

Free (Libre) Software Practices and Politics

Nick Montfort
nickm.com

Professor of Digital Media, MIT

Prof. II & PI, Center for Digital Narrative, University of Bergen

These slides © Nick Montfort 2023, CC BY-SA 4.0

ELO Conference 2023, Overcoming Divides / 2023-07-14

I've provided these slides for people who attended my workshop at the ELO Conference, July 14, 2023, so they can look back over them if they like.

Anyone else is welcome to read them, and even to modify and share them, as they are licensed as CC BY-SA.

However, these slides do not contain notes and were not meant to be used as a stand-alone resource. I just prepared them to facilitate our workshop.

Workshop Outcomes

1. Be aware of the principles behind free (libre) software
2. Be able to characterize the *free software movement*
3. Identify practical free (libre) software
4. See how our incremental/further adoption can happen
5. Learn with each other about how to promote free (libre) software to individuals & our institutions

Workshop Focus

... is on software.

Specifically, software relevant to artists, authors, & scholars.

Free culture / open access overall, although related,
is not our main topic.

“Software” will touch on Web services & even generative AI.

free



free



Free software is not “no cost” or \$0 software.

**It happens, today, that once one is online, one
can get it for no marginal cost.**

**WE DEMAND
OUR RIGHTS
AND FREEDOM**



**Free software is not fundamentally
“gratis,” but “libre.”**

**It is software that ensures
the user has freedoms.**

REALIZABLE FANTASIES

The GNU Manifesto

by Richard Stallman

This month's installment of Realizable Fantasies features a guest appearance by Richard Stallman. Stallman, inventor of the original much-imitated Emacs editor and formerly of the Artificial Intelligence Lab at MIT, has worked extensively on compilers, editors, debuggers, command interpreters, the Incompatible Timesharing System (ITS) and the Lisp Machine operating system. He pioneered terminal-independent display support in ITS. Since then he has implemented one crashproof file system and two window systems for Lisp machines, and designed a third window system now being implemented; this one will be ported to many systems including GNU.—M.S.

GNU, which stands for Gnu's Not Unix, is the name for the complete Unix-compatible software system that I am writing so that I can give it away free to everyone who can use it. Many other programmers are helping me. Contributions of time, money, programs and equipment are greatly needed.

Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crash-proof file system, filename completion, perhaps, terminal-independent display support, and eventually, a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen.

Both C and Lisp will be available as system programming languages. We will try to support UUCP, MIT Chaonet, and Internet protocols for communication.

GNU is aimed initially at machines in the 68000/16000 class, with virtual memory, because they are the easiest machines to make it run on. The extra effort to make it run on less powerful machines will be left to someone who wants to use it on them.

Why I Must Write GNU

If I like a program, I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not

not too bad. The essential features of Unix seem to be good ones, and I think I can fill in what Unix lacks without spoiling them. Furthermore a system compatible with Unix would be convenient for many other people to adopt.

How GNU Will Be Available

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

Why Many Other Programmers Want to Help

I have found many other programmers who are excited about GNU and want to help. Many programmers are unhappy about the commercialization of system software. It may enable them to make more money, but it requires that they feel like competitors with other programmers rather than like comrades. The fundamental act of friendship among programmers is the



Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crash-proof file system, filename completion, perhaps, terminal-independent display support, and eventually, a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen.

Both C and Lisp will be available as system programming languages. We will try to support UUCP, MIT Chaosnet, and Internet protocols for communication.

GNU is aimed initially at machines in the 68000/16000 class, with virtual memory, because they are the easiest machines to make it run on. The extra

not too bad. The essential features of Unix seem to be good ones, and I think I can fill in what Unix lacks without spoiling them. Furthermore a system compatible with Unix would be convenient for many other people to adopt.

How GNU Will Be Available

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

Why Many Other Programmers Want to Help

I have found many other programmers

What Freedoms?

To run, copy, distribute, study,
change, and improve the software.

The Four Freedoms

0. To run the program as you wish, for any purpose.
1. To study how the program works, and change it so it does your computing as you wish. (Access to the source code is a precondition for this.)
2. To redistribute copies so you can help others.
3. To distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. (Access to the source code is a precondition for this.)

Can free (libre) software be commercial?

The Four Freedoms

0. To run the program as you wish, **for any purpose.**
1. To study how the program works, and change it so it does your computing as you wish. (Access to the source code is a precondition for this.)
2. To redistribute copies so you can help others.
3. To distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. (Access to the source code is a precondition for this.)

If commercial uses are prohibited,
the software is not free.

So free (libre) software *must* be available
for commercial, and all other, uses.

**Can export-restricted software be
free software?**

The Four Freedoms

0. To run the program as you wish, for any purpose.
1. To study how the program works, and change it so it does your computing as you wish. (Access to the source code is a precondition for this.)
2. **To redistribute copies so you can help others.**
3. **To distribute copies of your modified versions to others.** By doing this you can give the whole community a chance to benefit from your changes. (Access to the source code is a precondition for this.)

**Can something “for educational use only”
be free software?**

The Four Freedoms

- 0. To run the program as you wish, **for any purpose.**
- 1. To study how the program works, and change it so it does your computing as you wish. (Access to the source code is a precondition for this.)
- 2. To redistribute copies so you can help others.
- 3. To distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes. (Access to the source code is a precondition for this.)

Is all open source software free?

(Strictly speaking...)

0. **To run the program as you wish, for any purpose.**
1. To study how the program works, and change it so it does your computing as you wish. (Access to the source code is a precondition for this.)
2. **To redistribute copies so you can help others.**
3. **To distribute copies of your modified versions to others. By doing this you can give the whole community a chance to benefit from your changes.** (Access to the source code is a precondition for this.)

As a practical matter, most people use “open source” to mean the same thing as “free (libre) software,” but without the same emphasis on user freedoms.

If you adhere to the ideals of the free software movement, it’s best to use a term like “free (libre) software.”

FLOSS and FOSS are also used.

**The freedom(s) to run, copy, distribute, study,
change, and improve the software.**

How is “software freedom” ensured?

Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crash-proof file system, filename completion, perhaps, terminal-independent display support, and eventually, a Lisp-based window system through which several Lisp programs and ordinary Unix programs can share a screen.

Both C and Lisp will be available as system programming languages. We will try to support UUCP, MIT Chaosnet, and Internet protocols for communication.

GNU is aimed initially at machines in the 68000/16000 class, with virtual memory, because they are the easiest machines to make it run on. The extra

not too bad. The essential features of Unix seem to be good ones, and I think I can fill in what Unix lacks without spoiling them. Furthermore a system compatible with Unix would be convenient for many other people to adopt.

How GNU Will Be Available

GNU is not in the public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.

Why Many Other Programmers Want to Help

I have found many other programmers

How is this “software freedom” ensured?

**A legal mechanism:
A free software license.**

Early ones pioneered the idea of “copyleft.”

“Share and share alike.”

Free software licenses do not *need* to be copyleft, however.

They just have to give people the right to make changes and to share them.

They have to ensure users can run, copy, distribute, study, change, and improve the software.

Types of software license and similar licenses. The highlighted columns are free software.

| | Public domain & equivalents | Permissive license | Copyleft (protective license) | Noncommercial license | Proprietary license | Trade secret |
|----------------------|-----------------------------|---|--|--|---|-------------------------------|
| Description | Grants all rights | Grants use rights, including right to relicense (allows proprietization , license compatibility) | Grants use rights, forbids proprietization | Grants rights for noncommercial use only. May be combined with copyleft. | Traditional use of copyright ; no rights need to be granted | No information is made public |
| Software | PD, CC0 | MIT , Apache , MPL | GPL , AGPL | JRL , AFPL | proprietary software , no public license | private, internal software |
| Other creative works | PD, CC0 | CC BY | CC BY-SA | CC BY-NC | Copyright , no public license | unpublished |

Workshop Outcomes

1. **Be aware of the principles behind free (libre) software**
2. Be able to characterize the *free software movement*
3. Identify practical free (libre) software
4. See how our incremental/further adoption can happen
5. Learn with each other about how to promote free (libre) software to individuals & our institutions

What is the free software movement?

Most essentially: People promoting software freedom through the use, sharing, and development of free software, as they also help to share the free software philosophy.

**You can join organizations such as the
Free Software Foundation
or the Mozilla Foundation.**

**You can go to an event such as LibrePlanet
or participate in Software Freedom Day.**

**Those connect you to others in the free
software community.**

You can contribute to a large-scale free software project by working on development.

**You don't have to be a programmer —
it's useful to have clear bug reports,
and translators can help, too.**

But joining an organization, conference, or established project is not the only way.

Using free software, making your own software free (libre), and promoting practices & philosophies are really what the movement is about.

**While software freedom involves technology,
the movement is a social movement.**

**The implications of software freedom
may extend beyond everyday
computer use, too.**

Hacking Capitalism

The Free and Open Source Software Movement

Johan Söderberg

2008, reprinted 2015

“The fact that a loosely knit community of tinkerers can rival the research departments of the world’s largest corporations, epitomised in GNU/Linux versus Windows, is quite remarkable. The significance of this observation pivots on the expectation that the success of FOSS development can be exported to other walks of life. Ultimately, we are enticed by the thought that the FOSS model could become the dominant mode of organising labour in the future.”

Hacking Capitalism

The Free and Open Source Software Movement

Johan Söderberg

2008, reprinted 2015

“Evidence in support of such a scenario, however, is left wanting. The scalability of FOSS projects has no comparison elsewhere in the economy. The ability to coordinate a large number of contributors is facilitated by the fact that the whole development process of software code takes place in communication networks.”

What are worker-owned tech cooperatives?

A very general definition:

Worker-owned cooperatives are business enterprises that are owned and governed by their employees. All worker cooperatives have two common characteristics: 1) member-owners invest in and own the business together, and share the enterprise's profits, and 2) decision-making is democratic, with each member having one vote.

– community-wealth.org

Tech cooperatives are worker-owned cooperatives that operate in the information technology industry. This list focuses on tech coops that are providing software development and consulting services.

Tech cooperatives could be a solution if you feel alienated by big software corporations and startup culture or you feel alone and in search of solidarity as a freelancer.

Table of Contents

1. [Coops](#)

i. [North America](#)

ii. [South America](#)

iii. [Asia](#)

iv. [Europe](#)

v. [Oceania](#)

vi. [Worldwide](#)

<https://github.com/hng/tech-coops>

Digital colonialism: US empire and the new imperialism in the Global South

MICHAEL KWET

Abstract: This article proposes a conceptual framework of how the United States is reinventing colonialism in the Global South through the domination of digital technology. Using South Africa as a case study, it argues that US multinationals exercise imperial control at the architecture level of the digital ecosystem: software, hardware and network connectivity, which then gives rise to related forms of domination. The monopoly power of multinational corporations is used for resource extraction through rent and surveillance – *economic domination*. By controlling the digital ecosystem, Big Tech corporations control computer-mediated experiences, giving them direct power over political, economic and cultural domains of life – *imperial control*. The centrepiece of surveillance capitalism, Big Data, violates the sanctity of privacy and concentrates economic power in the hands of US corporations – a system of *global surveillance capitalism*. As a feature of surveillance capitalism, Global North intelligence agencies partner with their own corporations to conduct mass and targeted surveillance in the Global South – which intensifies *imperial state surveillance*. US elites have persuaded people that society must proceed according to its ruling class conceptions of the digital world, setting the foundation for *tech hegemony*. The author argues for a different ecosystem that decentralises technology by placing control directly into the hands of the people to counter the rapidly advancing frontier of digital empire.

Michael Kwet is Visiting Fellow at the Information Society Project, Yale Law School and a PhD Candidate in Sociology at Rhodes University, South Africa.

“Software functions as the coded logic that constrains and enables particular user experiences. ... Microsoft maintains absolute control over how [Windows] works. The same goes for other proprietary apps, like Google Play or Adobe Photoshop. By design, non-free software provides the [corporate] owner power over the user experience. It is authoritarian software.”

Digital colonialism: US empire and the new imperialism in the Global South

MICHAEL KWET

Abstract: This article proposes a conceptual framework of how the United States is reinventing colonialism in the Global South through the domination of digital technology. Using South Africa as a case study, it argues that US multinationals exercise imperial control at the architecture level of the digital ecosystem: software, hardware and network connectivity, which then gives rise to related forms of domination. The monopoly power of multinational corporations is used for resource extraction through rent and surveillance – *economic domination*. By controlling the digital ecosystem, Big Tech corporations control computer-mediated experiences, giving them direct power over political, economic and cultural domains of life – *imperial control*. The centrepiece of surveillance capitalism, Big Data, violates the sanctity of privacy and concentrates economic power in the hands of US corporations – a system of *global surveillance capitalism*. As a feature of surveillance capitalism, Global North intelligence agencies partner with their own corporations to conduct mass and targeted surveillance in the Global South – which intensifies *imperial state surveillance*. US elites have persuaded people that society must proceed according to its ruling class conceptions of the digital world, setting the foundation for *tech hegemony*. The author argues for a different ecosystem that decentralises technology by placing control directly into the hands of the people to counter the rapidly advancing frontier of digital empire.

Michael Kwet is Visiting Fellow at the Information Society Project, Yale Law School and a PhD Candidate in Sociology at Rhodes University, South Africa.

“The Free Software Movement ... developed within the centre of empire in response to enclosure of the software commons, first through proprietary software and now through Internet centralization. It has concentrated on developing forms of technology that grant control to individuals and communities for the purpose of individual and collective freedom.”

Digital colonialism: US empire and the new imperialism in the Global South

MICHAEL KWET

Abstract: This article proposes a conceptual framework of how the United States is reinventing colonialism in the Global South through the domination of digital technology. Using South Africa as a case study, it argues that US multinationals exercise imperial control at the architecture level of the digital ecosystem: software, hardware and network connectivity, which then gives rise to related forms of domination. The monopoly power of multinational corporations is used for resource extraction through rent and surveillance – *economic domination*. By controlling the digital ecosystem, Big Tech corporations control computer-mediated experiences, giving them direct power over political, economic and cultural domains of life – *imperial control*. The centrepiece of surveillance capitalism, Big Data, violates the sanctity of privacy and concentrates economic power in the hands of US corporations – a system of *global surveillance capitalism*. As a feature of surveillance capitalism, Global North intelligence agencies partner with their own corporations to conduct mass and targeted surveillance in the Global South – which intensifies *imperial state surveillance*. US elites have persuaded people that society must proceed according to its ruling class conceptions of the digital world, setting the foundation for *tech hegemony*. The author argues for a different ecosystem that decentralises technology by placing control directly into the hands of the people to counter the rapidly advancing frontier of digital empire.

Michael Kwet is Visiting Fellow at the Information Society Project, Yale Law School and a PhD Candidate in Sociology at Rhodes University, South Africa.

“During the 2000s, Free Software ... was endorsed for public sector implementation across the Global South ... The development and dispersion of the Free Software philosophy across the world resembles the development of socialism within Europe as a reaction to land enclosure and industrial exploitation, and its subsequent spread across liberation movements throughout the world.”

Digital colonialism: US empire and the new imperialism in the Global South

MICHAEL KWET

Abstract: This article proposes a conceptual framework of how the United States is reinventing colonialism in the Global South through the domination of digital technology. Using South Africa as a case study, it argues that US multinationals exercise imperial control at the architecture level of the digital ecosystem: software, hardware and network connectivity, which then gives rise to related forms of domination. The monopoly power of multinational corporations is used for resource extraction through rent and surveillance – *economic domination*. By controlling the digital ecosystem, Big Tech corporations control computer-mediated experiences, giving them direct power over political, economic and cultural domains of life – *imperial control*. The centrepiece of surveillance capitalism, Big Data, violates the sanctity of privacy and concentrates economic power in the hands of US corporations – a system of *global surveillance capitalism*. As a feature of surveillance capitalism, Global North intelligence agencies partner with their own corporations to conduct mass and targeted surveillance in the Global South – which intensifies *imperial state surveillance*. US elites have persuaded people that society must proceed according to its ruling class conceptions of the digital world, setting the foundation for *tech hegemony*. The author argues for a different ecosystem that decentralises technology by placing control directly into the hands of the people to counter the rapidly advancing frontier of digital empire.

Michael Kwet is Visiting Fellow at the Information Society Project, Yale Law School and a PhD Candidate in Sociology at Rhodes University, South Africa.

“Impressed by the anti-possessive design of Free Software, Archbishop Desmond Tutu has endorsed the FSM. Introducing Stallman at the University of Western Cape in 2007, he stated: ‘There are those who will take the fruits of the human mind and lock them up, dishing them out to us in meted amounts for a fee that locks most of our people out. And there are laws that are reserved for business reasons and changed to rob society of its own rights ... Free Software and Open Source, Free and Open Resources for Education, new ways to create and share cultural artifacts such as music, writing, and art — all of these are changing the world for the better.’”

**WE DEMAND
OUR RIGHTS
AND FREEDOM**



Workshop Outcomes

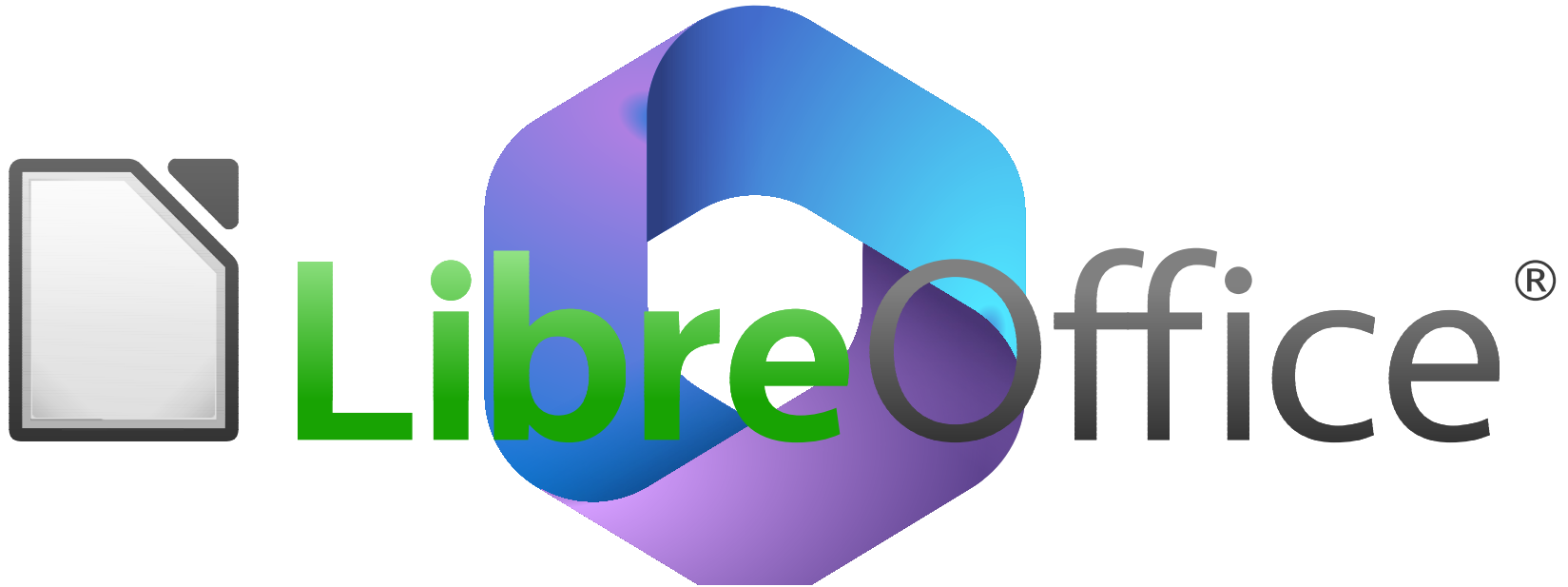
1. **Be aware of the principles behind free (libre) software**
2. **Be able to characterize the *free software movement***
3. **Identify practical free (libre) software**
4. **See how our incremental/further adoption can happen**
5. **Learn with each other about how to promote free (libre) software to individuals & our institutions**

Apache *
Audacity
BIND *
Blender
Calibre
Debian *
Drupal
Emacs
Firefox
GCC
GIMP
GNU utils
Inkscape
LibreOffice *



Linux (kernel) *
MediaWiki *
MySQL *
Nextcloud
Notepad++
Perl
PHP *
Pine (Alpine)
Processing
Python
Sendmail
Ubuntu
VLC
WordPress *

**Some free software substitutes
for propriety software.**

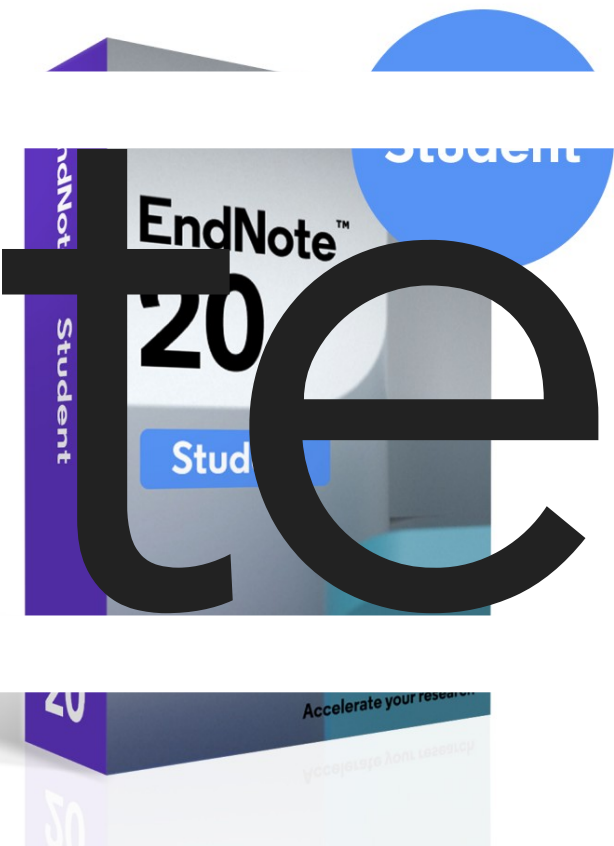


**Some free software substitutes
for propriety software.**

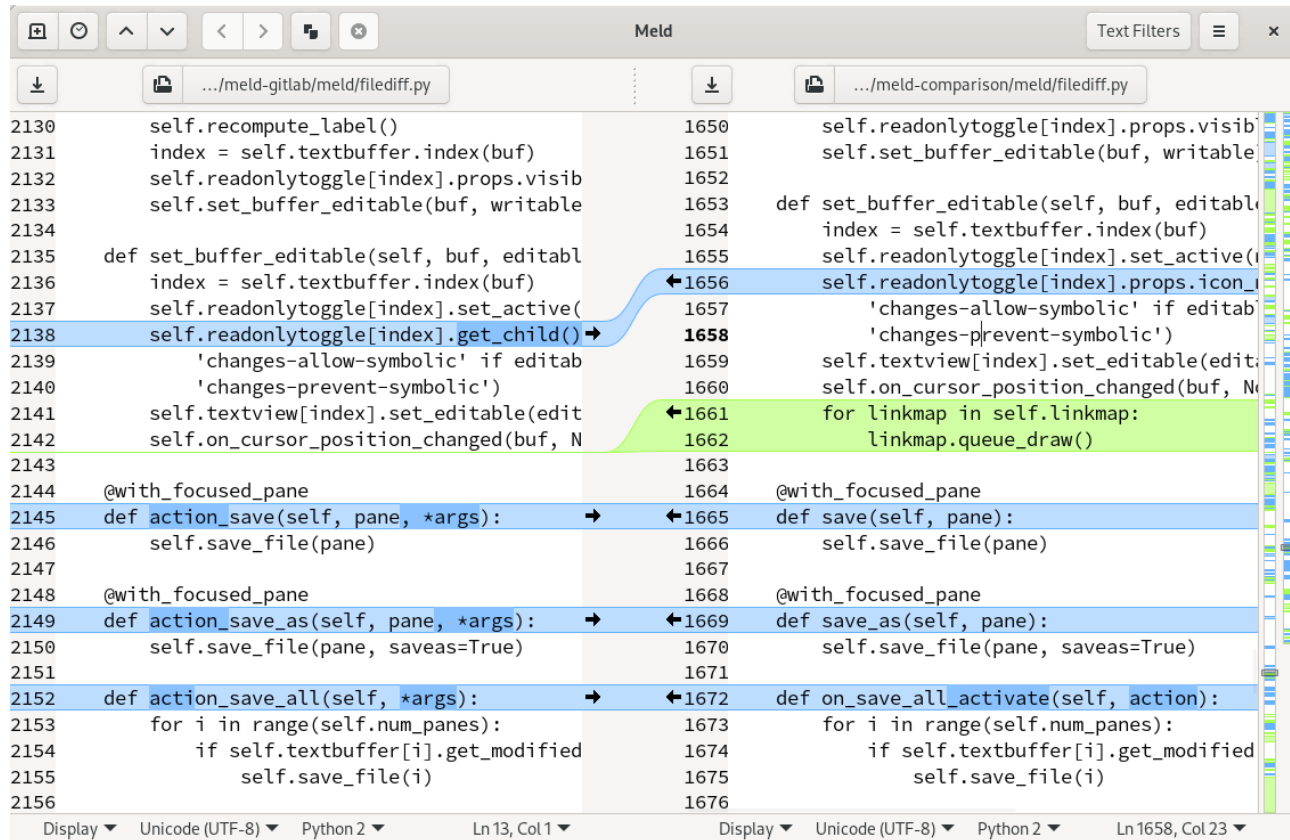


**Some free software substitutes
for propriety software.**

zotero

The background features a collage of software-related images. A central focus is a box for EndNote Student 2010, with its purple and grey design and text clearly visible. To the right of the box is a large, solid blue sphere. Below the main box, there is a reflection of the box and a smaller, partially visible box with the text 'Accelerate your research'. The word 'zotero' is superimposed over this collage, with the 'z' in red and the rest in black.

A lot of free software does things that no proprietary software does.



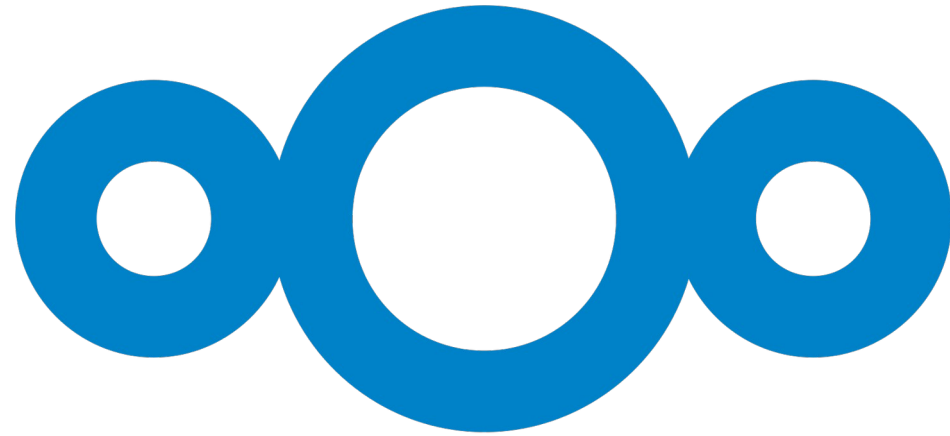
The screenshot shows the Meld application window with two files being compared: `.../meld-gitlab/meld/filediff.py` (left pane) and `.../meld-comparison/meld/filediff.py` (right pane). The left pane shows lines 2130 to 2156, and the right pane shows lines 1650 to 1676. The diff highlights several changes, with some lines highlighted in blue and others in green. A blue arrow points from line 2138 in the left pane to line 1656 in the right pane, indicating a change in the `self.readonlytoggle[index].props.icon` property. Another blue arrow points from line 2145 in the left pane to line 1665 in the right pane, indicating a change in the `def action_save` method. A green arrow points from line 2142 in the left pane to line 1661 in the right pane, indicating a change in the `for linkmap in self.linkmap:` loop. The bottom status bar shows the current line and column for both panes: `Ln 13, Col 1` and `Ln 1658, Col 23`.

```
2130 self.recompute_label()
2131 index = self.textbuffer.index(buf)
2132 self.readonlytoggle[index].props.visib
2133 self.set_buffer_editable(buf, writable
2134
2135 def set_buffer_editable(self, buf, editabl
2136     index = self.textbuffer.index(buf)
2137     self.readonlytoggle[index].set_active(
2138     self.readonlytoggle[index].get_child()
2139     'changes-allow-symbolic' if editab
2140     'changes-prevent-symbolic')
2141     self.textview[index].set_editable(edit
2142     self.on_cursor_position_changed(buf, N
2143
2144     @with_focused_pane
2145     def action_save(self, pane, *args):
2146         self.save_file(pane)
2147
2148     @with_focused_pane
2149     def action_save_as(self, pane, *args):
2150         self.save_file(pane, saveas=True)
2151
2152     def action_save_all(self, *args):
2153         for i in range(self.num_panes):
2154             if self.textbuffer[i].get_modified
2155                 self.save_file(i)
2156
1650 self.readonlytoggle[index].props.visib
1651 self.set_buffer_editable(buf, writable
1652
1653 def set_buffer_editable(self, buf, editabl
1654     index = self.textbuffer.index(buf)
1655     self.readonlytoggle[index].set_active(
1656     self.readonlytoggle[index].props.icon
1657     'changes-allow-symbolic' if editab
1658     'changes-prevent-symbolic')
1659     self.textview[index].set_editable(edit
1660     self.on_cursor_position_changed(buf, N
1661     for linkmap in self.linkmap:
1662         linkmap.queue_draw()
1663
1664     @with_focused_pane
1665     def save(self, pane):
1666         self.save_file(pane)
1667
1668     @with_focused_pane
1669     def save_as(self, pane):
1670         self.save_file(pane, saveas=True)
1671
1672     def on_save_all_activate(self, action):
1673         for i in range(self.num_panes):
1674             if self.textbuffer[i].get_modified
1675                 self.save_file(i)
1676
```

**Some free software is not a substitute,
so people hoping for one will be disappointed.**



**There is free software for cloud services.
You can run it yourself or pay to have a
company run this software (still respecting
the four freedoms) for you.**



nextcloud

**There are free software alternatives
even for cutting-edge systems.**



Apache *
Audacity
BIND *
Blender
Calibre
Debian *
Drupal
Emacs
Firefox
GCC
GIMP
GNU utils
Inkscape
LibreOffice *



Linux (kernel) *
MediaWiki *
MySQL *
Nextcloud
Notepad++
Perl
PHP *
Pine (Alpine)
Processing
Python
Sendmail
Ubuntu
VLC
WordPress *

Workshop Outcomes

1. **Be aware of the principles behind free (libre) software**
2. **Be able to characterize the *free software movement***
3. **Identify practical free (libre) software**
4. **See how our incremental/further adoption can happen**
5. **Learn with each other about how to promote free (libre) software to individuals & our institutions**

A user can move toward *more* software freedom, even if they aren't ready to purge their computer of all proprietary software.

**The best place to start is where
proprietary software enforces strong
types of domination:**

Lock-in

Restrictions on communication

Corporate surveillance & data gathering

Lock-in

Prevents users from using *any* different software, because their data is locked in a proprietary format.

You also invested your time in learning menu layouts, keyboard shortcuts, etc.

Restrictions on Communication

You cannot FaceTime with me because I have an Android phone.

You communicate using a social network where the rules may change at any point; your account may be suspended without explanation, you may be required to pay, your reading may be harshly limited.

Corporate Surveillance & Data Gathering

Your emails are all processed by a global megacorporation, which knows enough about them to suggest completions of phrases and serve you targeted advertisements.

Now, time for some workshop work!

What do we do with our computers?

(And what would we like to?)

**We'll list some opportunities to incrementally
adopt free software.**

Workshop Outcomes

1. Be aware of the principles behind free (libre) software
2. Be able to characterize the *free software movement*
3. Identify practical free (libre) software
4. See how our incremental/further adoption can happen
5. Learn with each other about how to promote free (libre) software to individuals & our institutions

**We all live in hierarchies...
patriarchies, systems of power & oppression.**

**Just as our labor allows us to eat, have
shelter, and get health care, our employer
may compel us to use proprietary software.**

**We can still make choices and find chances to
further the free software cause.**

**What are some practical ways those of us
here can think to do this?**

Can we brainstorm together now?

Workshop Outcomes

1. Be aware of the principles behind free (libre) software
2. Be able to characterize the *free software movement*
3. Identify practical free (libre) software
4. See how our incremental/further adoption can happen
5. Learn with each other about how to promote free (libre) software to individuals & our institutions

Thank you!

**And please let me know
if you join our movement!**

Nick Montfort

nickm@nickm.com

ELO Conference 2023, Overcoming Divides / 2023-07-14