

gcc doc futures

aka

you are the author

gcc.gnu.org is the publisher

Benjamin Kosnik
bkoz@redhat.com
2012 GNU Cauldron Prague

outline

1. basics: what, where
2. metrics
 - docs vs. sources
 - influence
3. current sticking points
4. solutions

basics

A complete survey of gcc documentation: what exists, in what formats and why, outstanding legal issues in the GPL vs. GFDL war, where documentation is located in the source, install, and website, how documentation is packaged for releases.

See the wiki for more background:

<http://gcc.gnu.org/wiki/DocumentationOverviewIssues>

docs == all of these

1. end user

- manuals, reference material

2. developer

- internals docs
- wiki

3. web site

- release notes
- porting
- configure

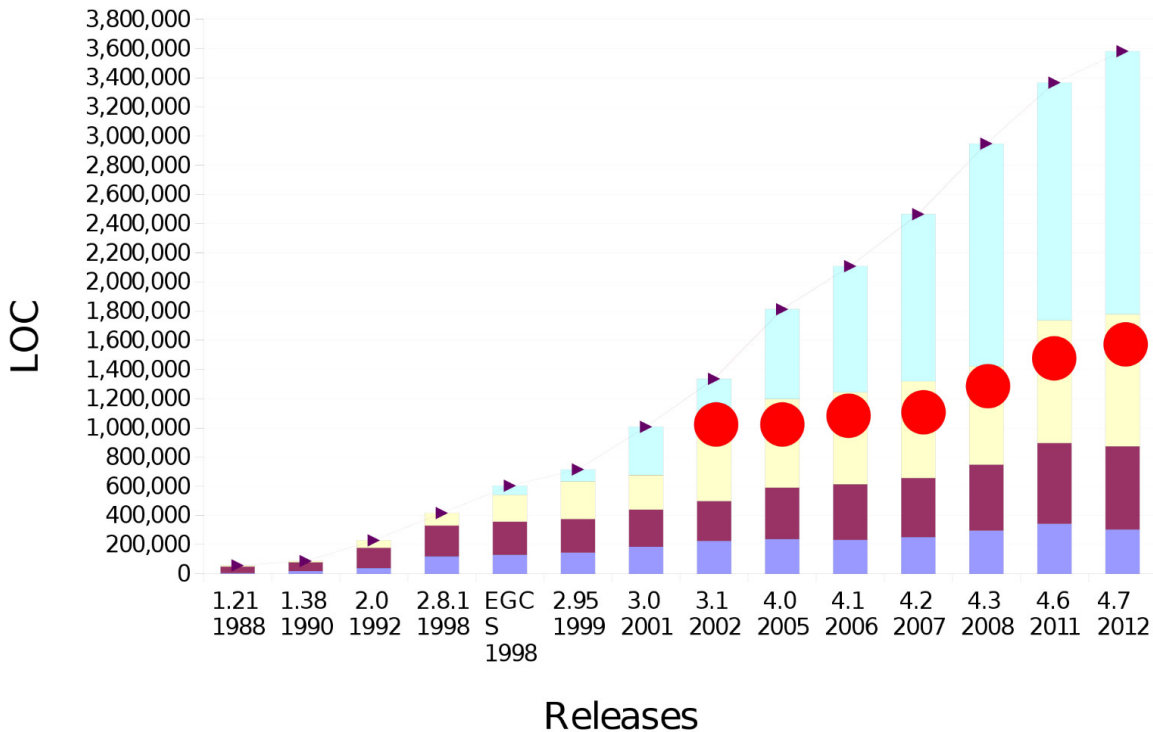
4. mailing lists

docs, oh docs!

- Documentation is the bugaboo of any long-term software project.
- All projects wish they had better docs.
- GCC is no exception:
 - GCC improvement projects
<http://gcc.gnu.org/wiki/ImprovementProjects>
 - GCC Architectural Goals
https://docs.google.com/document/pub?id=1ZfyfkB62EFaR4_g4JKm4--guz3vxm9pciOBziMHTnK4

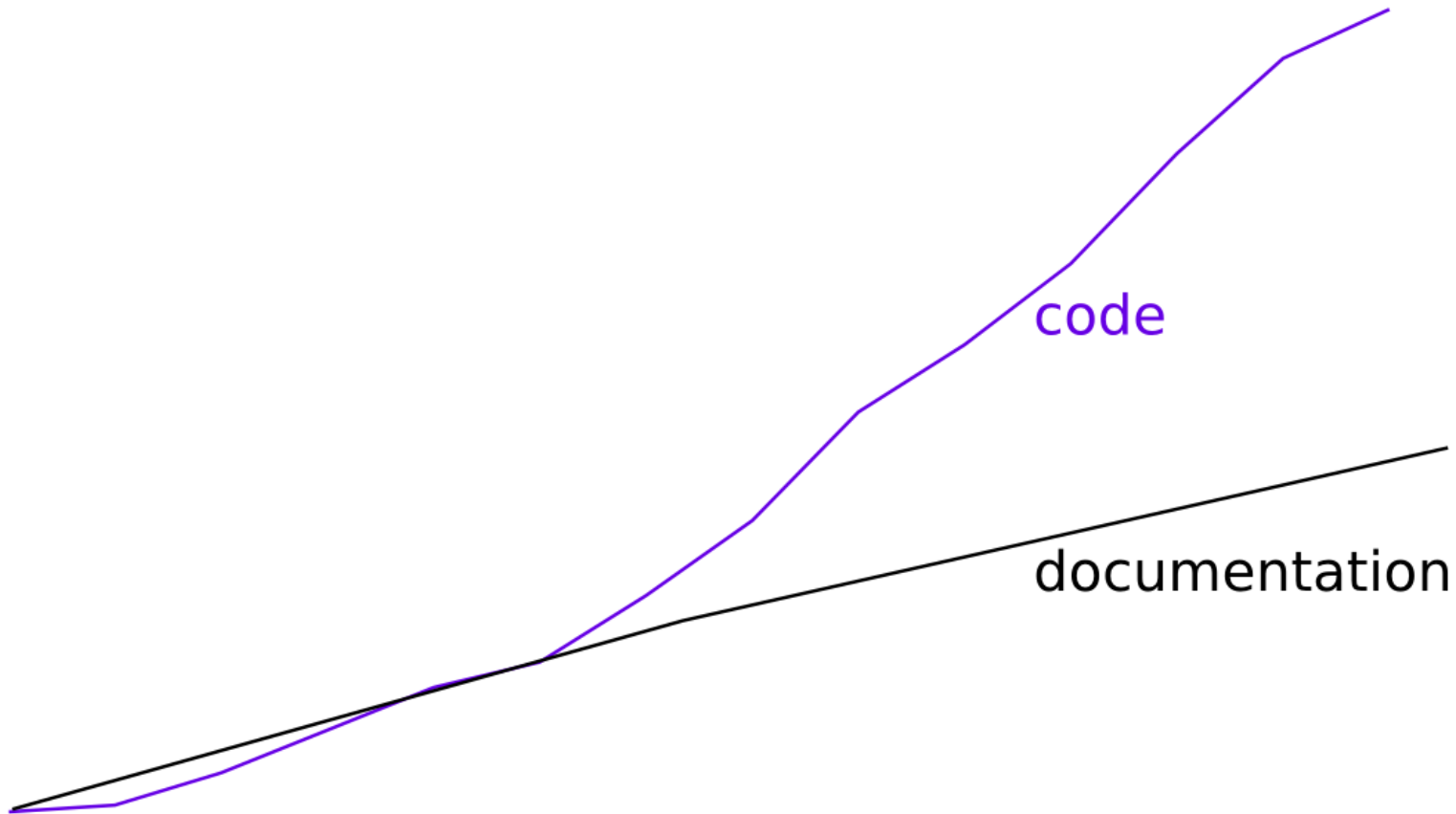
so how bad is it, really?

GCC Growth¹



¹generated using David A. Wheeler's 'SLOCCount'.

really?



really, really?

Document GCC Internals - GCC Wiki - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Inbox (1) - b.d... Home - Googl... cauldron: you ... cauldron2012... GNU Tools Ca... ImprovementP... Document GC... GCC Architec... GettingStarte...

gcc.gnu.org/wiki/Document GCC Internals define:further


GCC Wiki

BenjaminKosnik Settings Logout

Self: **Document GCC Internals**

HomePage RecentChanges FindPage HelpContents PCHHaters **Document GCC Internals**

Edit (Text) Edit (GUI) Info Subscribe Add Link Attachments More Actions:

**Help Wanted!!!**

The documentation of GCC is spread on many pages and documents in [GCC internal documentation](#), the [wiki](#) and the [web](#). We need any help that we can get to collect, organise and update it. Below there are a number of links to the main documents we think could be useful. However, we need help to actually write those documents. A well-organised collection of links would be a start. Then the most important information from those links should be extracted and written in a consistent, easy-to-use-and-read manner. Check also the [Uncategorized collection of related links](#).

Explanation of status color code

Empty Little content/badly organized/outdated Content badly organized/outdated Fair but it needs review Fair

Documents

This is a list of documents that we would like to have.

| Document | Status | Comments |
|--|--------------|-----------------------------------|
| General overview of compiler source | Empty | Information needs to be collected |
| Overview of internal compiler data structures | Empty | Information needs to be collected |
| Details of tree IR | Empty | Information needs to be collected |
| Details of GIMPLE IR | Empty | Information needs to be collected |
| How to write a new frontend | Some content | Needs further expansion |
| How to write a new optimization pass | Some content | Needs further expansion |
| How to write a new back end | Some content | Needs further expansion |

None: Document GCC Internals (last edited 2010-06-07 20:48:56 by ManuelLópezbáñez)

influence

1. assume gcc.gnu.org gets around 21k "hits" / month
2. canonical FSF manuals widely duplicated
 - lots of vendors re-package them
 - lots of search "hits" for gcc manuals, 14M for gcc manual
3. free really is a global advantage
 - gcc implementation is often "reference" implementation
 - GNU C, psABI, LSB, C++ABI, etc.

controversies

1. GPL/GFDL incompatibilities, impact on literate programming
2. FSF focused on "publishing" manuals
 - open source changes frequently
 - book publishing is slow
 - means most books are out of date
 - most of us don't read "books" anyway
3. no FSF support for creating content
4. writing is one of many software development skills
5. not everybody is going to be great at it



further out

agree on primary goal

- Not just "free software needs free documentation."
- Instead, "free software needs great free documentation"
- Want up-to-date docs.
- Want current holes filled.
- To get this, want to establish systems where current maintainers, new authors, new technical writers, and other content-makers can contribute effectively.
- focus on HTML forms

have a plan

- Can't keep falling behind the source changes
- We don't have to do everything all at once. We can break down large tasks over longer periods of time, and do a little bit with each release.
- Start focusing on 1 year/1 release improvements. Fill one hole ie build or config overviews, one pass, one internal bit at a time.
- Have an idea for 5 year/5 release improvements. Want build/config documented, accurate.
- Re-write or transfer wiki content. Could start using wiki for "staging" docs...
- Start writing release notes in usable format for re-use

solutions: maintainers/internals

- each maintainer, evaluate current state.
- each maintainer, make list of top 5 holes
<http://gcc.gnu.org/wiki/DocumentationHitList>
- for internals, could start generating with doxygen
- just start at the beginning
- consistently or automatically generate with scripts
- people who like it, can use it. people who hate it, can ignore

solutions: new authors

Think like an outside author about our current publishing platform

advantages

- excellent, innovative material
- widespread availability
- completeness of the archive

disadvantages

- weird technical platform
- odd license restrictions

Start to use HTML/web platform to our advantage.

- get metrics about what is popular
- get metrics about what is missing

solutions: new authors

Make it obvious when/how to contribute.

- we can try to let code evolve during development but have to try for timely commentary on contemporary state at regular intervals.
- doc checkpoints as part of the release process.
- propose bringing back stage 2 as a docs stage
 - May also help professionalize/bring in outside help.
 - Will help focus the maintainers on doc tasks
- switch to tools authors/editors expect for release notes etc.
- start experimenting with ebook formats