Source Code

Are we not forgetting something?

Guus Sliepen

FOSDEM 2017

February 4, 2017

Outline

- About
  - What is source code?
  - Why source code?
- File types
- Problems
- Conclusion
- Appendix
  - Executables
  - Documentation
  - Fonts
  - Images
  - Sound and Music

What is source code?

A very nice definition from the GPL version 3:

> *1. Source Code.*
>
> *The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.*

And later:

> *The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities.*

Why do we need source code?

- Because the license of the work requires it
  (GPL variants, Mozilla Public License, IBM Public License).
- Because your distribution requires it
  (Debian, Fedora, ???)
- Because we want to be able to fix bugs, improve, and learn!

Debian Free Software Guidelines §2

> *"The program must include source code, and must allow distribution in source code as well as compiled form."*

- One of the cornerstones of Debian
- Applies to all kinds of files, not just executables
- Tools to edit source code must be in main too

How do we benefit from source code?

- We can study it, and learn from it.
- We can fix bugs, and add new features.
- We can port it to different platforms.
- We can do the above with others.

This applies not only to executables, but to any kind of file!

File types

- Executables
- Manuals
- Markup (CSS, Glade, ...)
- Translations
- Fonts
- Images
- Sound
- Music
- Movies
- Databases (OSM, ephemerides, ...)
- ...

Possible problems with source code

- Not clear what preferred form is.
- It's too big (think Blender movie).
- It takes too much CPU power to compile.
- The compiler is non-free.
- The author lost it.
- The author doesn't want to give it.
- The author used the wrong license.
- It was once autogenerated.
- There is a dependency hell.

What should *you* do?

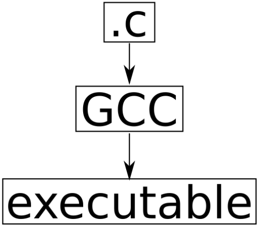
You should:

- Ensure all source code is available.
- Take inventory of what is missing (file bugs).
- Work with upstream to get the missing sources.
- Push back if that doesn't help (remove package?)

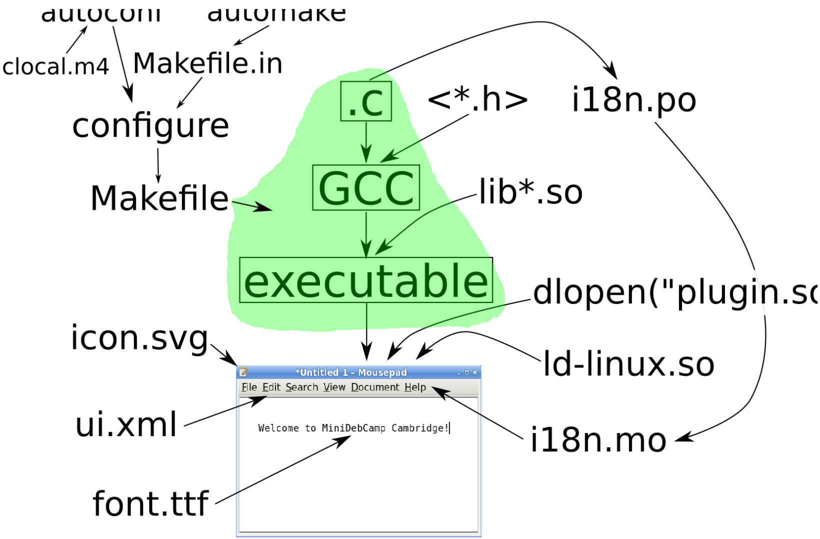
But remember:

- Be reasonable, use common sense.
- Avoid making problems for end-users.

Executables



Executables



Welcome to MiniDebCamp Cambridge!

Source of executables

- Binaries: C, C++, Rust, Go etc
- Scripts: their own source
- Sometimes written by another program:
  - Lex/yacc
  - GNOME Glade, Qt Creator
- Sometimes minified:
  - jquery.js (DocBook)
  - Web apps: CivicCRM, Moodle, LAVA server, RoundCube, Wordpress

Source of documentation

- Man pages: their own source, generated from *2man
- Info manuals: TEXinfo, DocBook
- PDF: LATEX, DocBook, *2pdf, ...
- HTML: their own source, DocBook, *2html, MarkDown, ...

Source of fonts

- Small handmade bitmap fonts: themselves
- TTF, OTF, other bitmap fonts: from FontForge SFD
- METAFONT
- Hinting bytecode (ttfautohint)

Images

- Bitmap output from XCF, PSD
- SVG can be it’s own source or not
- Destructive vs. nondesctructive editing
- Images takes with a camera
- Lossy vs. lossless
- Ray-tracers, OpenGL

Sound and Music

- Ogg, MP3 vs. WAV, FLAC
- Recorded audio
- Mixing, filtering, panning
- Software synthesizers
- MODule trackers are their own source