Software Heritage

Collecting, preserving and sharing the software source code of Mankind

Roberto Di Cosmo
Inria

roberto@dicosmo.org

November 7, 2017
Evry

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

Short Bio: Roberto Di Cosmo

Computer Science professor in Paris, now working at INRIA

- *30 years* of research (Theor. CS, Programming, Software Engineering, Erdos #: 3)
- *20 years* of Free and Open Source Software
- *10 years* building and directing structures for the common good

1999 *DemoLinux* – first live GNU/Linux distro

2007 *Free Software Thematic Group*
150 members 40 projects 200Me

2008 *Mancoosi project* `www.mancoosi.org`

2010 *IRILL* `www.irill.org`

2015 *Software Heritage* at INRIA

Outline

Software is everywhere

At the heart of our society

- communication, entertainment
- administration, finance
- health, energy, transportation
- education, research, politics
- …

Software is Knowledge

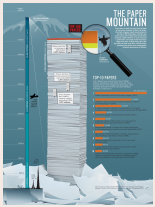
Key mediator for accessing *all* information (c) Banski

Information is a main pillar of our modern societies.

> *Absent an ability to correctly interpret digital information, we are left with [...] "rotting bits" [...] of no value.*
>
> *Vinton G. Cerf IEEE 2011*

Software is *an essential component* of modern scientific research

> *[...] the vast majority describe experimental methods or software that have become essential in their fields.*
>
> Top 100 papers (Nature, October 2014)



Software embodies our collective Knowledge and Cultural Heritage

Outline

Source code matters!

"The source code for a work means the preferred form of the work for making modifications to it."
— GPL Licence

Hello World

Program (excerpt of binary)

```
4004e6: 55
4004e7: 48 89 e5
4004ea: bf 84 05 40 00
4004ef: b8 00 00 00 00
4004f4: e8 c7 fe ff ff
4004f9: 90
4004fa: 5d
4004fb: c3
```

Program (source code)

```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

Software Source Code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.) 1985

"Programs must be written for people to read, and only incidentally for machines to execute."

Quake 2 source code (excerpt)

```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
can be removed

    return y;
}
```

Net. queue in Linux (excerpt)

```
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS  (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS      (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
        u16             qlen; /* length of virtual queue */
        u16             p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

"Source code provides a view into the mind of the designer."

Source code is special, ctd

Distinguishing features

- *executable* and *human readable* knowledge (an *all time new*)
  - even hardware is… software! (VHDL, FPGA, …)
  - *text files are forever*
- naturally *evolves* over time
  - the *development history* is key to its *understanding*
- complex: large *web of dependencies*, millions of SLOCs

In a word

- software *is not just another* sequence of bits
- a software archive *is not just another* digital archive

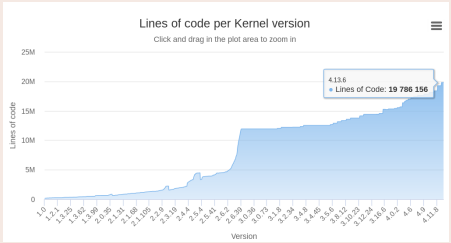
~ 50 years, a lightning fast growth

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel



… now in your pockets!

are we taking care of all this?

Outline

Software is spread all around

Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time

Where is the place …

where we can find, track and search *all* source code?

Software is fragile

Like all digital information, FOSS is fragile

- inconsiderate and/or malicious code loss (e.g., Code Spaces)
- business-driven code loss (e.g., Gitorious, Google Code)
- for obsolete code: physical media decay (data rot)

Where is the archive…

where we go if (a repository on) GitHub or GitLab.com goes away?

Software lacks its own research infrastructure

A wealth of software research on crucial issues…

- safety, security, test, verification, proof
- software engineering, software evolution
- big data, machine learning, empirical studies

If you study the stars, you go to Atacama…

… where is the *very large telescope* of source code?

We are at a turning point

Looking at the past

- a lot of old software misplaced, lost, or behind barriers, but…
- most founding fathers are still here, and willing to share
- urgent to collect their knowledge

Only a few years left.

Looking at the future

- software development and use skyrockets: more programmers, and more code!
- essential to provide a universal platform for all the future software source code

Every year that goes by makes the problem worse.

it is urgent to take action!

Outline

Software Heritage

THE GREAT LIBRARY OF SOURCE CODE

Our mission

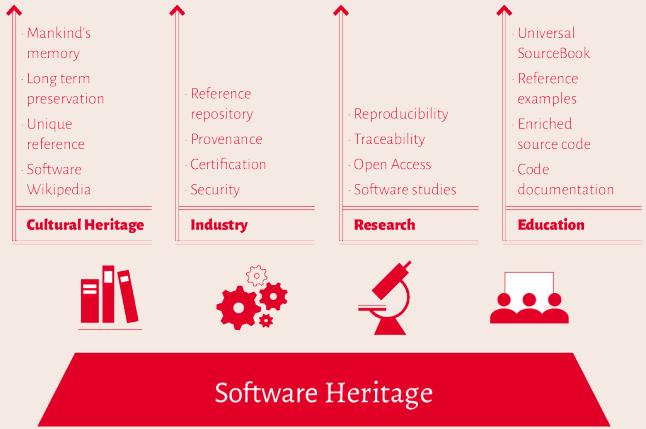
Collect, preserve and share the *source code* of *all the software* that is publicly available.

Past, present and future

Preserving the past, *enhancing* the present, *preparing* the future.

We are working on the foundations

One infrastructure to build them all



Preserving the world's software heritage

A structured archive of all of the world's software

- preserve humanity's technological and scientific knowledge
- enable continued access to all digital documents and information
- building block for thematic portals and collections

Better software for industry and society

A unique reference catalog of all industrial software components

- a single entry point to discover, explore and reuse source code
- eases vulnerability tracking for more secure software
- simplifies traceability for better software integration
- ensures long term preservation of critical software

Fostering wider education to computing

A global source referencing all software

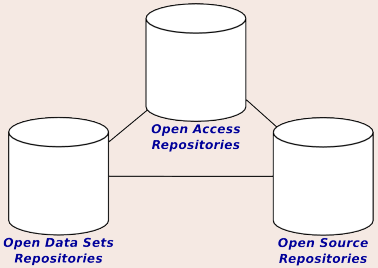
- the ultimate source book for computer science and programming classes
- intrinsic persistent identifiers for stable course materials
- enables real-world, semi-automated documentation

Supporting more accessible and reproducible science

A global library referencing all software used in all research fields

- completes the infrastructure for Open Access in science
- provides intrinsic persistent identifiers needed for scientific reproducibility
- enables large scale, verifiable software studies

The Knowledge Conservancy Magic Triangle

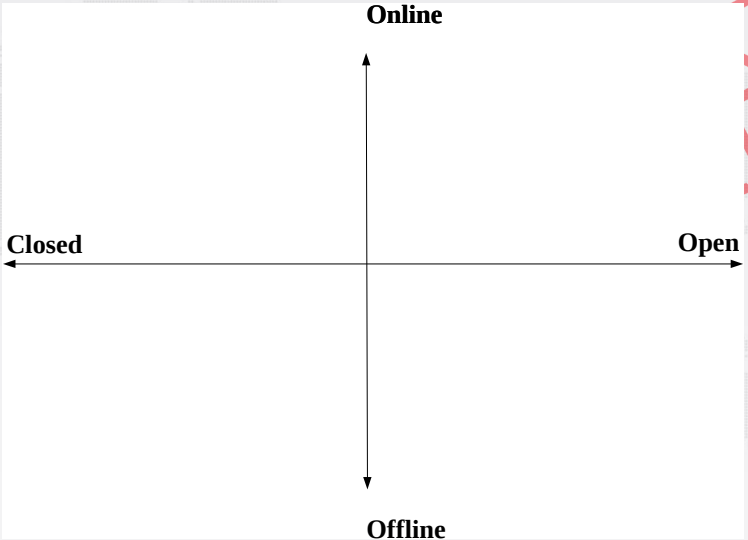


Legenda (links are important!)

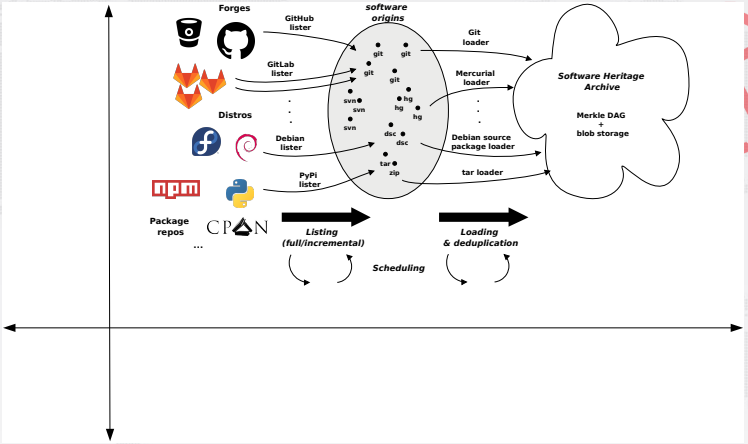
- articles: ArXiv, HAL, ...
- data: Zenodo, ...
- software: *Software Heritage* to the rescue

Outline

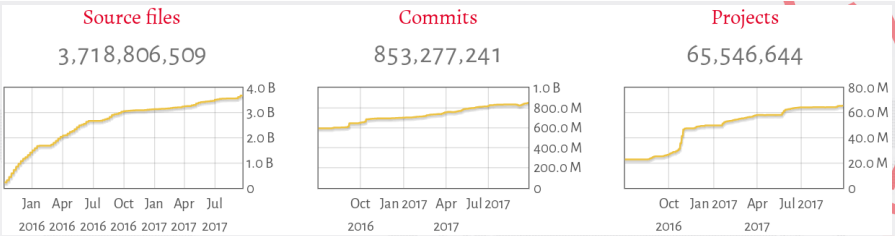
All the source code



Online, open source code: automation overview



Archive coverage



~150 TB blobs, ~5 TB database (as a graph: ~7 B nodes + ~60 B edges)

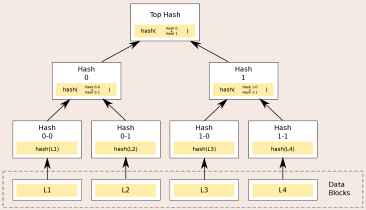
Our sources

- GitHub — full, up-to-date mirror
- Debian — automation in progress; GNU
- Gitorious, Google Code — processing (Archive Team & Google)
- Bitbucket — WIP

The *richest* source code archive already, … and growing daily!

Much more than an archive!

Merkle tree (R. C. Merkle, Crypto 1979)



Combination of

- tree
- hash function

Classical cryptographic construction

- fast, parallel signature of large data structures, built-in deduplication
- widely used in industry (e.g., Git, nix, blockchains, IPFS, …)

Using the archive

Features...

- (done) lookup by content hash
- browsing: "wayback machine" for archived code
  - (done) `http://archive.softwareheritage.org/api`
  - (in progress) via Web UI
- (in progress) download: `wget` / `git clone` from the archive
- (in progress) deposit of source code bundles directly to the archive
- (todo) provenance lookup for all archived content
- (todo) full-text search on all archived source code files

... and much more than one could possibly imagine

all the world's software development history in a single graph!

Outline

Open approach

- Transparency
- Free Software
- User and contributor community building

Objectiveness

- Facts and provenance
- *Intrinsic* identifiers
- Full development history

Long term

- Multi-stakeholder
- Nonprofit
- Replication *at all layers*

Three pillars

Science and technology

- build on sound basis
- fantastic playground for research

Resources

- fund the effort
- transfer to industry and society

Awareness

- promote public and private policies
- community building

Sponsoring Software Heritage work

Sharing the Software Heritage vision

See more

`http://www.softwareheritage.org/support/testimonials`

Going global

April 3rd, 2017: landmark Inria Unesco agreement…

https://www.softwareheritage.org/blog

September 28th, 2017

Mauritius Call on information access

Outline

You can help!

Coding

- `www.softwareheritage.org/community/developers/`
- `forge.softwareheritage.org` — our own code

Current development priorities

★★★	listers for unsupported forges, distros, pkg. managers
★★★	loaders for unsupported VCS, source package formats
★	content indexing and search
★	efficient data representation

… *all* contributions equally welcome!

Join us

- `www.softwareheritage.org/jobs` — job openings
- `wiki.softwareheritage.org` — internships

Come in, we're open!

Learn more

A white paper is available
http://bit.ly/swhpaper

Presented at iPres 2017

Software Heritage: Why and How to Preserve Software Source Code*

Roberto Di Cosmo
Inria and University Paris Diderot
France
roberto@dicosmo.org

Stefano Zacchiroli
University Paris Diderot and Inria
France
zack@irif.fr

ABSTRACT

Software is now a key component present in all aspects of our society. Its preservation has attracted growing attention over the past years within the digital preservation community. We claim that source code—the only representation of software that contains human readable knowledge—is a precious digital object that needs special handling: it must be a first class citizen in the preservation landscape and we need to take action immediately, given the increasingly more frequent incidents that result in permanent losses of source code collections.

In this paper we present Software Heritage, an ambitious initiative to collect, preserve, and share the entire corpus of publicly accessible software source code. We discuss the archival goals of the project, its use cases and role as a participant in the broader digital preservation ecosystem, and detail its key design decisions. We also report on the project road map and the current status of the Software Heritage archive that, as of early 2017, has collected more than 3 billion unique source code files and 700 million commits coming from more than 50 million software development projects.

ACM Reference format:
Roberto Di Cosmo and Stefano Zacchiroli. 2016. Software Heritage: Why and How to Preserve Software Source Code. In *Proceedings of 14th International Conference on Digital Preservation, Kyoto, Japan, September 2017 (iPRES2017)*, 10 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Software is everywhere: it powers industry and fuels innovation, it lies at the heart of the technology we use to communicate, entertain, trade, and exchange, and is becoming a key player in the formation of opinions and political powers. Software is also an essential mediator to access all digital information [1, 5] and is a fundamental pillar of modern scientific research, across all fields and disciplines [38]. In a nutshell, software embodies a rapidly growing part of our cultural, scientific, and technical knowledge.

Looking more closely, though, it is easy to see that the actual *knowledge* embedded in software is not contained into executable binaries, which are designed to run on specific hardware and software platforms and that often become, once optimized, incomprehensible for human beings. Rather, knowledge is contained in software *source code* which is, as eloquently stated in the very well crafted definition found in the GPL license [13], "the preferred form [of a program] for making modifications to it [as a developer]".

Yes, software *source code* is a unique form of knowledge which is designed to be *understood by a human being*, the developer, and at the same time is easily converted into *machine executable* form. As a digital object, software source code is also subject to a peculiar workflow: it is routinely evolved to cater for new needs and changed contexts. To really understand software source code, access to its entire development history is an essential condition.

Software source code has also established a new relevant part of our information commons [20]: the software commons—i.e., the body of software that is widely available and can be reused and modified with minimal restrictions. The raise of Free/Open Source Software (FOSS) over the past decades has contributed enormously to nurture this commons [32] and its funding principles postulate source code accessibility.

Authoritative voices have spoken eloquently of the importance of source code: Donald Knuth, a founding father of computer science, wrote at length on the importance of writing and sharing source code as a mean to understand what we want computers to do for us [19]; Len Shustek, board director of the Computer History Museum, argued that "source code provides a view into the mind of the designer" [33]; more recently the importance of source code preservation has been argued for by digital archivists [4, 22].

And yet, little *action* seem to have been put into long-term source code preservation. Comprehensive archives are available for variety of digital objects, pictures, videos, music, texts, web pages, even binary executables [18]. But source code in its own merits, despite its significance, has not yet been given the status of first class citizen in the digital archive landscape.

In this article, we claim that it is now important and urgent to focus our attention and actions to source code preservation and build a comprehensive archive of all publicly available software source code. We detail the basic principles, current status, and design choices underlying *Software Heritage*,[1] launched last year to fill what we consider a gap in the digital archiving landscape.

This article is structured as follows: Sections 2 through 4 discuss the state of the art of source code preservation and the mission of Software Heritage in context. Sections 5 and 6 detail the design and intended use cases of Software Heritage. Before concluding, Sections 7 through 9 present the data model, architecture, current status, and future roadmap for the project.

*Unless noted otherwise, all URLs in the text have been retrieved on March 30, 2017.

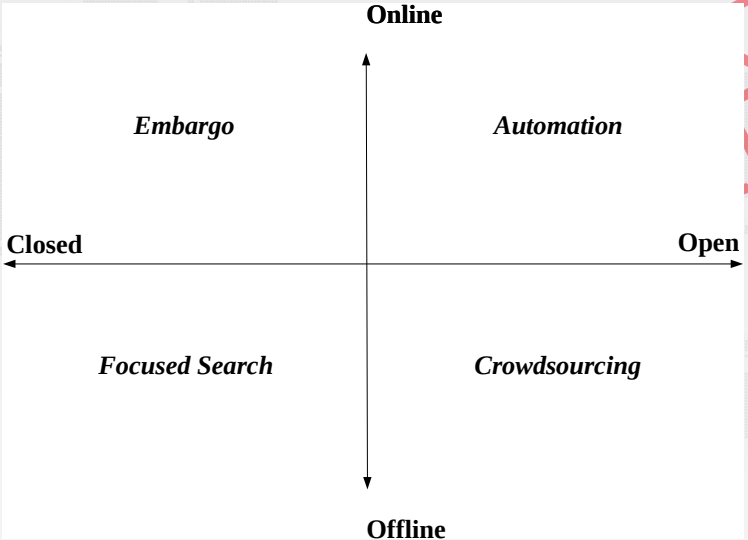
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
iPRES2017, Kyoto, Japan
© 2017 Copyright held by the owner/author(s). 978-x-xxxx-xxxx-x/YY/MM...$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

[1] https://www.softwareheritage.org

Get involved

sponsoring / partnership	sponsorship.softwareheritage.org
working groups, leads	wiki.softwareheritage.org
our own code	forge.softwareheritage.org

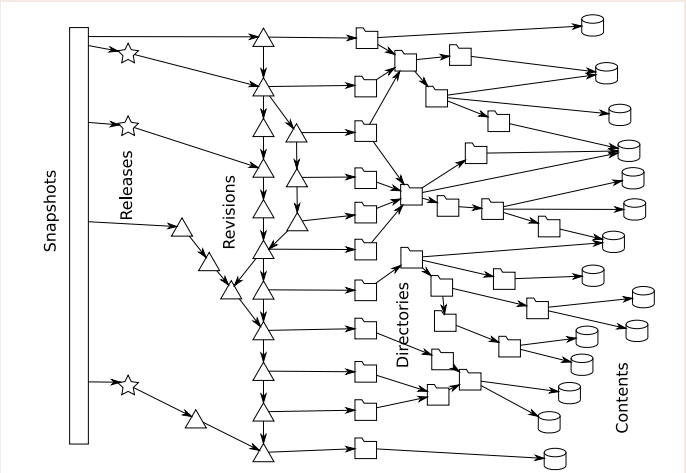
All the source code, strategies



Outline

The archive in a few pictures

A giant (extended) Merkle DAG



Web API

First public version of our Web API (Feb 2017)
https://archive.softwareheritage.org/api/

Features

- pointwise browsing of the Software Heritage archive
  - … releases ⟶ revisions ⟶ directories ⟶ contents …
- full access to the metadata of archived objects
- crawling information
  - *when have you last visited this Git repository I care about?*
  - *where were its branches/tags pointing to at the time?*

Complete endpoint index

https://archive.softwareheritage.org/api/1/

A tour of the Web API — origins & visits

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      git/url/https://github.com/hylang/hy
{ "id": 1,
  "origin_visits_url": "/api/1/origin/1/visits/",
  "type": "git",
  "url": "https://github.com/hylang/hy"
}
```

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      1/visits/
[ ...,
  { "date": "2016-09-14T11:04:26.769266+00:00",
    "origin": 1,
    "origin_visit_url": "/api/1/origin/1/visit/13/",
    "status": "full",
    "visit": 13
  }, ...
]
```

A tour of the Web API — snapshots

```
GET https://archive.softwareheritage.org/api/1/origin/ \
      1/visit/13/
{ ...,
  "occurrences": { ...,
    "refs/heads/master": {
      "target": "b94211251...",
      "target_type": "revision",
      "target_url": "/api/1/revision/b94211251.../"
    },
    "refs/tags/0.10.0": {
      "target": "7045404f3...",
      "target_type": "release",
      "target_url": "/api/1/release/7045404f3.../"
    }, ...
  },
  "origin": 1,
  "origin_url": "/api/1/origin/1/",
  "status": "full",
  "visit": 13
}
```

A tour of the Web API — revisions

```
GET https://archive.softwareheritage.org/api/1/revision/ \
      6072557b6c10cd9a21145781e26ad1f978ed14b9/
{
  "author": {
    "email": "tag@pault.ag",
    "fullname": "Paul Tagliamonte <tag@pault.ag>",
    "id": 96,
    "name": "Paul Tagliamonte"
  },
  "committer": { ... },
  "date": "2014-04-10T23:01:11-04:00",
  "committer_date": "2014-04-10T23:01:11-04:00",
  "directory": "2df4cd84e...",
  "directory_url": "/api/1/directory/2df4cd84e.../",
  "history_url": "/api/1/revision/6072557b6.../log/",
  "merge": false,
  "message": "0.10: The Oh f*ck it's PyCon release",
  "parents": [ {
     "id": "10149f66e...",
     "url": "/api/1/revision/10149f66e.../"
```

A tour of the Web API — contents

```
GET https://archive.softwareheritage.org/api/1/content/ \
      adc83b19e793491b1c6ea0fd8b46cd9f32e592fc/
{
  "data_url": "/api/1/content/sha1:adc83b19e.../raw/",
  "filetype_url": "/api/1/content/sha1:.../filetype/",
  "language_url": "/api/1/content/sha1:.../language/",
  "length": 1,
  "license_url": "/api/1/content/sha1:.../license/",
  "sha1": "adc83b19e...",
  "sha1_git": "8b1378917...",
  "sha256": "01ba4719c...",
  "status": "visible"
}
```

Caveats

- rate limits apply throughout the API
- blob download available for selected contents