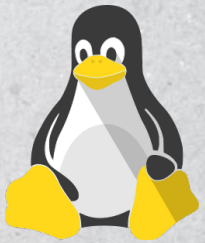
Systems Advanced II
Linux

Software Licenses



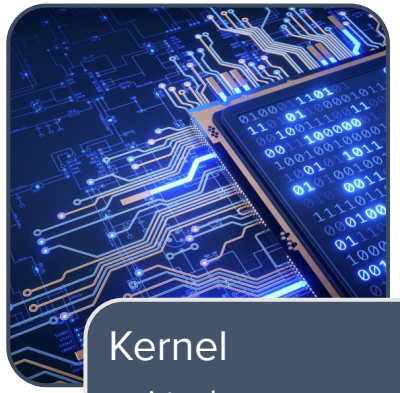
DE HOGESCHOOL MET HET NETWERK

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

Doelstellingen

- De student:
  - De student kan Netwerk-services installeren, configureren en onderhouden.
  - De student kan microservices-infrastructuur opzetten en beheren.
  - De student kan een (eigen) cloud systeem opzetten a.d.h.v. opgelegde voorwaarden.
  - De student kan een systeem beveiligen.

Systems Advanced II - Linux



Kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

Clusters

- Storage
- Kubernetes

Licensing & open-source software

Systems Advanced II - Linux

kernel

- virtual memory
- user space and kernel space
- processes and threads
- sockets & pipes
- systemd
- Docker

Networking

- routing
- firewall
- eBPF
- certificates
- ssh
- latency
- bandwidth

Services

- nfs
- load balancers and reverse proxies
- databases
- logging

Clusters

- Storage
- Kubernetes

Licensing & open-source software

Contents

- Origins
  - Counterculture
  - Counterculture in IT
- software patents
- software licensing
- OSF
- types of open-source licensing
- Companies
- landscape today
- tools

Historical context

- 1939-1945: WWII
- 1950-ies: Beatniks and the Beat generation
  - anti-conformist post-war youth movement in New York
  - exploration of American and Eastern religions and spiritualism
  - **rejection of economic materialism**
  - explicit portrayals of the human condition
  - experimentation with psychedelic drugs
  - sexual liberation
- 1955-1975: Vietnam War
- 1958-1969: hippies and the counterculture
  - beatnik subculture
  - women's liberation
  - (racial) equality and civil rights movement
  - environmentalism

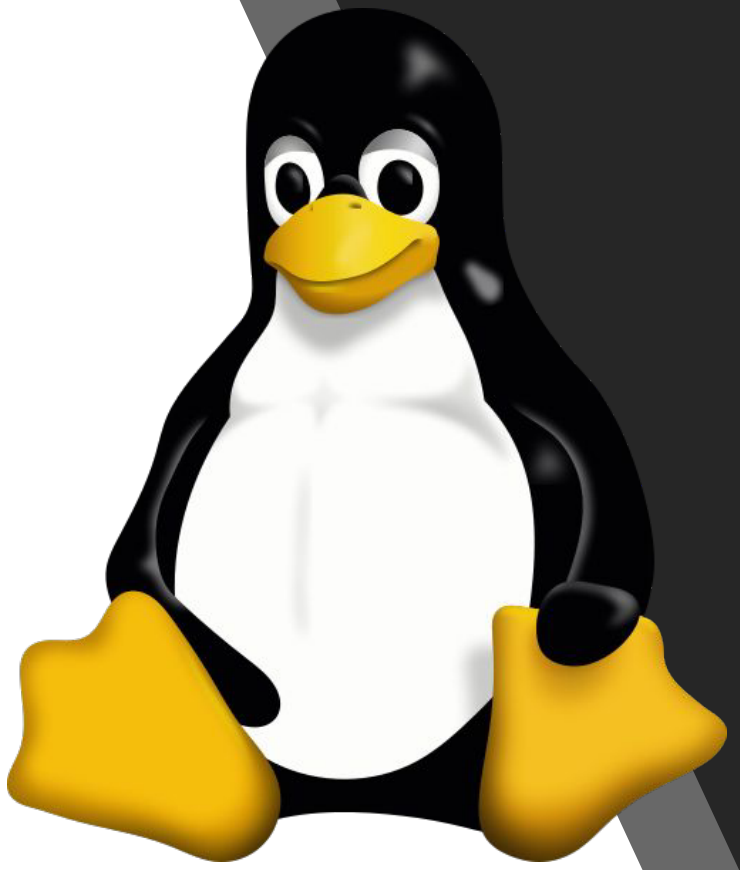
Historical context

- 1968: Cold-war ArpaNet emerges (later renamed to 'The Internet')
- The UNIX operating system is developed by Ken Thompson, Dennis Ritchie, and others at Bell Labs in 1969.
- Until the creation of UNIX, software was not sold separately, but bundled with the hardware.
- Afterwards, UNIX software was initially almost always sold and/or distributed as C-language source code, no binaries.

Understanding UNIX: A Brief Overview

- UNIX is a class of operating systems based on the original UNIX operating system by Ken Thompson, Dennis Ritchie, and others at Bell Labs in 1969.
- Known for its portability, multi-user capabilities, and support for a wide range of hardware platforms.
- Hierarchical file system, the C programming language, and the CLI.
- Widely used in servers, supercomputers, mobile devices and embedded systems, as well as in other applications where stability and reliability are important.
- Today, Linux, BSD and MacOS are the main remaining UNIX systems in use.

UNIX: A Brief History



- The UNIX operating system is developed by Ken Thompson, Dennis Ritchie, and others at Bell Labs in 1969.
- UNIX V1-V4 were released in the 1970s, introducing various new features and support for the C programming language.
- The Berkeley Software Distribution (BSD) version of UNIX is released in 1977, introducing many new features such as TCP/IP networking and improved security.
- The 1980s saw the release of commercial versions of UNIX, including UNIX System III and UNIX System V, which became the dominant version in the commercial market.
- The 1990s saw the release of various UNIX-based operating systems, such as Solaris, AIX, HP-UX, IRIX, Tru64 UNIX, Microsoft Xenix, as well as the establishment of standards and organizations such as POSIX, OSF, and The Open Group.
- The X Window System, a graphical user interface for UNIX-like operating systems, is released in 1985.
- **The first version of the Linux kernel is released in 1990, becoming a popular open-source alternative to UNIX.**
- The first version of the BSD-based operating system, macOS, is released by Apple in 2001.
- Google releases **open-source** Linux-based operating system Android in 2008.
- Many popular technologies such as the internet, cloud computing, and ML/AI rely on Linux for their infrastructure and development. MacOS is used in personal computers and mobile devices.

Understanding Open-Source Software

- A type of software whose source code is publicly available and can be modified and distributed by anyone.
- Allows for collaboration and community-driven development, as well as the ability for users to customize and improve the software to fit their specific needs.
- Often distributed under open-source licenses, which allow for free use, modification, and distribution of the software.

The Evolution of Open-Source Software (OSS)

- The concept of open-source software dates back to the early days of computing, with the first open-source software being released in the 1960s.
- Richard Stallman founded the free software movement in 1983 and the Free Software Foundation in 1985 when he announced he would develop the GNU operating system, a Unix-like operating system entirely using free software.
- Stallman pioneered the concept of *copyleft*, and is the main author of the GNU General Public License, the most widely used free software license.
- The term "open source" was coined in 1998 by a group of software developers in response to the popularity of the Internet and the need for a unified term.
- The Open Source Initiative (OSI) was established in 1998 to promote and protect open-source software.
- Today, open-source software is widely used across industries and has become an important part of the technology ecosystem and R&D.
- Examples: BSD, Linux, Apache, Firefox, Android, Kubernetes, ...

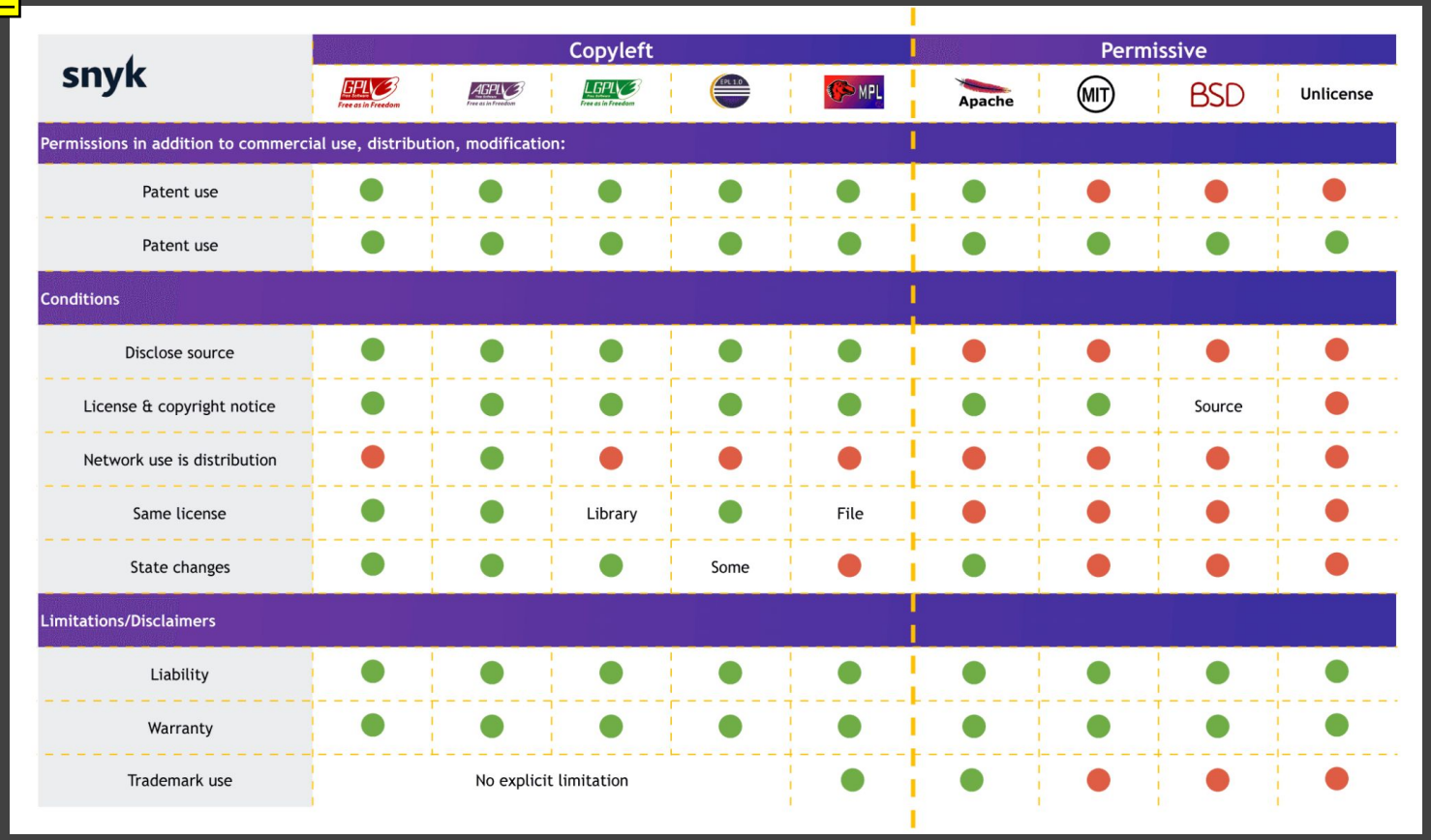
open source initiative

Software Licensing Overview

	Copyright	Copyleft	Permissive	Creative Commons
What is a user allowed to do with the code?	What creator dictates	What user wants under certain rules	What user wants with a few restrictions	What user wants without restrictions
Clause of the use	As creator dictates	Derivative work must be attributed to creator, open-source and copyleft	Derivative work must be attributed to a creator	Derivative work must be attributed to a creator
Source code	As creator dictates	Must be open	Don't have to be open	No specific terms about the distribution of source code
Is creator liable for bugs?	✓ YES	✓ YES	✗ NO	✗ NO
Re-licensing	As creator dictates	Derivative work cannot be released as proprietary software	Derivative work can be released under another licen se or as proprietary software	Derivative work can be released under another licens e or as proprietary software
Commercial restrictions	As creator dictates	Permitted	Permitted	Permitted

- **Software licenses dictate how software can be used, modified, and distributed.**
- Copyright: default type of license, exclusive rights to use, distribute, and modify the software.
- Copyleft: open-source license, requires derivative works to be released under the same license.
- Permissive: allows for free use, modification, and distribution.
- Creative Commons: not a software-specific license but widely used for creative works like images, video, and audio.

Open-Source Licensing Types: An Overview



snyk	Copyleft					Permissive			
	GPL	AGPL	LGPL		MPL	Apache	MIT	BSD	Unlicense
Permissions in addition to commercial use, distribution, modification:									
Patent use	●	●	●	●	●	●	●	●	●
Patent use	●	●	●	●	●	●	●	●	●
Conditions									
Disclose source	●	●	●	●	●	●	●	●	●
License & copyright notice	●	●	●	●	●	●	●	Source	●
Network use is distribution	●	●	●	●	●	●	●	●	●
Same license	●	●	Library	●	File	●	●	●	●
State changes	●	●	●	Some	●	●	●	●	●
Limitations/Disclaimers									
Liability	●	●	●	●	●	●	●	●	●
Warranty	●	●	●	●	●	●	●	●	●
Trademark use	No explicit limitation				●	●	●	●	●

- Open-source licenses dictate how software can be used, modified, and distributed.
- Different open-source licenses have different terms and conditions.
- Major open-source licenses include:
  - Apache License: permissive license, free use, modification, and distribution without any additional conditions, includes a patent protection provision.
  - BSD License: permissive license, free use, modification, and distribution without any additional conditions.
  - MIT License: permissive license, free use, modification, and distribution without any additional conditions
  - GNU General Public License (GPL): copyleft license, any derivative works must also be released under the GPL
  - GNU Lesser General Public License (LGPL): allows for derivative works to be released under a different license, as long as they still make use of the original software.
  - GNU Affero General Public License (AGPL): copyleft license, any derivative works must also be released under the AGPL and any use over a network must be open-sourced. (Cloud!)
- Open-source licenses can be complex, and it's essential to read and understand the terms and conditions of the specific license before using or distributing open-source software.
- There are many alternatives (80+)
  - https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses
  - http://www.wtfpl.net

Understanding the GPL: Versions and Implications

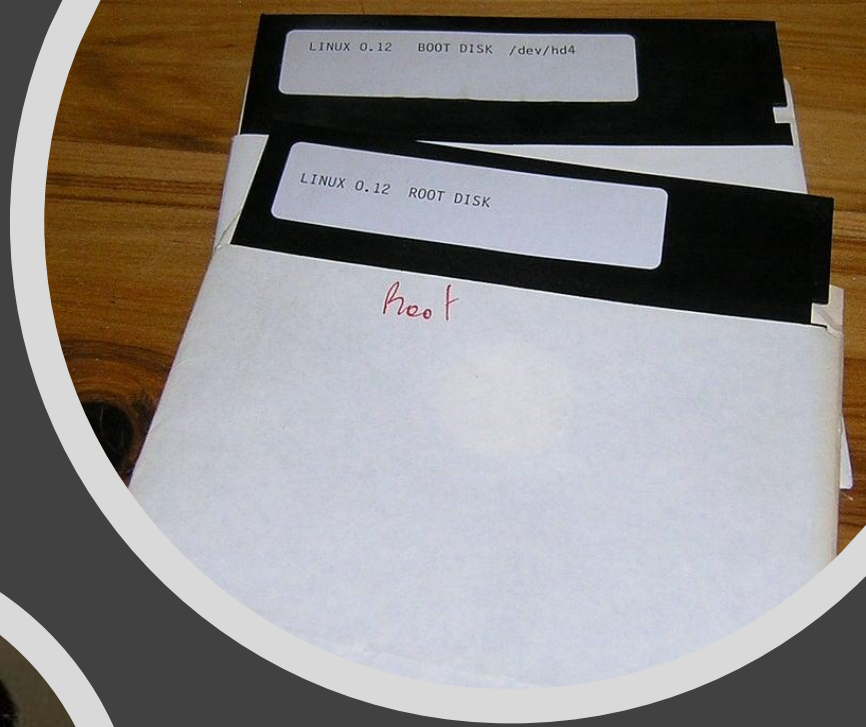
- The GNU General Public License (GPL) is a widely used open-source license
- Copyleft license: any derivative works of the original software must also be released under the GPL.
- Several revisions, most recent GPLv3 (2007)
  - GPLv1 (1989) - protect the rights of users and ensure that software remains free and open-source.
  - GPLv2 (1991) - clarified and expanded upon the terms of GPLv1, including provisions for patent protection and internationalization.
  - GPLv3 (2007) - new provisions for digital rights management, patent protection, and software patents.
- GPLv2 is considered to be more permissive than GPLv3, which is considered to be more restrictive.
  - GPLv2 software can't be upgraded to GPLv3, if a software's copyright holder releases a new version of the software under GPLv3, it will be considered a separate work and will have to be distributed and licensed separately.

Understanding GPLv3: Criticisms and Considerations

- GPLv3 (2007) released as an updated version of GPLv2.
- Some criticism and resistance from some members of the open-source community.
- GPLv3 is considered more restrictive than GPLv2
  - Digital Rights Management (DRM): GPLv2 did not explicitly restrict DRM, allowing software to enforce hardware-based restrictions. GPLv3's "anti-tivoization" clause bans such restrictions, ensuring users can run modified versions.
  - Contributors (authors who add code) grant an implicit patent license to users (those who receive and use the software).
  - Contributors cannot sue users for patent infringement based on their contributions.
  - Software using or linking to GPLv3 must also adopt GPLv3.
- GPLv3 split the open-source community, leading to incompatible software versions.
- Some developers oppose it for limiting monetization.
- The Linux kernel is licensed under the GNU General Public License version 2 (GPLv2).

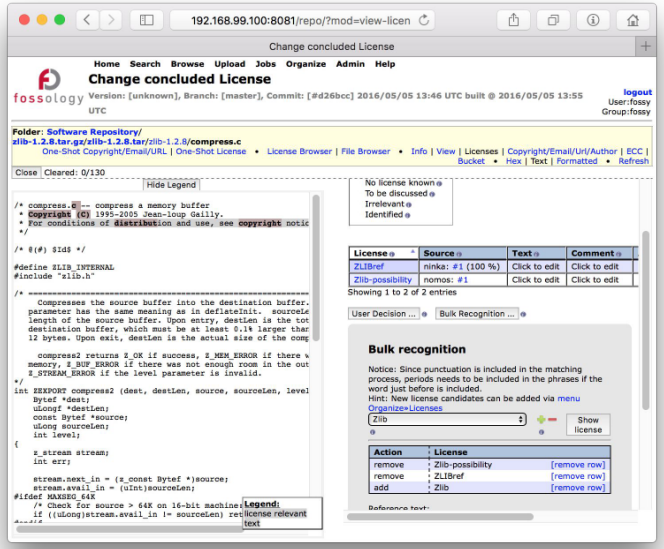
Minix and Linux

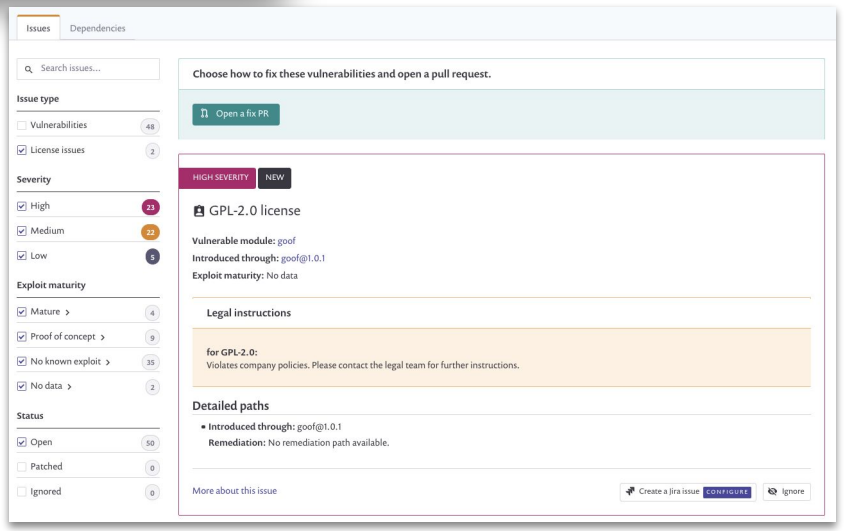
- Minix and Linux are closely related operating systems rooted in Unix.
- Minix was developed in 1987 by Andrew S. Tanenbaum as a teaching tool.
- Linux was created in 1991 by Linus Torvalds as a free, open-source operating system and was inspired by Minix.
- Both have a modular structure consisting of a kernel and user-space utilities.
- Minix was released under a restrictive license, while Linux was released under the GPL.
- Linux's open-source nature and ability to run on various hardware platforms made it very quickly more popular than Minix.



Open-Source License Compliance Tools

- Popular free, open-source tools for checking open-source licenses in a project:
  - FOSSology - a software license compliance software system and toolkit.
  - LicenseFinder - a Ruby gem for finding and reporting on licenses of dependencies.
  - ScanCode - a tool for scanning code and discovering license, copyrights, and other information.
  - Snipe-IT - asset management in IT Operations
- Cloud-based tools
  - Snyk - its Open Source License Compliance offering is a cloud-based solution that analyzes open-source licenses and vulnerabilities.
  - Mend - a cloud-based solution for managing open-source software licenses and vulnerabilities.
- These tools can help organizations ensure they are in compliance with open-source licenses and avoid potential legal issues.
- Integrated into CI/CD pipelines.



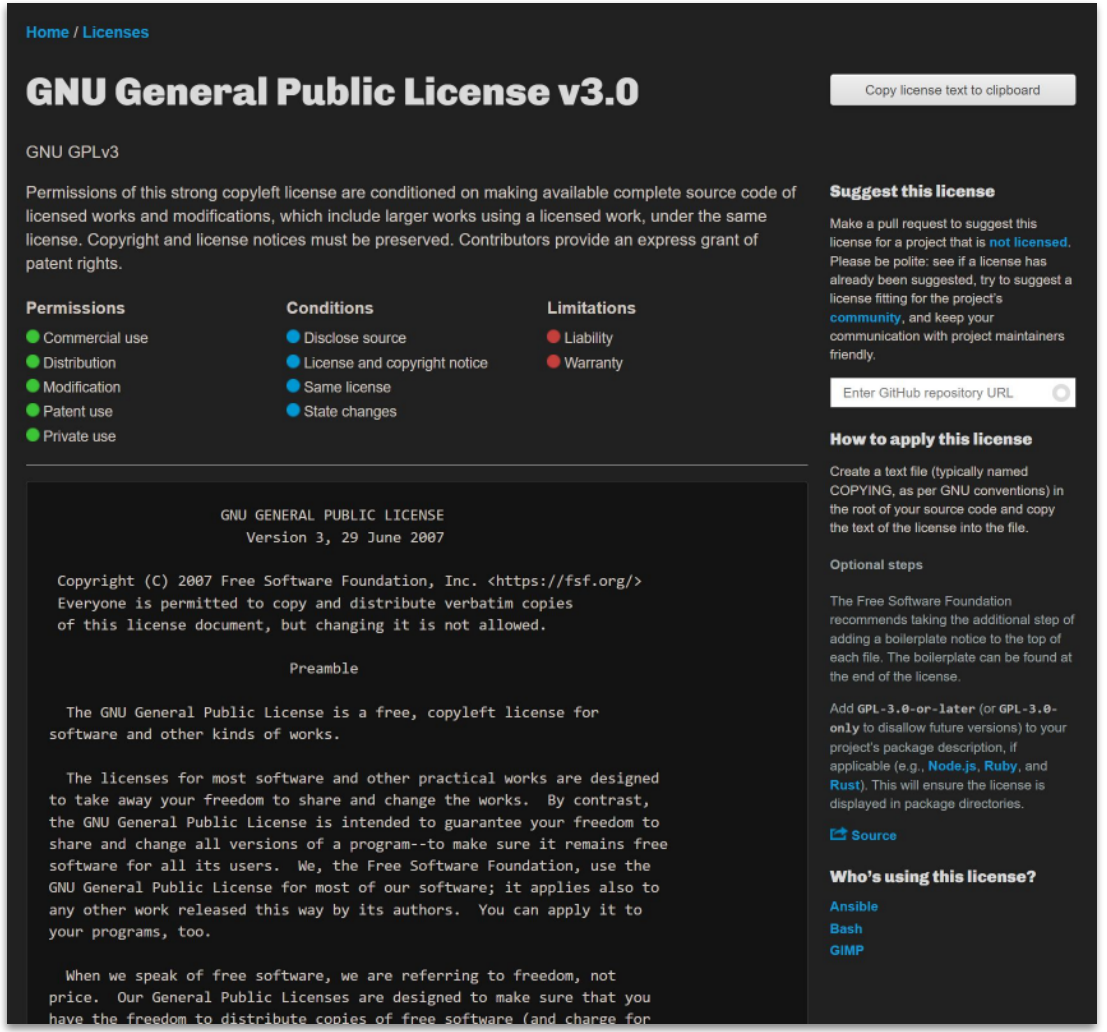


Practically - GPLv3 example

- To release software under GPL licenses, users need to obtain a copyright disclaimer from any superseding entity like an employer or school.
- Once the disclaimer is in place, **each file** should receive the proper copyright notices while clearly defining what versions users can use.
- Finally, add a **`COPYING`** file containing the complete text of the GNU GPL terms and conditions, while adding license notices and a statement of permission in each file. Adding the copyright notice display at startup is optional.

or

- use a license tool
  - https://choosealicense.com/
  - https://spdx.org/licenses/



MIT License

```
Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Notable projects that use the MIT License include the X Window System, Ruby on Rails, Nim, Node.js, Lua, and jQuery. Notable companies using the MIT License include Microsoft (.NET), Google (Angular), and Meta (React).

GPLv2 License

```
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy  name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
```

Actual license file:
https://www.gnu.org/licenses/old-licenses/gpl-2.0.txt

GPLv2 License

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

GPLv3 License

```
<one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

Actual license file:
https://www.gnu.org/licenses/gpl-3.0.txt

GPLv3 License

Actual license file:
https://www.gnu.org/licenses/gpl-3.0.txt

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program>  Copyright (C) <year>  <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

end