Software Heritage

Does software preservation have an ethical impact on society?

Morane Gruenpeter

Software engineer and metadata specialist
Inria, Software Heritage

morane@softwareheritage.org

November 30th, 2020
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
Short Bio: Morane Gruenpeter

Goal: Building the Semantic Web of Free and Open Source Software

1999-2011  Harpist
2012-2015  Licence in Computer Science CNAM
2015-2017  Master STL - M2 R&D UPMC
2017  Internship Software Heritage (SWH)
2018-2019  European project EU2020 CROSSMINER (on SWH team)
2020-2022  European project FAIRsFAIR (on SWH team)

Working groups for Open Science and digital preservation

- the Research Data Alliance’s Software Source Code Interest Group (SSC IG),
- the FORCE11’s Software Citation Implementation Working Group (SCI WG),
- the joint RDA & FORCE11 Software Identification Working Group (SCID WG)
Goals today

Terminology
- understand terminology around software and source code

History
- a few elements to grasp recent and not so recent evolutions

Society
- software as a key to view society
- review initiatives working on software preservation
Software is all around us

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

The World Wide Web, 1989, at CERN on a NeXT machine

“When somebody has learned how to program a computer … You’re joining a group of people who can do incredible things. They can make the computer do anything they can imagine.”

From An Insight, An Idea with Tim Berners-Lee (2013)
What is software?
Software definition

Encyclopædia Britannica

“Software, instructions that tell a computer what to do. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system. The term was coined to differentiate these instructions from hardware—i.e., the physical components of a computer system.”

Software as a concept
- software project / entity
- the creators and the community around it

Software artifact
- the binaries for different environments
- the software source code for each version
This is *software*?

What about *software source code*?
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
“The source code for a work means the preferred form of the work for making modifications to it.”

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");
}
Source code is special

Executable and human readable knowledge

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

Harold Abelson

Software evolves over time

- projects may last decades
- the development history is key to its understanding

Complexity

- millions of lines of code
- large web of dependencies
  - easy to break, difficult to maintain
- sophisticated developer communities
# THE MASTER IGNITION ROUTINE IS DESIGNED FOR USE BY THE FOLLOWING LEM PROGRAMS: P12, P40, P42, P61, P63.
# IT PERFORMS ALL FUNCTIONS IMMEDIATELY ASSOCIATED WITH APS OR DPS IGNITION: IN PARTICULAR, EVERYTHING LYING
# BETWEEN THE PRE-IGNITION TIME CHECK -- ARE WE WITHIN 45 SECONDS OF TIG? -- AND TIG + 20 SECONDS, WHEN DPS
# PROGRAMS THROTTLE UP.
# 
# VARIATIONS AMONG PROGRAMS ARE ACCOMMODATED BY MEANS OF TABLES CONTAINING CONSTANTS (FOR AVEGEXIT, FOR
# WAITLIST, FOR PINBALL) AND TCF INSTRUCTIONS. USERS PLACE THE ADDRESSES OF THE HEAD OF THE APPROPRIATE TABLE
# (OF P12TABLE FOR P12M, FOR EXAMPLE) IN ERASABLE REGISTER 'WHICH' (E4). THE IGNITION ROUTINE THEN INDEXES BY
# WHICH TO OBTAIN OR EXECUTE THE PROPER TABLE ENTRY. THE IGNITION ROUTINE IS INITIATED BY A TCF BURNBABY,
# THROUGH BANK JUMP IF NECESSARY. THERE IS NO RETURN.
# 
# THE MASTER IGNITION ROUTINE WAS CONCEIVED AND EXECUTED, AND (NOTA BENE) IS MAINTAINED BY ADLER AND EYLES.
#
# MONI SOIT QUI MAL Y PENSE
#
# **************************************************************************
# TABLES FOR THE IGNITION ROUTINE
# **************************************************************************
#
# NOLI SE TANGERE

P12TABLE VN 0674 # (0)
TCF ULGNOT # (1)
TCF COMFAIL3 # (2)
TCF GOCUTOFF # (3)
TCF TASKOVER # (4)
TCF P12SPOT # (5)
DEC 0 # (6) NO ULLAGE
EBANK= WHICH
2CADR SERVEXIT # (7)
TCF DISPATCH # (11)
TCF WAITBIT # (12)
TCF P12IGN # (13)
P40TABLE VN 0648 # (0)
Version Control System timeline

Version control system (VCS)
- records changes made to a (set of) source code file(s)
- allows to operate on versions: diff/merge/fork/recover etc.
- essential tool for software development

Three decades of evolution

- Local VCS
  - RCS (1982)
  - CVS (1990)
  - Subversion (2000)

- Centralized VCS
  - CVS (1990)

- Distributed VCS
  - Mercurial (2005)
  - Git (2005)

- Software Heritage
  - Official launch of archive (2016)
  - 90 M repositories archived sep 2019
Adoption

**Stack Overflow [Survey 2018]**

<table>
<thead>
<tr>
<th>Version Control</th>
<th>All Respondents</th>
<th>Professional Developers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Git</td>
<td>87.2%</td>
<td></td>
</tr>
<tr>
<td>Subversion</td>
<td>16.1%</td>
<td></td>
</tr>
<tr>
<td>Team Foundation Version Control</td>
<td>10.9%</td>
<td></td>
</tr>
<tr>
<td>Zip file back-ups</td>
<td>7.9%</td>
<td></td>
</tr>
<tr>
<td>Copying and pasting files to network</td>
<td>7.9%</td>
<td></td>
</tr>
<tr>
<td>shares</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I don't use version control</td>
<td>4.8%</td>
<td></td>
</tr>
<tr>
<td>Mercurial</td>
<td>3.6%</td>
<td></td>
</tr>
</tbody>
</table>

74,398 responses; select all that apply

Git is the dominant choice for version control for developers today, with almost 90% of developers checking in their code via Git.

---

**In numbers**

**GitHub [Octoverse 2017] [Blog 2018]**
- 100,000,000+ repositories
- 40,000,000+ developers worldwide

**Bitbucket** [Blog 2019]
- 28,000,000+ repositories
- 10,000,000+ developers worldwide

**GitLab** [Blog 2019]
- 1,000,000 MRs March 19’

---

Let’s use it!
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
Software Heritage in a nutshell

Collect, preserve and share *all* software source code

Preserving our heritage, enabling better software and better science for all

**Reference catalog**
find and reference all software source code

**Universal archive**
preserve all software source code

**Research infrastructure**
enable analysis of all software source code
Software Heritage

As of today the archive already contains and keeps safe for you the following amount of objects:

**Source files**: 8,846,381,610

**Commits**: 1,880,663,008

**Projects**: 140,348,311

**Directories**: 7,506,954,410

**Authors**: 38,603,337

**Releases**: 15,051,940
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!
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
November, 2018 at the UNESCO headquarters experts signed the engagement:

- **Recognise** software source code as a precious asset of humankind
- **Support** the development of shared infrastructures
- **Foster** international collaboration to build a common framework

*see full text*
"Preserving Software Source Code is crucial and captures human civilization"

"It includes the need to raising awareness of the importance of SSC among decision makers, recognizing Software creators and the contributions of women and minorities to digital innovations".

"Considering that documents produced and preserved overtime, in all their analog and digital forms through time and space, constitute the primary means of knowledge creation and expression, having an impact on all areas of humanity’s civilization and its further progress".

"Considering at the same time that the preservation of, and long term accessibility to documentary heritage underpins fundamental freedoms of opinion, expression and information as human rights".
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
Joining forces in the urgent effort to preserve humankind’s source code.

- A testament to the importance of software source code preservation
- A multi-partners strategy for archiving code
- A range of storage solutions, from real-time to long-term storage
The data is encoded on frames with 8.8 million pixels each.
Paris Call on Software Source Code

“We call to] support efforts to gather and preserve the artifacts and narratives of the history of computing, while the earlier creators are still alive”

- **Rescue** Legacy Software from different media
  - physical
  - digital
    - legacy / unsupported
    - recent / supported

- **Curate** the code
  - reconstructing the development history
  - collecting metadata

- And **illustrate** with dedicated presentations
SWHAP: Four phases workflow to streamline the effort
The control code for the music synthesizer TAU2

FORTRAN II and TAUmus command language

Istituto di Elaborazione dell’Informazione CNR

Group led by the late M° P. Grossi

Le Sacre du Printemps (ABSTRACT)
SWHAP@PISA: Capturing metadata in branch master

```json
{
    "@context": "https://doi.org/10.5063/schema/codemeta-2.0",
    "@type": "SoftwareSourceCode",
    "identifier": "TAUmus",
    "description": "Code controlling TAU2, a music synthesizer of the '70s.",
    "name": "TAUmus",
    "codeRepository": "https://github.com/Unipisa/TAUmus",
    "applicationCategory": "Computer Music",
    "license": "Copyright (C) 1972 Pietro Grossi",
    "version": "1.1"
}
```
SWHAP@PISA: Recreating development history in branch SourceCode

Unipisa / TAUmus

Commits on Oct 8, 2019

v1.1 -

Pietro Grossi authored and laurab1 committed on Oct 16, 1972

v1.0 -

Pietro Grossi authored and laurab1 committed on Sep 16, 1972
SWHAP@PISA: Archiving source code with Save Code Now

Save code now

Origin type: git
Origin url: 
Submit
SWHAP@PISA: Referencing source code on SWH

To reference or cite the objects present in the Software Heritage archive, permalinks based on persistent identifiers must be used instead of copying and pasting the URL from the address bar of the browser (as there is no guarantee the current URI scheme will remain the same over time).

Select below a type of object currently browsed in order to display its associated persistent identifier and permalink.

- Content
- Directory
- Revision
- Snapshot

Add origin info
Add selected lines info

Copy identifier
Copy permalink

SUBROUTINE_CALMUS.FOR

1 SUBROUTINE CALMUS
2 COMMON FR,T ,R,5,INIT,IF3
3 *ALR, DURA, NOA, IP, KONT
4 DIMENSION NHH(190),J(19),NNT,
5 FRE (500), T(5889) M(400)
6 REAL KFF(8)
7 READ(5,10)N,N1,N2
8 10 FORMAT(314)
9 N=4
10 LN=2
11 KK=0
12 KONT=0
13 KONT=0
14 L=9
15 LL=0
16 KP=0
17 S K=0
18 IF(K.LE.N)GO TO 12
19 LENL=1
20 IF(LN.GT.5)GO TO 50
21 GO TO 8
Outline

1. Introduction
2. The knowledge is in the source code!
3. Software Heritage: the universal source code archive
4. The Paris call: Software Source Code is our Heritage
5. Strategies for archiving
6. Software preservation and ethics
## Software Heritage ethical charter

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Avoid harm</td>
<td></td>
</tr>
<tr>
<td>- Protect Personal Data</td>
<td></td>
</tr>
<tr>
<td>- Avoid useless copies</td>
<td></td>
</tr>
<tr>
<td>- Care about derived data</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Avoid harm</td>
<td></td>
</tr>
<tr>
<td>- Protect Personal Data</td>
<td></td>
</tr>
<tr>
<td>- Maintain coherent terms of use</td>
<td></td>
</tr>
<tr>
<td>- Ensure fair and non discriminatory access</td>
<td></td>
</tr>
<tr>
<td>- Foster Collaboration</td>
<td></td>
</tr>
</tbody>
</table>
Software preservation and ethics

Discussion topics

1. Transparency and conflict of interest
2. Digital divide
3. Ownership of the physical archive
4. Responsibility for the incorrect use of source codes
5. The positive potential of this archive
6. Possible role of philosophers in interdisciplinary teams with scientists
Thank you! Any questions?

contact: morane@softwareheritage.org

Jean-François Abramatic, Roberto Di Cosmo, Stefano Zacchiroli
*Building the Universal Archive of Source Code*, Communications of the ACM, October 2018

Roberto Di Cosmo, Morane Gruenpeter, Stefano Zacchiroli

Acknowledgements

- Roberto Di Cosmo, Founder and Director of Software Heritage
- Leah Gruenpeter-Gold, PhD Philosophy Dept., Tel Aviv University