Project: Compile exFAT-FUSE in RHEL 8

I have a large external SSD disk attached to my new desktop, and because I multiboot between Windows and different versions of Linux, I decided to keep it factory formatted with exFAT, which is an extended FAT32 implementation easily accessible from all of the operating systems for both read and write operations.

In today’s Unix Tutorial project I compile exFAT-FUSE software to access the exFAT partition from that external disk my RHEL 8 PC.

What is exFAT
FAT32 and ExFAT originate from Windows. It’s a family of filesystems that are native to Windows but go back to MS-DOS roots so the format is generally well-known, well-documented and widely implemented in modern Unix and Linux systems.

What is exFAT-FUSE

Usually filesystems require kernel modules to be providing desired functionality. Working with filesystems is considered to be low-level enough functionality that they’re done in kernel space (vs regular programs that run in user space).

But when this functionality is not easily available, it’s possible to get some filesystem drivers working in the userspace, by using FUSE technology.

**FUSE means Filesystem in USER space.** It requires no kernel module for it to work, provided that functionality is slightly limited.

**exFAT-FUSE** is an example of FUSE implementation of filesystem driver – you compile binaries and run commands without any updates to Linux kernel or modules.

Installing exFAT Packages from EPEL

Usual approach is to use the **EPEL repository** and install exFAT utils from it...
Unfortunately, exfat packages are not available in RHEL 8 version of EPEL repository yet:

```
root@redhat:~ # yum --disablerepo="*" --enablerepo="epel" list available | grep exfat
root@redhat:~ #
```

This leaves us with the option of compiling exfat-fuse package ourselves.

**IMPORTANT:** you need to have development tools (automake, autoconf, make, gcc and a few other bits and pieces) installed on your RHEL 8 system before going through the rest of procedure.

You can install these tools using **dnf:**

```
root@redhat:~ # dnf group install "Development Tools"
```

**Compile exFAT-FUSE in RHEL 8**

Let’s download the exfat-fuse source code from GitHub:

```
root@redhat:~ # cd /dist
root@redhat:/dist  # git clone https://github.com/relan/exfat.git
Cloning into 'exfat'...
remote: Enumerating objects: 3394, done.
remote: Total 3394 (delta 0), reused 0 (delta 0), pack-reused
Receiving objects: 100% (3394/3394), 657.61 KiB | 1.58 MiB/s, done.
Resolving deltas: 100% (2184/2184), done.

Now prepare the configuration files for compiling:

root@redhat:/dist # cd exfat
root@redhat:/dist/exfat # autoreconf --install
  configure.ac:32: installing './ar-lib'
  configure.ac:29: installing './compile'
  configure.ac:34: installing './config.guess'
  configure.ac:34: installing './config.sub'
  configure.ac:28: installing './install-sh'
  configure.ac:28: installing './missing'
  dump/Makefile.am: installing './depcomp'

... and attempt running the `configure` script:

root@redhat:/dist/exfat # ./configure --prefix=/soft
  checking for a BSD-compatible install... /bin/install -c
  checking whether build environment is sane... yes
  checking for a thread-safe mkdir -p... /bin/mkdir -p
  checking for gawk... gawk
  checking whether make sets $(MAKE)... yes
  checking whether make supports nested variables... yes
  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 whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking for gcc option to accept ISO C99... none needed
checking for ranlib... ranlib
checking for ar... ar
checking the archiver (ar) interface... ar
checking for special C compiler options needed for large files... no
checking for _FILE_OFFSET_BITS value needed for large files... no
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for pkg-config... /bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking for UBLIO... no
checking for FUSE... no
configure: error: Package requirements (fuse) were not met:
Package 'fuse', required by 'virtual:world', not found
Consider adjusting the PKG_CONFIG_PATH environment variable if you installed software in a non-standard prefix.

Alternatively, you may set the environment variables FUSE_CFLAGS and FUSE_LIBS to avoid the need to call pkg-config.

See the pkg-config man page for more details.

Ok, that didn’t work. Need the FUSE development library installed:

root@redhat:/dist/exfat # yum install fuse-devel
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:03 ago on Tue 15 Oct 2019 08:48:59 IST.

Dependencies resolved.

Package       Arch      Version         Repository
Installing:
  fuse-devel        x86_64  2.9.7-12.el8  rhel-8-for-x86_64-baseos-rpms  43 k

This time `configure` works:

```
root@redhat:/dist/exfat # ./configure --prefix=/soft
checking for a BSD-compatible install... /bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
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... C
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 FOR GCC ACCEPTS -g... yes
checking for gcc option to accept ISO C89... none needed
cHECKING FOR GCC UNDERSTANDS -c AND -O TOGETHER... yes
checking whether make supports the include directive... yes (GNU style)
cHECKING DEPENDENCY STYLE OF GCC... GCC3
cHECKING FOR GCC OPTION TO ACCEPT ISO C99... none needed
cHECKING FOR RANLIB... RANLIB
cHECKING FOR AR... AR
cHECKING THE ARCHIVER (AR) INTERFACE... AR
cHECKING FOR SPECIAL C COMPILER OPTIONS NEEDED FOR LARGE FILES... no
cHECKING FOR __FILE_OFFSET_BITS VALUE NEEDED FOR LARGE FILES... no
cHECKING BUILD SYSTEM TYPE... x86_64-pc-linux-gnu
cHECKING HOST SYSTEM TYPE... x86_64-pc-linux-gnu
cHECKING FOR PKG-CONFIG... /bin/pkg-config
```
checking pkg-config is at least version 0.9.0... yes
checking for UBLIO... no
checking for FUSE... yes
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating libexfat/Makefile
config.status: creating dump/Makefile
config.status: creating fsck/Makefile
config.status: creating fuse/Makefile
config.status: creating label/Makefile
config.status: creating mkfs/Makefile
config.status: creating Makefile
config.status: creating libexfat/config.h
config.status: executing depfiles commands

Let’s make the software. Make command compiles source codes into binary objects and eventually gives us executable files with commands that we can execute. End result of making exFAT-FUSE will be a number of exFAT specific commands for creating and mounting exFAT filesystems.

root@redhat:/dist/exfat # make
Making all in libexfat
make[1]: Entering directory '/dist/exfat/libexfat'
(CPATH="${ZSH_VERSION+.:}:": & & cd .. & & /bin/sh /dist/exfat/missing autoheader)
rm -f stamp-h1
touch config.h.in
cd .. & & /bin/sh ./config.status libexfat/config.h
config.status: creating libexfat/config.h
config.status: libexfat/config.h is unchanged
make all-am
make[2]: Entering directory '/dist/exfat/libexfat'
...
mv -f .deps/mkexfatfs-vbr.Tpo .deps/mkexfatfs-vbr.Po
gcc -g -O2 -o mkexfatfs mkexfatfs-cbm.o mkexfatfs-fat.o
mkexfatfs-main.o mkexfatfs-mkexfat.o mkexfatfs-rootdir.o
mkexfatfs-uct.o mkexfatfs-uctc.o mkexfatfs-vbr.o
That’s done. Now we need to install the software, for this we run `make install`:

```
root@redhat:/dist/exfat # make install
Making install in libexfat
make[1]: Entering directory '/dist/exfat/libexfat'
make[2]: Entering directory '/dist/exfat/libexfat'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/dist/exfat/libexfat'
make[1]: Leaving directory '/dist/exfat/libexfat'
Making install in dump
make[2]: Entering directory '/dist/exfat/dump'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/dist/exfat/dump'
make[1]: Leaving directory '/dist/exfat/dump'
Making install in fsck
make[2]: Entering directory '/dist/exfat/fsck'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/dist/exfat/fsck'
```

```
make[3]: Entering directory '/dist/exfat/mkfs'
ln -sf mkexfatfs /soft/sbin/mkfs.exfat
make[3]: Leaving directory '/dist/exfat/mkfs'
make[2]: Entering directory '/dist/exfat/mkfs'
```

```
ln -sf mkexfatfs /soft/sbin/mkfs.exfat
make[2]: Leaving directory '/dist/exfat/mkfs'
```
make[1]: Leaving directory '/dist/exfat/mkfs'
make[1]: Entering directory '/dist/exfat'
make[2]: Entering directory '/dist/exfat'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/dist/exfat'
make[1]: Leaving directory '/dist/exfat'

We now can go into /soft/sbin directory and find new binaries we just installed:

```
root@redhat:/dist/exfat # cd /soft/sbin
root@redhat:/soft/sbin # ls -al
total 1160
drwxr-xr-x. 2 root root 154 Oct 15 08:50 .
drwxr-xr-x. 5 root root 45 Oct 15 08:50 ..
-rwxr-xr-x. 1 root root 228632 Oct 15 08:50 dumpexfat
-rwxr-xr-x. 1 root root 226872 Oct 15 08:50 exfatfsck
-rwxr-xr-x. 1 root root 217272 Oct 15 08:50 exfatlabel
lrwxrwxrwx. 1 root root 9 Oct 15 08:50 fsck.exfat -> exfatfsck
-rwxr-xr-x. 1 root root 249264 Oct 15 08:50 mkexfatfs
lrwxrwxrwx. 1 root root 9 Oct 15 08:50 mkfs.exfat -> mkexfatfs
lrwxrwxrwx. 1 root root 16 Oct 15 08:50 mount.exfat -> mount.exfat-fuse
-rwxr-xr-x. 1 root root 256544 Oct 15 08:50 mount.exfat-fuse
root@redhat:/soft/sbin #
```

root@redhat:/dist/exfat # cd /soft/sbin
root@redhat:/soft # ls
deluge sbin share
root@redhat:/soft # cd s
-bash: cd: s: No such file or directory
root@redhat:/soft # cd sbin/
root@redhat:/soft/sbin # ls
dumpexfat exfatlabel mkexfatfs mount.exfat
exfatfsck fsck.exfat mkfs.exfat mount.exfat-fuse
root@redhat:/soft/sbin # ./
mkexfatfs mkfs.exfat mount.exfat
mount.exfat-fuse

Transaction Summary

Install 1 Package
Total download size: 43 k
Installed size: 124 k
Is this ok [y/N]: y
Downloading Packages:

```
  fuse-devel-2.9.7-12.el8.x86_64.rpm     17 kB/s |  43 kB     00:02
  Total                               17 kB/s |  43 kB     00:02
```

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing          :
  1/1
  Installing        : fuse-devel-2.9.7-12.el8.x86_64
  1/1
  Running scriptlet: fuse-devel-2.9.7-12.el8.x86_64
  1/1
  Verifying         : fuse-devel-2.9.7-12.el8.x86_64
  1/1
Installed products updated.

Installed:
  fuse-devel-2.9.7-12.el8.x86_64
Complete!

Mount exFAT Filesystem

The moment of truth!

While in the same /soft/sbin directory, let’s run the mount.exfat-fuse command and attempt to mount the /dev/sdc1 filesystem again:
root@redhat:/soft/sbin # mkdir /exfat
root@redhat:/soft/sbin # ./mount.exfat-fuse /dev/sdc1 /exfat FUSE exfat 1.3.0

Success!

root@redhat:/soft/sbin # df -h /exfat
Filesystem Size Used Avail Use% Mounted on
/dev/sdc1 932G 556G 377G 60% /exfat

That’s it! It’s been fun

See Also

- Unix Tutorial Projects
- Project: Connect LG 5K Dispaly to PC via USB-C Thunderbolt 3
- RHEL 8
- Red Hat Linux
- List installed packages with yum
- Unix commands
- mount command
- fsck command