A tale of ELFs and DWARFs

A glimpse into the world of linkers, loaders and binary formats

Volker Krause
vkrause@kde.org
@VolkerKrause

Our Workflow

- Write code
- Run compiler
- …
- Run application
- Profit!

Why care?

- Understanding linker errors
- Debugging weird runtime behavior
- Performance
- Surviving without CMake convenience

The Magic

- Build time
  - combine multiple object files into one library or executable
- Run time
  - find all needed libraries
  - map them into memory
  - resolve dynamic symbols
  - call the entry point functions

Linker Errors

- `undefined reference to `Class::method()'`
- `-Wl,--fatal-warnings -Wl,--no-undefined`
  - Catch missing symbols at compile time

- Causes
  - Method is declared but not defined
  - Signature mismatch
  - Using a method that is not exported

Looking Into Libraries

- Binutils: **`nm`**, **`readelf`**, **`objdump`**
- Mac: **`otool`**
- Windows: Dependency Walker

```
0000000000003a344 T _ZNK4KJob11errorStringEv
0000000000039fc8 T _ZNK4KJob11isSuspendedEv
```

Name Mangling

- For C: just using the method name
  - `extern “C” {…}`
- C++: overloading, templates, namespaces,…
- Information included:
  - argument types, cv-qualifiers
  - library name (Mach-O two-level lookup)
  - public/protected/private (MSVC)
- Demangle: `c++filt`, `nm -C`

Export Macros

- `-fvisibility=hidden`
- UNIX
  - `__attribute__((visibility("default")))`
- Windows
  - `__declspec(dllexport)`
  - `__declspec(dllimport)`
- Upper-case letters in **`nm`** output indicate exported symbols

File Formats

- ELF (Linux), Mach-O (Mac), PE (Windows)
- Simple, designed for zero-copy mapping into memory
- Essentially the same for libraries and executables
- Consist of multiple sections, custom sections possible
  - Symbol table
  - Text (executable code)
  - Data (string literals, QRC, ...)

Qt5 Plug-ins

- Implementation hidden by **`moc`**
- Plug-in meta-data in custom section
  - Contains a QJsonObject in binary format: **`qt_pluginMetaData`**
  - `__attribute__((section(".qtmetadata")))`
  - `readelf --sections plugin.so`
- Plug-in entry symbol (must be exported!)
  - `nm -C plugin.so | grep qt_plugin_instance`

Debug Information

- DWARF
  - Custom section or separate file
  - -g[0-3] decides level of detail
- Index for faster loading in GDB:

```
(gdb) save gdb-index <lib>
objcopy \
  --add-section .gdb_index=<lib>.gdb-index \
  --set-section-flags .gdb_index=readonly \
  <lib> <lib>
```

- With `gold`: `--gdb-index`

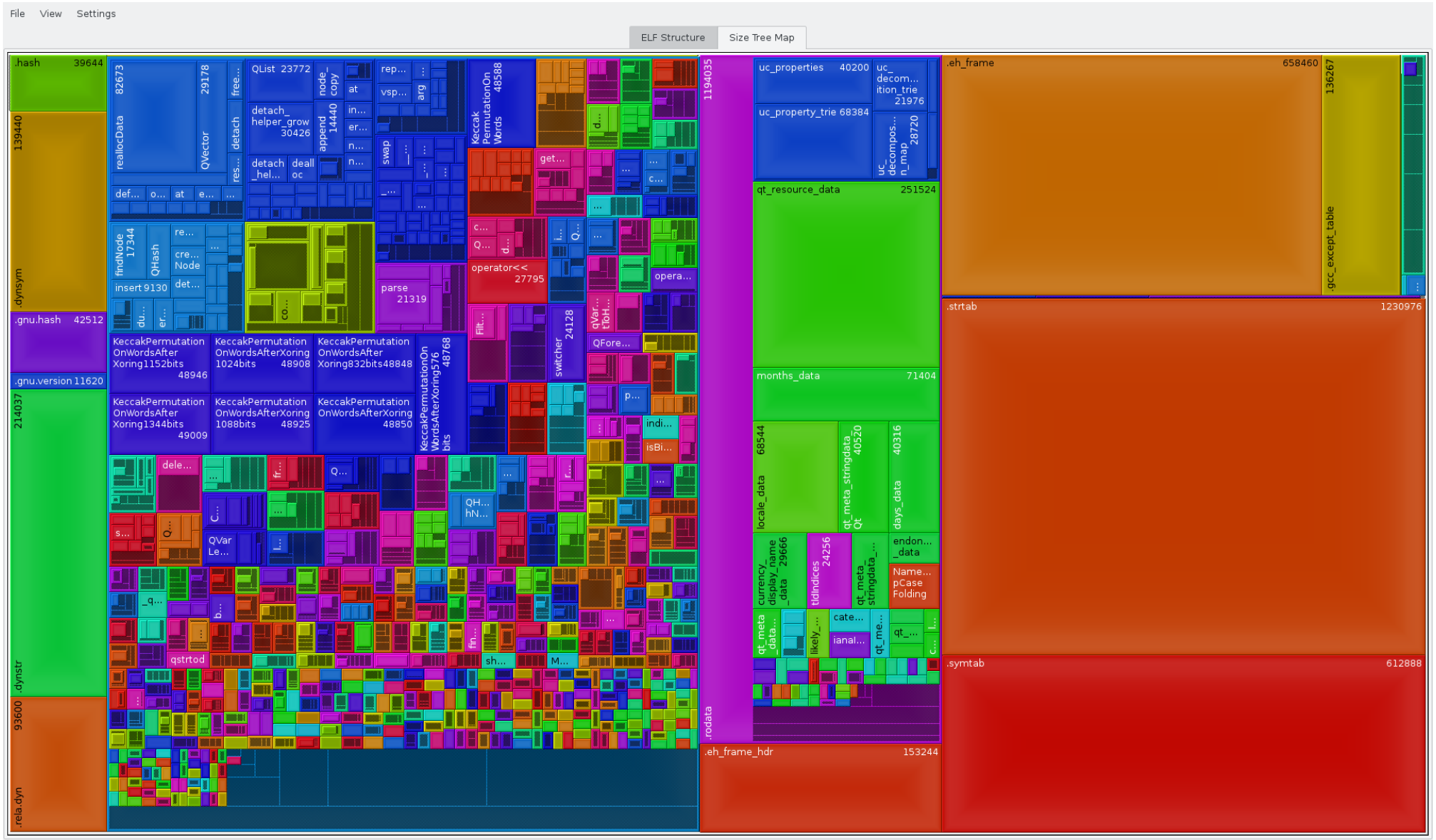
Finding Libraries

- Search paths:
  - RPATH if RUNPATH not set (`-rpath`, `chrpath`)
  - LD_LIBRARY_PATH
  - RUNPATH (`--enable-new-dtags`)
  - /etc/ld.so*
- Diagnostics:
  - Static: `ldd`, `readelf -d <elf>`
  - Dynamic: `LD_DEBUG=libs`, `strace`

Does size matter?

- mmap'ed in 3 sections:
  - Read-only
  - Read-only + executable
  - Read-write
- Read-only sections are shared
- Writable sections use copy-on-write

Code Size


File
View
Settings
ELF Structure
Size Tree Map
.hash 39644
.dynsym 139440
.gnu.hash 42512
.gnu.version 11620
.dynstr 214037
.rela.dyn 93600
reallocData 82673
QVector 29178
QList 23772
detach_helper_grow 30426
append 14440
node_copy
detach_hel...
deallo c
findNode 17344
QHash
insert 9130
parse 21319
KeccakPermutationOnWordsAfterXoring1152bits 48946
KeccakPermutationOnWordsAfterXoring1024bits 48908
KeccakPermutationOnWordsAfterXoring832bits48848
KeccakPermutationOnWordsAfterXoring1344bits 49009
KeccakPermutationOnWordsAfterXoring1088bits 48925
KeccakPermutationOnWordsAfterXoring 48850
KeccakPermutationOnWordsAfterXoring576bits 48768
KeccakPermutationOnWords 48588
operator<< 27795
switcher 24128
QFore...
qstrtod
1194035
.rodata
uc_properties 40200
uc_property_trie 68384
uc_decomposition_trie 21976
uc_decomposition_map 28720
qt_resource_data 251524
months_data 71404
locale_data 68544
qt_meta_stringdata_Qt 40520
days_data 40316
currency_display_name_data 29666
tzIndices 24256
qt_meta_stringdata_...
endon..._data
Name...pCase Folding
likely_...
cate...
ianal...
qt_me...
qt_...
.eh_frame_hdr 153244
.eh_frame 658460
.gcc_except_table 136267
.strtab 1230976
.symtab 612888

Writable Data

- Keep small, this cannot be shared

```
const char *c1 = "...";
const char c2[] = "...";
static const char *c3 = "...";
char* c4;
```

- Verify with **nm**: “r” vs. “d” or “b”

```
0000000000601040 D c1
0000000000400633 r c2
0000000000601048 d c3
0000000000601060 B c4
```

Read-only Data

```
const char *data = “hello world”;
int main(int, char**) {
  const_cast<char*>(data)[0] = 'H';
}
```

- Valgrind:

```
Process terminating with default action of
 signal 11 (SIGSEGV)
Bad permissions for mapped region at address 0x400684
  at 0x4005F0: main (main.cpp:3)
```

Relocation

- Procedure Linkage Table (PLT)
  - Stubs for calling external functions
- Global Offset Table (GOT)
  - Filled during dynamic linking
  - Contains addresses of external symbols
- Calling another function:
  - Intra-library: relative jump (`-Bsymbolic`)
  - Inter-library: PLT stub + GOT

Qt5 PMF Connections

- Function pointer **`connect()`**
  - Compares pointer to signal with **`moc`** data
  - Must avoid comparison between PLT stub and actual function…
- Addressed by **`-fPIC`**/**`-fPIE`**
  - Safety check in **`qglobal.h`** enforces that
  - On ARM: possibly also **`-pie`** due to toolchain bug

Conclusion

- CMake hides most of this from us
- Nasty to debug if something goes wrong
- To learn more, read Ulrich Drepper's “How to write shared libraries”

Questions?

References

- John R. Levine: “Linkers and Loaders”, Morgan-Kauffman, ISBN 1-55860-496-0. 2000
  http://www.iecc.com/linker/
- Ulrich Drepper: “How To Write Shared Libraries”. 2011
  http://www.akkadia.org/drepper/dsohowto.pdf

- Slides & code:
  http://www.kdab.com/~volker/akademy/2014/
  kde:scratch/vkrause/elf-dissector.git