正式的完整过程

环境

VMware

8GB内存

8核处理器

60GB硬盘

Ubuntu 16.04.7 desktop_x86

软件	版本
Linux	4.15.0_x86
gcc	8.4.0
clang	3.8
llvm	3.8
python	3.8.2
pip3	23.0.1

准备

apt-get install openssh-server vim -y

可选：

vim /etc/ssh/sshd_config

```
PermitRootLogin yes
```

/etc/init.d/ssh restart

换源

以下操作均为root操作

cp /etc/apt/sources.list /etc/apt/sources.list.bak

vim /etc/apt/sources.list

末尾添加：

```
# deb cdrom:[Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)]/ xenial
main restricted
deb-src http://archive.ubuntu.com/ubuntu xenial main restricted #Added by
software-properties
```

```
deb http://mirrors.aliyun.com/ubuntu/ xenial main restricted
deb-src http://mirrors.aliyun.com/ubuntu/ xenial main restricted multiverse
universe #Added by software-properties
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted
multiverse universe #Added by software-properties
deb http://mirrors.aliyun.com/ubuntu/ xenial universe
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe
deb http://mirrors.aliyun.com/ubuntu/ xenial multiverse
deb http://mirrors.aliyun.com/ubuntu/ xenial-updates multiverse
deb http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted universe
multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted
universe multiverse #Added by software-properties
deb http://archive.canonical.com/ubuntu xenial partner
deb-src http://archive.canonical.com/ubuntu xenial partner
deb http://mirrors.aliyun.com/ubuntu/ xenial-security main restricted
deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security main restricted
multiverse universe #Added by software-properties
deb http://mirrors.aliyun.com/ubuntu/ xenial-security universe
deb http://mirrors.aliyun.com/ubuntu/ xenial-security multiverse
deb http://mirrors.aliyun.com/ubuntu focal main universe
```

apt-get update

装包

后续操作需要(主要是git)：

apt-get install git tar mlocate unzip -y

编译相关：

apt-get install make flex bison libssl-dev libc6-dev libelf-dev build-essential pkg-config libncurses5-dev openssl libncurses-dev -y

GCC:

apt-get gcc-8 -y

update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 100

如果有其他版本的同名软件，执行：

update-alternatives --config gcc

llvm-clang： https://releases.llvm.org/3.8.0/clang+llvm-3.8.0-x86_64-linux-gnu-ubuntu-16.04.tar.xz

tar -xvf clang+llvm-3.8.0-x86_64-linux-gnu-ubuntu-16.04.tar.xz mv clang+llvm-3.8.0-x86_64-linux-gnu-ubuntu-16.04 /usr/local/llvm vim /etc/profile

```
export LLVM_HOME=/usr/local/llvm/bin
export PATH=$LLVM_HOME:$PATH
```

source /etc/profile llvm-as -version clang --version

克隆

内核

git clone --branch ExtFUSE-1.0 https://github.com/extfuse/linux

或者在官网下载zip包：linux-ExtFUSE-1.0.zip

项目

git clone https://github.com/extfuse/extfuse

依赖库

git clone --branch ExtFUSE-1.0 https://github.com/extfuse/libfuse

测试用例

git clone https://github.com/ashishbijlani/StackFS

编译内核

unzip linux-ExtFUSE-1.0.zip

mv linux-ExtFUSE-1.0 /usr/src/linux

cd /usr/src/linux

```
make menuconfig
--> General setup
    [*] Enable bpf() system call
--> File systems
    <*> FUSE (Filesystem in Userspace) support
    [*]   Extension framework for FUSE
```

(注意是*号编译进内核，M代表以模块形式安装)

make -j$(nproc) all

mkdir /lib/modules/5.2.0

cp ./modules.* /lib/modules/5.2.0

make modules_install

make install

make headers_install

update-grub

编译完成以后重启，进入新内核5.2.0

编译extFuse

cd /root/extfuse

vim /etc/profile 末尾添加

```
export EXTFUSE_REPO_PATH=/root/extfuse
```

source /etc/profile

LLC=llc CLANG=clang make

编译libfuse

apt-get install libtool automake -y

./makeconf.sh

```
Running libtoolize...
Running autoreconf...
libtoolize: putting auxiliary files in '.'.
libtoolize: copying file './ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'm4'.
libtoolize: copying file 'm4/libtool.m4'
libtoolize: copying file 'm4/ltoptions.m4'
libtoolize: copying file 'm4/ltsugar.m4'
libtoolize: copying file 'm4/ltversion.m4'
libtoolize: copying file 'm4/lt~obsolete.m4'
configure.ac:9: installing './compile'
configure.ac:4: installing './config.guess'
configure.ac:4: installing './config.sub'
configure.ac:5: installing './install-sh'
configure.ac:5: installing './missing'
Makefile.am: installing './INSTALL'
example/Makefile.am: installing './depcomp'
To compile run './configure', and then 'make'.
```

./configure

```
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
```

```
checking target system type... x86_64-pc-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
checking how to print strings... printf
checking whether make supports the include directive... yes (GNU style)
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for a sed that does not truncate output... /bin/sed
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for fgrep... /bin/grep -F
checking for ld used by gcc... /usr/bin/ld
checking if the linker (/usr/bin/ld) is GNU ld... yes
checking for BSD- or MS-compatible name lister (nm)... /usr/bin/nm -B
checking the name lister (/usr/bin/nm -B) interface... BSD nm
checking whether ln -s works... yes
checking the maximum length of command line arguments... 1572864
checking how to convert x86_64-pc-linux-gnu file names to x86_64-pc-linux-gnu
format... func_convert_file_noop
checking how to convert x86_64-pc-linux-gnu file names to toolchain format...
func_convert_file_noop
checking for /usr/bin/ld option to reload object files... -r
checking for objdump... objdump
checking how to recognize dependent libraries... pass_all
checking for dlltool... no
checking how to associate runtime and link libraries... printf %s\n
checking for ar... ar
checking for archiver @FILE support... @
checking for strip... strip
checking for ranlib... ranlib
checking command to parse /usr/bin/nm -B output from gcc object... ok
checking for sysroot... no
checking for a working dd... /bin/dd
checking how to truncate binary pipes... /bin/dd bs=4096 count=1
checking for mt... mt
checking if mt is a manifest tool... no
checking how to run the C preprocessor... gcc -E
checking for ANSI C header files... yes
checking for sys/types.h... yes
```

```
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking for dlfcn.h... yes
checking for objdir... .libs
checking if gcc supports -fno-rtti -fno-exceptions... no
checking for gcc option to produce PIC... -fPIC -DPIC
checking if gcc PIC flag -fPIC -DPIC works... yes
checking if gcc static flag -static works... yes
checking if gcc supports -c -o file.o... yes
checking if gcc supports -c -o file.o... (cached) yes
checking whether the gcc linker (/usr/bin/ld -m elf_x86_64) supports shared
libraries... yes
checking whether -lc should be explicitly linked in... no
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
checking whether stripping libraries is possible... yes
checking if libtool supports shared libraries... yes
checking whether to build shared libraries... yes
checking whether to build static libraries... yes
checking for gcc option to accept ISO C99... none needed
checking for gcc option to accept ISO Standard C... (cached) none needed
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking for fork... yes
checking for setxattr... yes
checking for fdatasync... yes
checking for splice... yes
checking for vmsplice... yes
checking for utimensat... yes
checking for pipe2... yes
checking for posix_fallocate... yes
checking for fstatat... yes
checking for openat... yes
checking for readlinkat... yes
checking for struct stat.st_atim... yes
checking for struct stat.st_atimespec... no
checking for library containing dlopen... -ldl
checking for library containing clock_gettime... none required
checking for ulockmgr_op in -lulockmgr... no
checking for ld used by gcc... /usr/bin/ld -m elf_x86_64
checking if the linker (/usr/bin/ld -m elf_x86_64) is GNU ld... yes
checking for shared library run path origin... done
checking for iconv... yes
checking for working iconv... yes
checking for iconv declaration...
         extern size_t iconv (iconv_t cd, char * *inbuf, size_t *inbytesleft, char
* *outbuf, size_t *outbytesleft);
configure: MOUNT_FUSE_PATH env var not set, using default /sbin
```

```
configure: UDEV_RULES_PATH env var not set, using default NONE/lib/udev/rules.d
configure: INIT_D_PATH env var not set, using default /etc/init.d
checking if umount supports --fake --no-canonicalize...
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating fuse3.pc
config.status: creating Makefile
config.status: creating lib/Makefile
config.status: creating util/Makefile
config.status: creating example/Makefile
config.status: creating include/Makefile
config.status: creating doc/Makefile
config.status: creating include/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
configure: WARNING:
****************************************************************
* Please install util-linux version 2.18 or later which supports *
* --fake and --no-canonicalize options in mount and umount       *
****************************************************************
```

make -j8

make install

python3.8.2

wget https://www.python.org/ftp/python/3.8.2/Python-3.8.2.tgz tar -xzvf Python-3.8.2.tgz cd Python-3.8.2 ./configure make all make install

update-alternatives --install /usr/bin/python3 python3 /usr/local/bin/python3.8 1 update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.5 2

python3 -V

```
Python 3.8.2
```

pip3

wget https://bootstrap.pypa.io/get-pip.py

python3 get-pip.py

pip3 install --upgrade pip

```
Collecting pip
  Downloading
https://files.pythonhosted.org/packages/07/51/2c0959c5adf988c44d9e1e0d940f5b074516
ecc87e96b1af25f59de9ba38/pip-23.0.1-py3-none-any.whl (2.1MB)
```

```
     |████████████████████████████████| 2.1MB 70kB/s
Installing collected packages: pip
  Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
```

(有warning可以多试几次)

pip3 -V

```
pip 23.0.1 from /usr/local/lib/python3.8/site-packages/pip (python 3.8)
```

libfuse test

pip3 install pytest

python3 -m pytest test/

```
============================= test session starts ==============================
platform linux -- Python 3.8.2, pytest-7.2.2, pluggy-1.0.0
rootdir: /root/libfuse
collected 0 items

============================== no tests ran in 0.00s ===============================
```

StackFS

git clone https://github.com/ashishbijlani/StackFS

vim /etc/profile 末尾添加

```
export ROOT_DIR=/root
export MNT_DIR=/mnt
```

source /etc/profile

cd /StackFS/

make -j8

cp $EXTFUSE_REPO_PATH/bpf/extfuse.o /tmp

挂载

语句

```
sh -c "LD_LIBRARY_PATH=$EXTFUSE_REPO_PATH ./StackFS_ll -o max_write=131072 -o
writeback_cache -o splice_read -o splice_write -o splice_move -r $ROOT_DIR
$MNT_DIR -o allow_other"
```

报错

```
./StackFS_ll: error while loading shared libraries: libfuse3.so.0: cannot open
shared object file: No such file or directory
```

解决

```
ldconfig
用处：搜寻目录 /lib 和 /usr/lib 以及动态库配置文件 /etc/ld.so.conf 内所列的目录下，搜索
出可共享的动态链接库
```

然后重新挂载就行

效果

挂载窗口会持续显示操作信息：

```
root@ubuntu:~/StackFS# sh -c "LD_LIBRARY_PATH=$EXTFUSE_REPO_PATH ./StackFS_ll -o
max_write=131072 -o writeback_cache -o splice_read -o splice_write -o splice_move
-r $ROOT_DIR $MNT_DIR -o allow_other"
Multi Threaded : 1
loading extfuse prog event 1
loading extfuse prog event 3
loading extfuse prog event 15
loading extfuse prog event 16
loading extfuse prog event 4
loading extfuse prog event 22
loading extfuse prog event 25
loading extfuse prog event 12
loading extfuse prog event 11
loading extfuse prog event 10
    ExtFUSE eBPF bytecode loaded: ctxt=0x7fdfa4000d80 fd=11
```

另开窗口查看挂载信息，最后一条就是StackFS：

```
root@ubuntu:~/StackFS# mount | grep fuse
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
vmware-vmblock on /run/vmblock-fuse type fuse.vmware-vmblock
```

```
(rw,relatime,user_id=0,group_id=0,default_permissions,allow_other)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse
(rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/root/StackFS/StackFS_ll on /mnt type fuse.StackFS_ll
(rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
```

问题

虽然复现了，但是挂载目录无法进入，cd之后就会卡住