

基于 deepin 的操作系统构建

第 7 课 构建 Linux 操作系统交叉工具链和临时工具 - 编译交叉工具链

徐小东

统信软件

2022-10-17

课程目录

- ① 交叉编译术语及目标机器表示
- ② 需要的软件包
- ③ 编译交叉工具链
- ④ 补充阅读材料
- ⑤ 参考资料

Section 1

交叉编译术语及目标机器表示

交叉编译术语

► 基本术语

名称	说明
build	指构建程序时使用的机器
host	指将来会运行被构建程序的机器
target	指编译器产生代码的目标机器，仅用于编译器

► 一个场景

- A 机：有编译器 ccA，但较慢
- B 机：较快，但没有编译器
- 目标：为 C 机构建编译器

阶段	Build	Host	Target	操作
1	A	A	B	在 A 机上用 ccA 构建交叉编译器 cc1
2	A	B	C	在 A 机上用 cc1 构建交叉编译器 cc2
3	B	C	C	在 B 机上用 cc2 构建编译器 ccC

► 本课使用

阶段	Build	Host	Target	操作
1	pc	pc	lfs	在 pc 上用 cc-pc 构建交叉编译器 cc1
2	pc	lfs	lfs	在 pc 上用 cc1 构建编译器 cc-lfs
3	lfs	lfs	lfs	在 lfs 上用自身重新构建及测试 cc-lfs

目标机器表示

- ▶ CPU-供应商-内核-操作系统
构建系统使用这种格式来表示目标机器
- ▶ 通过 LFS_TGT 环境变量设置目标机器
即 x86_64-lfs-linux-gnu (参考第 5 课)

Section 2

需要的软件包

需要的软件包

- ▶ 工具链
由 GNU 工程产生的编程工具集合

包名	版本	用途
Binutils	2.38	包含链接器、汇编器及其它工具
GCC	11.2.0	C、C++ 编译器
Linux	5.16.9	Linux API 头文件
Glibc	2.35	C 标准库，包含头文件、库及动态加载程序

Section 3

编译交叉工具链

构建 binutils

► 包含汇编器、链接器以及其它用于处理目标文件的工具

```
tar xf binutils-2.38.tar.xz
cd binutils-2.38

mkdir -v build
cd      build

../configure --prefix=$LFS/tools \ (1)
            --target=$LFS_TGT   \ (2)
            --with-sysroot=$LFS \ (3)
            --disable-nls       \ (4)
            --disable-werror    (5)

make
make install -j1 (6)
```

- (1). 安装到 `$LFS/tools` 目录中
- (2). 以构建交叉链接器
- (3). 交叉编译时在 `$LFS` 中寻找目标系统的库
- (4). 禁用国际化支持，临时工具不需要
- (5). 阻止警告停止构建
- (6). 构建系统中的一个问题可能导致安装过程失败

构建交叉编译器 GCC

- ▶ 包含 GNU 编译器集合，其中有 C 和 C++ 编译器
- ▶ GCC 依赖 GMP、MPFR 和 MPC

包名	版本	用途
gmp	6.2.1	包含提供任意精度算术函数的数学库
mpfr	4.1.0	包含多精度数学函数
mpc	1.2.1	包含一个任意高精度，且舍入正确的复数算术库

```
tar xf gcc-11.2.0.tar.xz
cd gcc-11.2.0

tar xf ../gmp-6.2.1.tar.xz
mv -v gmp-6.2.1 gmp
tar xf ../mpfr-4.1.0.tar.xz
mv -v mpfr-4.1.0 mpfr
tar xf ../mpc-1.2.1.tar.gz
mv -v mpc-1.2.1 mpc

sed -e '/m64=/s/lib64/lib/' \
    -i.orig gcc/config/i386/t-linux64

mkdir -v build
cd      build
```

```
../configure \
  --prefix=$LFS/tools \
  --target=$LFS_TGT \
  --with-sysroot=$LFS \
  --with-glibc-version=2.35 \ (1)
  --with-newlib \ (2)
  --without-headers \ (3)
  --enable-initfini-array \ (4)
  --disable-nls \
  --disable-shared \ (5)
  --disable-multilib \ (6)
  --disable-decimal-float \ (7)
  --disable-threads \
  --disable-libatomic \
  --disable-libgomp \
  --disable-libquadmath \
```

```
--disable-libssp \
--disable-libvtv \
--disable-libstdcxx \ (8)
--enable-languages=c,c++
```

```
make
```

```
make install
```

```
cd ..
```

```
cat gcc/limitx.h gcc/glimits.h gcc/limity.h > \
`dirname $($LFS_TGT-gcc -print-libgcc-file-name)`
/install-tools/include/limits.h (9)
```

- (1). 指定目标所用的 glibc 版本
- (2). 防止编译任何需要 libc 支持的代码
- (3). 防止 GCC 查找标准头文件
- (4). 强制启用一些内部数据结构
- (5). 强制 GCC 静态链接到内部库，此时 glibc 尚未准备好
- (6). 禁用多库支持
- (7)~(8). 禁用浮点扩展、线程、libatomic 等，交叉编译不需要
- (9). 创建一个完整版本的内部头文件

导出 Linux 内核 API 头文件

▶ 导出内核 API 供 glibc 使用

```
tar xf linux-5.16.9.tar.xz
cd linux-5.16.9

# 确保软件包中没有遗留陈旧的文件
make mrproper

make headers
find usr/include -name '.*' -delete
rm usr/include/Makefile
cp -rv usr/include $LFS/usr
```

构建 glibc

► 包含 C 语言标准库

创建一个动态链接器正常工作所必须的符号链接

```
ln -sfv ../lib/ld-linux-x86-64.so.2 $LFS/lib64
ln -sfv ../lib/ld-linux-x86-64.so.2 $LFS/lib64/
    ld-lsb-x86-64.so.3
```

应用补丁

```
patch -Np1 -i ../glibc-2.35-fhs-1.patch
```

```
mkdir -v build
cd      build
```

确保将 *ldconfig* 和 *sln* 工具安装到 */usr/sbin*

```
echo "rootsbindir=/usr/sbin" > configparms
```

```
../configure \
  --prefix=/usr \
  --build=$(../scripts/config.guess) \
  --host=$LFS_TGT \ (1)
  --enable-kernel=3.2 \ (2)
  --with-headers=$LFS/usr/include \ (3)
  libc_cv_slibdir=/usr/lib \ (4)
```

make

make DESTDIR=\$LFS install

改正 *ldd* 脚本中硬编码的可执行文件加载器路径

```
sed '/RTLDDLIST=/s@/usr@@g' -i $LFS/usr/bin/ldd
```

- (1). 使用 `$LFS/tools` 中的交叉编译器和交叉链接器
- (2). 编译出支持 3.2 版或更新的 Linux 内核
- (3). 使用 Linux API 头文件
- (4). 确保库安装到 `/usr/lib` 中

► 确认新工具链的基本功能 (编译和链接) 能工作

```
echo 'int main(){}' > dummy.c
$LFS_TGT-gcc dummy.c
readelf -l a.out | grep '/ld-linux'

rm -v dummy.c a.out
```

► 完成 limits.h 头文件的安装

```
$LFS/tools/libexec/gcc/$LFS_TGT/11.2.0/install-tools/
mkheaders
```

构建 GCC 中的 libstdc++

▶ C++ 标准库

```
cd gcc-11.2.0
rm -r build

mkdir -v build
cd      build

../libstdc++-v3/configure \
  --prefix=/usr           \
  --build=$(../config.guess) \
  --host=$LFS_TGT         \ (1)
  --disable-multilib      \
  --disable-nls           \
  --disable-libstdcxx-pch \ (2)
```

```
--with-gxx-include-dir=/tools/$LFS_TGT/  
include/c++/11.2.0 (3)
```

```
make
```

```
make DESTDIR=$LFS install
```

- (1). 指定使用刚刚构建的交叉编译器
- (2). 阻止安装预编译的 include 文件，此阶段不需要
- (3). 在这个位置搜索标准头文件

Section 4

补充阅读材料

补充阅读材料

- ▶ 《精通 Linux》
 - 第 15.1 节 C 编译器
 - 第 15.2 节 make
 - 第 16 章 从 C 代码编译出软件

Section 5

参考资料

参考资料

- ▶ GNU toolchain
- ▶ GNU Binutils
- ▶ GCC
- ▶ Linux kernel
- ▶ glibc
- ▶ GMP
- ▶ MPFR
- ▶ MPC