Software Heritage
Building the Universal Software Archive and Knowledge Base

Roberto Di Cosmo
roberto@dicosmo.org

October 17th, 2017
Software is Pervasive

At the heart of our society

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

Roberto Di Cosmo www.dicosmo.org
Software is Pervasive

At the heart of our society
- communication, entertainment
- administration, finance
- health, energy, transportation
- education, research, politics
- ...

At the heart of technology
- house appliances $\approx 10 \text{M SLOC}$
- phones $\approx 20 \text{M SLOC}$, cars $\approx 100 \text{M SLOC}$
- Internet of things, …
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 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 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)

Sofware embodies our Knowledge and Cultural Heritage
"The source code for a work means the preferred form of the work for making modifications to it."

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

— GPL Licence

Hello World
"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
"The source code for a work means the preferred form of the work for making modifications to it."

— GPL Licence

Program (source code)

```c
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("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
```
Software Source Code is special

Harold Abelson, Structure and Interpretation of Computer Programs

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

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

Roberto Di Cosmo www.dicosmo.org www.softwareheritage.org October 17th, 2017 5 / 26
Harold Abelson, Structure and Interpretation of Computer Programs

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

Quake 2 source code (excerpt)

```c
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalves = 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 * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this
    // can be removed

    return y;
}
```

Net. queue in Linux (excerpt)

```c
/*
 * SFB uses two B[i][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 "qln" */
struct sfb_bucket {
    u16 qlen; /* length of virtual queue */
    u16 p_mark; /* marking probability */
};
```
Software Source Code is special

Harold Abelson, Structure and Interpretation of Computer Programs

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

Quake 2 source code (excerpt)

```c
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalves = 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 * ( threehalves - ( x2 * y * y ) ); // 1st iteration
    // y = y * ( threehalves - ( x2 * y * y ) ); // 2nd iteration, this can be removed
    return y;
}
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

Net. queue in Linux (excerpt)

```c
/*
 * SF8 uses two B[I][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 rxFhash 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 */

/* SF8 algo uses a virtual queue, named "B1n" */
struct sfb_bucket {
    u16 qlen; /* length of virtual queue */
    u16 p_mark; /* marking probability */
};
```
"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton
~ 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!
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?
Fashion victims

- many disparate development platforms
- a myriad places where distribution may happen
- projects tend to migrate from one place to another over time
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?
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)
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
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.
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.
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!
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
Preserving and sharing 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
- precious resource for education
- WikiPedia of software
Better software for industry and society

Universal knowledge base for all industrial software components

- a single entry point to discover, explore and reuse source code
- simplifies traceability (licensing, supply chain management)
Universal knowledge base for all industrial software components

- a single entry point to discover, explore and reuse source code
- simplifies traceability (licensing, supply chain management)

Reference platform for Big Code

- unique point of observation for all software development
- big data, machine learning paradise: trends, coding patterns, code completion…
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
All the source code
Online, open source code: automation overview

Diagram showing the software origins and various package repos. The software Heritage Archive is connected to various software origins, including GitHub, GitLab, and Mercurial. The diagram also shows listsers and loaders for different packages, such as Debian, PyPI, and npm. The process involves scheduling, listing (full/incremental), loading, and deduplication. The blob storage is indicated for Merkle DAG.
~150 TB blobs, ~5 TB database (as a graph: ~7 B nodes + ~60 B edges)
~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
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
- widely used (e.g., Git, blockchains, IPFS, ...)
- built-in deduplication
Using the archive

Features...

- (done) **lookup** by content hash
- **browsing:** "wayback machine" for archived code
  - (done) [http://archive.softwareheritage.org/api](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
Using the archive

Features…

- (done) **lookup** by content hash
- **browsing**: "wayback machine" for archived code
  - (done) [http://archive.softwareheritage.org/api](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!
Our principles

- 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

Software Heritage
Our principles

Software Heritage

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

Roberto Di Cosmo
www.dicosmo.org

www.softwareheritage.org
October 17th, 2017

Three pillars

Science and technology
build on sound basis
Three pillars

Science and technology build on sound basis

Resources fund the effort
Three pillars

Science and technology build on sound basis

Resources fund the effort

Awareness
- promote public and private policies
- community building
Sponsoring Software Heritage work

- >= 100Ke/year
- >= 50Ke/year
- >= 25Ke/year
- >= 10Ke/year
Sharing the Software Heritage vision

See more

http://www.softwareheritage.org/support/testimonials
April 3rd, 2017: landmark Inria Unesco agreement…

https://www.softwareheritage.org/blog

September 28th, 2017

September 2017: Mauritius Call on information access
An unique opportunity

Library of Alexandria of code

Take *urgent* action to
- recover the past
  - founding fathers still here
- structure the future
  - programming skyrockets
An unique opportunity

Library of Alexandria of code

Take *urgent* action to
- recover the past
  - founding fathers still here
- structure the future
  - programming skyrockets

A CERN for Software

Build a *common infrastructure*
- supporting industry needs
- enabling software research
- fostering better science
- for society as a whole

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills
Everybody is needed

| partnerships | mailto:roberto@dicosmo.org |
| research    | tackle the new challenges |
| funding     | sponsorship.softwareheritage.org |
| our own code | forge.softwareheritage.org |
All the source code, strategies

Embargo

Automation

Focused Search

Crowdsourcing

Online

Closed

Offline

Open
Selected research challenges: building the archive

- Distributed infrastructure
  - efficient p2p protocols
- Metadata
  - 65M projects
- Data compression
- Software phylogenetcs
Selected research challenges: using the archive

- Project classification
  - 65M projects
- Code search
  - AST, functions, features
- Software as Big Data
- Efficient data representation