDIR install for Glibc
- LFS-2.3.6 - using chroot instead of reboot
- LFS-3.0 - build final Glibc after chroot, before rebuilding GCC
- LFS-4.0 - using /static symlink as temporary prefix

Main issue: contamination from the host

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Current Building Technique

- LFS-5.0 - toolchain technique similar to cross compiling
  - Build Pass 1 Binutils and GCC
  - Build temporary Glibc with Pass 1 tools
  - Build temporary system
  - Chroot into the temporary system and build the final system
- LFS-6.5 - cross compiling
  - Pass 1 tools are real cross toolchain now
  - Temporary Glibc, Binutils and GCC are cross compiled

en: cn:

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Triplets

arch-vendor-os-abi

- x86_64-pc-linux-gnu
- x86_64-suse-linux-gnu
- x86_64-linux-gnu
- i686-pc-linux-gnu
- x86_64-unknown-linux-musl
- arm-unknown-linux-gnueabihf
- arm-none-eabi

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Cross Compilation Specification

- Host triplet
- Build triplet
- Target Triplet

Preparation

```
LFS_TGT=$(uname -m)-lfs-linux-gnu
# x86_64-lfs-linux-gnu, mipsel-lfs-linux-gnu, etc.

ln -sv $LFS/tools /
```

Binutils Pass 1

```
../configure --prefix=/tools \
  --target=$LFS_TGT
  --with-sysroot=$LFS
  --with-lib-path=/tools/lib
```

Native triplet is `x86_64-pc-linux-gnu`, but the target is `x86_64-lfs-linux-gnu`. So binutils will be configured as cross tools.

en: cn:

GCC Pass 1

- Adjust dynamic linker path to /tools/lib/ld-*
- Again, specify $LFS_TGT as target triplet
- Disable libstdc++ etc.

en: cn:

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Temporary Glibc

- Specify $LFS_TGT as host triplet, and generate build triplet with GNU config.guess script
- The host triplet is not the build triplet so Glibc will be built with cross Binutils and GCC built in Pass 1
- In cross compiling the generated code can't be contaminated
- All other temporary tools should use this Glibc

en: cn:

Building LFS

- Chroot into $LFS
- Build the final system with temporary tools

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Start Your LFS Building Now

- Step 0. Install a beginner friendly distribution (for e.g. Debian or Deepin)
- Step 1. Build a LFS system strictly following the book
- Step 2. Build necessary packages in BLFS
- Step 3. Build LFS system with your modification ...

en: cn:

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Contributing to LFS

- Propose a change to the book
- Help me to translate the book

lfs-dev: translation:

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Questions

Linux From Scratch

YOUR DISTRO. YOUR RULES.

Questions

- Q: Is LFS a fully free GNU/Linux distribution?
- A: No. But you can make it so :)

Questions

- Q: Is LFS a fully free GNU/Linux distribution?
- A: No. But you can make it so :)
- Q: May I install LFS on non-x86 platform?
- A: Yes. But you need to modify some instructions. And, please send your modification to lfs-dev. But do not try LFS on rasberrypi etc.

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Questions

- Q: Is LFS a fully free GNU/Linux distribution?
- A: No. But you can make it so :)
- Q: May I install LFS on non-x86 platform?
- A: Yes. But you need to modify some instructions. And, please send your modification to lfs-dev. But do not try LFS on rasberrypi etc.
- Q: Can I install *foo* in LFS?
- A: If *foo* is FLOSS, yes. But if it is not in BLFS you need to figure out how to build it. (Try Debian rules, Arch PKGBUILD, and Google.)

Linux From Scratch
YOUR DISTRO. YOUR RULES.

Contact Me

[OT] Welcome to XDU ACM/ICPC!

- The 2018 Xidian University Programming Contest for Freshmen - TBD, maybe Dec 2018
- The 2019 Xidian University Programming Contest - TBD, maybe Apr 2019

[OT] Join Us

We need talented people in the fields:

- Electronic System: Communication, Embedded, Power, System Reliability
- Physics: Plasma, Electromagnetics, Nuclear
- Electrochemistry
- Numerical Simulation

webpage: email: