Software Heritage
Building the Universal Software Archive for Open Science

Roberto Di Cosmo
roberto@dicosmo.org

March 15th, 2018
1. Software is everywhere around us
2. Software source code for Science!
3. The Software Heritage initiative
4. Status
5. Building for the long term
6. Conclusion
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 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 Knowledge and Cultural Heritage**
"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)

```assembly
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
```
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)

```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 0[][][] : 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.”
~ 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?
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)

Where is the archive...

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

1. Software is everywhere around us
2. Software source code for Science!
3. The Software Heritage initiative
4. Status
5. Building for the long term
6. Conclusion
We face a science crisis

"Sub-prime science"? (Nicholas Humprey)

- inconsistencies
- data corruption, fraud
- non reproducible findings…  (picture from Nature, Sep. 2015)

The world starts noticing

October 2013

John Oliver, *Science* May 2016
How we built our scientific knowledge

The experimental method
- make an observation
- formulate an hypothesis
- set up an experiment
- formulate a theory

And then we reproduce and verify.

Reproducibility is the key

non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934
For an experiment involving software, we need

open access to the scientific article describing it

open data sets used in the experiment

source code of all the components

environment of execution

stable references between all this

Remark

The first two items are already widely discussed!

... what about software?
Collberg’s report from the trenches

Analysis of 613 papers

- 8 ACM conferences: ASPLOS’12, CCS’12, OOPSLA’12, OSDI’12, PLDI’12, SIGMOD’12, SOSP’11, VLDB’12
- 5 journals: TACO’9, TISSEC’15, TOCS’30, TODS’37, TOPLAS’34

all very practical oriented

The basic question

can we get the code to build and run?
The result

This can be debated (see http://cs.brown.edu/~sk/Memos/Examining-Reproducibility/), but...

... that’s a whopping 81% of non reproducible works!
URL decay disrupts the web of reference

Web links are not permanent (even permalinks)

there is no general guarantee that a URL… which at one time points to a given object continues to do so


URLs used in articles decay!

Analysis of IEEE Computer (Computer), and the Communications of the ACM (CACM): 1995-1999

- the half-life of a referenced URL is approximately 4 years from its publication date


### Scholar roster of broken links

**An example from Astronomy**

<table>
<thead>
<tr>
<th>Domain</th>
<th>links (broken)</th>
<th>.html</th>
<th>.txt</th>
<th>.dat</th>
<th>.gz</th>
<th>.tar</th>
<th>.fits</th>
<th>tilde</th>
</tr>
</thead>
<tbody>
<tr>
<td>cplearn.harvard.edu</td>
<td>802 (110)</td>
<td>336 (70)</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>5 (4)</td>
<td>1</td>
</tr>
<tr>
<td>nasa.gsfc.nasa.gov</td>
<td>640 (33)</td>
<td>423 (27)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.nstic.edu">www.nstic.edu</a></td>
<td>498 (61)</td>
<td>205 (29)</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15 (10)</td>
</tr>
<tr>
<td>asic.harvard.edu</td>
<td>471 (112)</td>
<td>212 (99)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (1)</td>
</tr>
<tr>
<td>sci.spitzer.caltech.edu</td>
<td>427 (194)</td>
<td>135 (70)</td>
<td>3 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>cfa-www.harvard.edu</td>
<td>352 (68)</td>
<td>277 (52)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>54 (17)</td>
</tr>
<tr>
<td>archive.stsci.edu</td>
<td>308 (58)</td>
<td>57 (9)</td>
<td>2</td>
<td>1 (0)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.ipac.caltech.edu">www.ipac.caltech.edu</a></td>
<td>285 (14)</td>
<td>209 (12)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.amc.cairn.eu">www.amc.cairn.eu</a></td>
<td>211 (21)</td>
<td>12 (6)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7 (5)</td>
</tr>
<tr>
<td>space.mit.edu</td>
<td>193 (10)</td>
<td>58 (5)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2 (1)</td>
</tr>
<tr>
<td><a href="http://www.astroniu.edu">www.astroniu.edu</a></td>
<td>186 (4)</td>
<td>103 (1)</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td><a href="http://www.esc.org">www.esc.org</a></td>
<td>186 (58)</td>
<td>54 (22)</td>
<td>1 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4 (1)</td>
</tr>
<tr>
<td>esa.ipac.caltech.edu</td>
<td>166 (5)</td>
<td>38 (1)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.seds.org">www.seds.org</a></td>
<td>156 (2)</td>
<td>109 (1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>hea-www.harvard.edu</td>
<td>125 (37)</td>
<td>42 (17)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>26 (16)</td>
</tr>
<tr>
<td>physics.nist.gov</td>
<td>125 (3)</td>
<td>63 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.nasa.edu">www.nasa.edu</a></td>
<td>120 (3)</td>
<td>50 (2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.ubpava.es">www.ubpava.es</a></td>
<td>118 (33)</td>
<td>23 (19)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (1)</td>
</tr>
<tr>
<td><a href="http://www.astroph.princeton.edu">www.astroph.princeton.edu</a></td>
<td>115 (31)</td>
<td>43 (14)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>53 (12)</td>
</tr>
<tr>
<td>astro.navy.mil</td>
<td>110 (27)</td>
<td>98 (22)</td>
<td>3 (3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1 (1)</td>
</tr>
</tbody>
</table>

This table lists total number of links and broken links (HTTP status codes 3xx, 4xx, and 5xx) to top domains (domains with over 100 links) found within articles published in the four main astronomy journals between 1997 and 2008. The table also shows, for each domain, the portion of links to common filename extensions, as well as links that contain the tilde character.

doi:10.1371/journal.pone.0104798

---

**How Do Astronomers Share Data?**

Pepe, Goodman, Muench, Crosas, Erdmann

dx.doi.org/10.1371/journal.pone.0104798

**PLOS August 28, 2014**
Example: doi:10.1109/MSR.2015.10

- to find what 10.1109/MSR.2015.10 is, go to a resolver (e.g. doi.org)
- this returns http://ieeexplore.ieee.org/document/7180064/
- at this URL we find …

Architecture of the DOI infrastructure

- DOI resolution can change
- content at URL can change
- no intrinsic way of noticing
- persistence based on good will of multiple parties
No catalog, no archive, no references, … and 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 skyrocket: 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 available

Past, present and future

Preserving the past, enhancing the present, preparing the future
1. Software is everywhere around us
2. Software source code for Science!
3. The Software Heritage initiative
4. Status
5. Building for the long term
6. Conclusion
Architecture (simplified)
Archive coverage

Current sources

- live: GitHub, Debian
- one-off: Gitorious, Google Code, GNU
- WIP: Bitbucket

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

The richest public source code archive, … and growing daily!
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

Roberto Di Cosmo www.dicosmo.org
Demo time (breaking news!)

Browsing the archive contents
- archive.softwareheritage.org

Archiving scientific software via HAL
- open on the Inria instance, see the deposit guide at http://bit.ly/swhdeposithalen
Outline

1. Software is everywhere around us
2. Software source code for Science!
3. The Software Heritage initiative
4. Status
5. Building for the long term
6. Conclusion
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*
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

>= 100K/year

>= 50K/year

>= 25K/year

>= 10K/year

Microsoft

intel

SOCIETE GENERALE

HUAWEI

GitHub

DANS

Nokia Bell Labs

Data Archiving and Networking Services
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
1. Software is everywhere around us
2. Software source code for Science!
3. The Software Heritage initiative
4. Status
5. Building for the long term
6. Conclusion
Come in, we’re open!

Software Heritage

www.softwareheritage.org  @swheritage

Everybody is needed

become a mirror  mailto:roberto@dicosmo.org
our code  forge.softwareheritage.org
research  use cases, new challenges
funding  sponsorship.softwareheritage.org
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
All the source code
Online, open source code: automation overview