Linux – The Operating System of Freedom

Zoltán Szabó, Department of Statistics, LSE (May 19, 2023)
- A bit of Linux history.
- Linux distributions.
- Installation.
- Applications.
- Ricing and phones.
My journey

Win

start

Zoltán Szabó  Linux
My journey

Win → 'Unix'

start → school

school → home

home → school

school → home

home → school

school → home.

Zoltán Szabó

Linux
My journey

Win → 'Unix' → Fedora

school

start

home

Zoltán Szabó

Linux
My journey

Win → 'Unix' → Fedora → Ubuntu

Zoltán Szabó
My journey

Win → 'Unix' → Fedora → Ubuntu, macOS

Zoltán Szabó Linux
My journey

Win → 'Unix' → Fedora → Ubuntu, macOS → CentOS
My journey

Win → 'Unix' → Fedora → Ubuntu, macOS → CentOS → Arch.
Some fun

From: Linus Benedict Torvalds  
Date: Aug 25, **1991**, 10:57:08 PM  
Newsgroups: comp.os.mimix  
Body:  
Hello everybody ...  

I’m doing a (free) operating system (just a hobby, won’t be big...
From: Linus Benedict Torvalds
Date: Aug 25, 1991, 10:57:08 PM
Newsgroups: comp.os.mimix
Body:
Hello everybody ...

I’m doing a (free) operating system (just a hobby, won’t be big...
From: Linus Benedict Torvalds  
Date: Aug 25, 1991, 10:57:08 PM  
Newsgroups: comp.os.mimix  
Body:
Hello everybody ...

I’m doing a (free) operating system (just a hobby, won’t be big...

Linus Torvalds (∼now):
Linux today

- Supercomputer world: 100% market share,
- Runs: from old laptops to top 500 supercomputers,
Linux today

- Supercomputer world: 100% market share,
- Runs: from old laptops to top 500 supercomputers,
- At the heart of > 3 billion Android devices,
Linux today

- Supercomputer world: 100% market share,
- Runs: from old laptops to top 500 supercomputers,
- At the heart of $>3$ billion Android devices,
- Web-facing servers, Microsoft’s own Azure cloud, IoT devices, international space stations, . . .
Linux today

- Supercomputer world: 100% market share,
- Runs: from old laptops to top 500 supercomputers,
- At the heart of > 3 billion Android devices,
- Web-facing servers, Microsoft’s own Azure cloud, IoT devices, international space stations, ...

One of the main secrets

free and open source ⇒ knowledge sharing ⇒ creativity can kick in ⇒ versatility!
In fact, Linux = GNU/Linux: Linus used the GNU development tools for his kernel, . . .

- Late 1970s: companies started to spread proprietary software ⇒

- GNU project = GNU is Not Unix:
  - Goal: write a UNIX-like operating system entirely of free software.
  - Users are legally free (GPL)
    0 to use,
    1 to study,
    2 to modify, and
    3 to distribute the software.
In fact, Linux = GNU/Linux: Linus used the GNU development tools for his kernel, ... 

- Late 1970s: companies started to spread proprietary software ⇒

- GNU project = GNU is Not Unix:
  - Goal: write a UNIX-like operating system entirely of free software.
  - Users are legally free (GPL)
    0 to use,
    1 to study,
    2 to modify, and
    3 to distribute the software.

user freedom (to make choices; free ≠ gratis)
In fact, Linux = GNU/Linux: Linus used the GNU development tools for his kernel, ... 

- Late 1970s: companies started to spread proprietary software ⇒

GNU project = GNU is Not Unix:
  - Goal: write a UNIX-like operating system entirely of free software.
  - Users are legally free (GPL)

0 to use,
1 to study,
2 to modify, and
3 to distribute the software.

user freedom (to make choices; free ≠ gratis)

- give computer users freedom and control in their use of their computers.
In fact, Linux = GNU/Linux: Linus used the GNU development tools for his kernel, . . .

- Late 1970s: companies started to spread proprietary software ⇒

- GNU project = GNU is Not Unix:
  - Goal: write a UNIX-like operating system entirely of free software.
  - Users are legally free (GPL)
    0 to use,
    1 to study,
    2 to modify, and
    3 to distribute the software.

user freedom (to make choices; free ≠ gratis)

- give computer users freedom and control in their use of their computers.
- mass collaboration announced by Richard Stallman (’83; his website).
Richard (Matthew) Stallman: RMS – his hacker name;

- Founder of the Free Software Foundation ('85) → resources,
- non-profit organization to support the free software movement.
Richard (Matthew) Stallman: RMS – his hacker name;

- Founder of the Free Software Foundation ('85) → resources, non-profit organization to support the free software movement.
- Author of the GNU GPL license ('89): for the GNU project, 1st copyleft license
  any derivative work must be distributed under the same terms.
Richard (Matthew) Stallman: RMS – his hacker name;

Founder of the **Free Software Foundation** ('85) → resources,
- non-profit organization to support the free software movement.

Author of the **GNU GPL license** ('89): for the GNU project,
- 1st copyleft license
- any derivative work must be distributed under the same terms.

Creator of **GNU Emacs**: 'text editor' (LISP interpreter).
GNU/Linux (’92-), shortly **Linux**

1. GNU utils: high-level utilities.
2. **Kernel:**
   - low-level ’stuff’, written in C, GPLv2,
   - manages the CPU, memory, device drivers, file system, . . .
Tux (the mascot of the Linux kernel) := Torvalds UniX.
- Tux (the mascot of the Linux kernel) := Torvalds UniX.

- Author of Git (GPLv2):
  - distributed version control system,
  - gold standard in collaborative coding efforts,
  - developed for the Linux kernel (’05),
  - # of lines in the code of Linux kernel: 30+ million.
Free vs open; O := Odysee

Free software (a.k.a. FOSS, libre software):

- goal: to respect user freedom and privacy.

\textit{to not constrain the user}
Free vs open; $O := \text{Odysee}$

Free software (a.k.a. FOSS, libre software):
- goal: to respect user freedom and privacy.

- *not constrain* the user

- free $\in$ open-source, but
  - free $\neq$ open-source: text, vid$[O]$, open source code can 'spy' on you.

Zoltán Szabó  Linux
Free vs right to repair

- idea in 60s: $12 \ll $1500 (MacBook Pro),
- my experience: battery replacement in Surface Pro = 600€,
Free vs right to repair

- idea in 60s: $12 \ll $1500 (MacBook Pro),
- my experience: battery replacement in Surface Pro = 600€,
- I like to invest in our free future:
  - System76: repairable laptops, Launch keyboard, Pop!_OS,
Free vs right to repair

- idea in 60s\textsubscript{vid}: $12 \ll $1500 (MacBook Pro),
- my experience: battery replacement in Surface Pro = 600€,
- I like to invest in our free future:
  - System76: repairable laptops, Launch keyboard, Pop!_OS,
  - a laptop initiative: frame.work.
Linux: free ⇒

- community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.
community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.

specifically:
- no force to
  1. upgrade to the latest hardware,
  2. throw money out of the window (e.g., Win 11 Pro: £219.99),
  3. create accounts or watch dummy ads on the UI.
community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.

specifically:

- no force to
  1. upgrade to the latest hardware,
  2. throw money out of the window (e.g., Win 11 Pro: £219.99),
  3. create accounts or watch dummy ads on the UI.

- significant storage reduction ⇔ code sharing.
community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.

specifically:
- no force to
  1. upgrade to the latest hardware,
  2. throw money out of the window (e.g., Win 11 Pro: £219.99),
  3. create accounts or watch dummy ads on the UI.
- significant storage reduction ⇐ code sharing.
- no intentional slow down of your system (e.g., iPhone).
Linux: free ⇒

- community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.

- specifically:
  - no force to
    1. upgrade to the latest hardware,
    2. throw money out of the window (e.g., Win 11 Pro: £219.99),
    3. create accounts or watch dummy ads on the UI.
  - significant storage reduction ⇐ code sharing.
  - no intentional slow down of your system (e.g., iPhone).
  - standard for computing clusters (example: slurm).
Various distros (tree):

- there have been >1000 distributions,
- currently (May 23, 2023): 274 distributions
Versatility ⇒

Various distros (tree):
- there have been $>1000$ distributions,
- currently (May 23, 2023): 274 distributions,
- but minor differences.

Primary choice to make

point release vs rolling release.
Point release model: Win

Example:

- Windows: 3.0, 3.1x, 95, 98, Me, NT, 2000, XP, Vista, 7, 8, 10, 11.
Example:

- **Windows**: 3.0, 3.1x, 95, 98, Me, NT, 2000, XP, Vista, 7, 8, 10, 11.
- **macOS**: Cheetah, Puma, Jaguar, Panther, Tiger, Leopard, Snow Leopard, Lion, Mountain Lion, Mavericks, Yosemite, El Capitan, Sierra, High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura.
Point release model: Win, macOS

Example:
- **Windows**: 3.0, 3.1x, 95, 98, Me, NT, 2000, XP, Vista, 7, 8, 10, 11.
- **macOS**: Cheetah, Puma, Jaguar, Panther, Tiger, Leopard, Snow Leopard, Lion, Mountain Lion, Mavericks, Yosemite, El Capitan, Sierra, High Sierra, Mojave, Catalina, Big Sur, Monterey, Ventura.

Properties:
1. occasional big changes,
2. end-of-life date!
Point release: Linux distributions

(Debian → ) Ubuntu → Pop!_OS; Fedora.
Point release: Linux distributions

(Debian →) Ubuntu → Pop!_OS; Fedora.

- **Ubuntu:**
  - developer: Canonical Ltd.,
  - released every six months, LTS every 2 years.
Point release: Linux distributions

(Debian → ) Ubuntu → Pop!_OS; Fedora.

- Ubuntu:
  - developer: Canonical Ltd.,
  - released every six months, LTS every 2 years.

- Pop!_OS:
  - ≈ Ubuntu with customized Gnome (COSMIC),
  - maintained by System76.
Point release: Linux distributions

(Debian → ) Ubuntu → Pop!_OS; Fedora.

- **Ubuntu:**
  - developer: Canonical Ltd.,
  - released every six months, LTS every 2 years.

- **Pop!_OS:**
  - ≈ Ubuntu with customized Gnome (COSMIC),
  - maintained by System76.

- **Fedora:**
  - upstream source for Red Hat Enterprise Linux (developed by Red Hat),
  - released every 6 months.
Rolling release: Linux distributions

- **Arch Linux:**
  - one-time installation with continuous upgrades,
  - lightweight and flexible,
  - follows the keep it simple (KISS) principle,
  - designed to teach its user.
Rolling release: Linux distributions

- other examples: openSUSE Tumbleweed, Gentoo.
## Point release vs rolling release

<table>
<thead>
<tr>
<th></th>
<th>Point</th>
<th>Rolling</th>
</tr>
</thead>
<tbody>
<tr>
<td>always up-to-date</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>(new software features, bug fixes, security patches)</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>supports even very new hardware</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>more secure</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>no need to reinstall it</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>requires semi-decent internet</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>less suited for servers (where stability is max-ed)</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
My choice: Arch (released in 2002)

0. rolling release.
1. great package manager (pacman),
   - fast,
   - allows parallel downloading.

[Package managers handle dependencies.]
My choice: Arch (released in 2002)

0 rolling release.
1 great package manager (pacman),
   - fast,
   - allows parallel downloading.
   
   [Package managers handle dependencies.]
2 superb documentation (Arch Wiki):
   - base reference even for other distributions;)
My choice: Arch (released in 2002)

0. rolling release.

1. great package manager (pacman),
   - fast,
   - allows parallel downloading.

   [Package managers handle dependencies.]

2. superb documentation (Arch Wiki):
   - base reference even for other distributions;)

3. excellent software availability:
   - main: 14K
My choice: **Arch** (released in 2002)

0. rolling release.
1. great package manager (pacman),
   - fast,
   - allows parallel downloading.
   [Package managers handle dependencies.]
2. superb documentation (**Arch Wiki**):
   - base reference even for other distributions;
   - readable and searchable offline.
3. excellent software availability:
   - main: **14K**, **AUR**: **90K** packages.

Both are searchable.
1. Download, check, **burn the installation .iso to a USB stick**, leave the stick in your machine, reboot.
Installation

1. Download, check, **burn the installation .iso to a USB stick**, leave the stick in your machine, reboot.

2. Enter into the ’BIOS’ (by pressing Esc, F1/F2/...):
   - disable Secure boot,
   - choose the boot medium to be the USB stick.
Installation

1. Download, check, **burn the installation** .iso to a **USB stick**, leave the stick in your machine, reboot.

2. Enter into the ‘BIOS’ (by pressing Esc, F1/F2/…):
   - disable Secure boot,
   - choose the boot medium to be the USB stick.

3. Boot from the USB stick.
1. Download, check, burn the installation .iso to a USB stick, leave the stick in your machine, reboot.

2. Enter into the 'BIOS' (by pressing Esc, F1/F2/...):
   - disable Secure boot,
   - choose the boot medium to be the USB stick.

3. Boot from the USB stick.

4. Follow the instructions.
iso size:

- 11 GB (MacOS Ventura) – for comparison.
Downloading note

- .iso size:

- 5.2 GB (Windows 11) – for comparison.
- 11 GB (MacOS Ventura) – for comparison.
iso size:

- 4.6 GB (Ubuntu),
- 5.2 GB (Windows 11) – for comparison.
- 11 GB (MacOS Ventura) – for comparison.
iso size:

- 2.5 GB (Pop!_OS),
- 4.6 GB (Ubuntu),
- 5.2 GB (Windows 11) – for comparison.
- 11 GB (MacOS Ventura) – for comparison.
Download note

- .iso size:
  - 2 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 4.6 GB (Ubuntu),
  - 5.2 GB (Windows 11) – for comparison.
  - 11 GB (MacOS Ventura) – for comparison.
Downloading note

- `.iso` size:
  - 810 MB (Arch),
  - 2 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 4.6 GB (Ubuntu),
  - 5.2 GB (Windows 11) – for comparison.
  - 11 GB (MacOS Ventura) – for comparison.
Downloading note

- .iso size:
  - 810 MB (Arch),
  - 2 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 4.6 GB (Ubuntu),
  - 5.2 GB (Windows 11) – for comparison.
  - 11 GB (MacOS Ventura) – for comparison.

- downloading:
  - `http://torrent`: this can be faster (⇐ sharing).
Notes on the boot process — a 'bit’ technical

1. system’s firmware (such as BIOS/UEFI/Coreboot/Libreboot) starts
2. bootloader (such as GRUB \( \Leftrightarrow \) GNU; features & others) loads
3. the kernel (your operating system).

In practice:
Notes on the boot process — a ’bit’ technical

1. system’s **firmware** (such as BIOS/UEFI/Coreboot/Libreboot) starts

2. bootloader (such as GRUB ⇐ GNU; features & others) loads

3. the **kernel** (your operating system).

In practice:

- **firmware**:
  - probes for hardware, simple health checks,
  - it has a UI accessible with a magic key (Esc, F1/F2/…),
  - allows you to designate a boot device (USB/hard/CD/DVD drive, …),
  - consults the GPT\(^\dagger\) partition table to identify the ESP\(^\ddagger\), and launches the target application (typically the bootloader).

\(^\dagger\)no chat 😊, \(^\ddagger\)EFI System Partition.
Notes on the boot process — a ‘bit’ technical

1. system’s firmware (such as BIOS/UEFI/Coreboot/Libreboot) starts
2. bootloader (such as GRUB ⇐ GNU; features & others) loads
3. the kernel (your operating system).

In practice:
- **bootloader:**
  - gives a menu on which kernel / operating system to invoke.
Instructions: for Arch – scary;

1. **Step-by-step text guide** (official one).

2. **Video guide:**
   - \( \text{vid}_1 \): UEFI; check the YouTube comments as well!
   - \( \text{vid}_2 \): BIOS, UEFI, UEFI-LVM-LUKS.

Definitions:
- **firmware**: BIOS (very old machine), UEFI (semi-new computer).
- **partition table**: BIOS \( \Rightarrow \) MBR (a.k.a. DOS, MS-DOS); UEFI \( \Rightarrow \) GPT.
- **LVM**: adjustable layout, **LUKS**: encryption.
- **LUKS**: your data can't be read even if your laptop is stolen.
Instructions: for Arch – scary ;)
Instructions: BIOS or UEFI

**BIOS:**

![BIOS Image]

**UEFI:**

![UEFI Image]
Installation hints

1. use ethernet: faster.
Installation hints

1. use ethernet: faster.

2. start simply: no encryption, no LVM.
Installation hints

2 start simply – elaborated (nerdness level dependent):
   1 Live media/USB/image (Fedora, Ubuntu):
      - .iso writing, hardware support check, quick look at the system ✓,
      - slower than SSD.
   2 graphical installer (Fedora, Ubuntu).
   3 command line installer (Arch):
      - UEFI, no LVM, no LUKS: basic understanding,
      - UEFI, LVM, LUKS: slightly deeper understanding.

† Start with a DE before a WM.
Installation hints

2 start simply – elaborated (nerdness level dependent):

1. Live media/USB/image (Fedora, Ubuntu):
   - .iso writing, hardware support check, quick look at the system ✔,
   - slower than SSD.

2. graphical installer (Fedora, Ubuntu).
Installation hints

2 start simply – elaborated \(\text{(nerdness level dependent)}\):

1. Live media/USB/image (Fedora, Ubuntu):
   - .iso writing, hardware support check, quick look at the system ✓,
   - slower than SSD.

2. graphical installer (Fedora, Ubuntu).

3. command line installer (Arch):
   - UEFI, no LVM, no LUKS: basic understanding,
Installation hints

2 start simply – elaborated (nerdness level dependent):

1. **Live media/USB/image (Fedora, Ubuntu):**
   - .iso writing, hardware support check, quick look at the system ✓,
   - slower than SSD.

2. **graphical installer (Fedora, Ubuntu).**

3. **command line installer (Arch):**
   - UEFI, no LVM, no LUKS: basic understanding,
   - UEFI, LVM, LUKS: slightly deeper understanding.
Installation hints

2 start simply – elaborated† (nerdness level dependent):

1. Live media/USB/image (Fedora, Ubuntu):
   - .iso writing, hardware support check, quick look at the system ✓,
   - slower than SSD.

2. graphical installer (Fedora, Ubuntu).

3. command line installer (Arch):
   - UEFI, no LVM, no LUKS: basic understanding,
   - UEFI, LVM, LUKS: slightly deeper understanding.

†Start with a DE before a WM.
Installation hints – continued

- **partition formatting:**
  - **ext4:** more settled – my choice,
  - ![Ext4 File System](Ext4.png)
  - **btrfs:**
    - modern alternative,
    - supports compression ⇒ less space, increased storage lifespan,
    - copy-on-write ⇒ consistency even in case of power loss,
    - snapshot feature,
    - limited LUKS support.
  ⇒ It is worth keeping an eye on it!
kernel (stable), LTS kernel (longterm) [others]:
- stable: maintained until the next stable release,
- LTS: maintained for a few extra years,
- good to have both: flexibility.
4 kernel (stable), LTS kernel (longterm) [others]:
   - stable: maintained until the next stable release,
   - LTS: maintained for a few extra years,
   - good to have both: flexibility.

5 swap:
   - helps if RAM is exhausted (but slower, $\times 1000!$); size recommendations.
   - 2 types:
     1. swap partition: often preferred,
     2. swap file: easier to resize, but less tested.
Installation hints – continued

6 good boot time (∼ 11s):
   • SSD matters: Samsung 970 EVO Plus ← my choice (for laptop).
Installation hints – continued

6. good boot time (\(\sim 11\)s):
   - SSD matters: Samsung 970 EVO Plus \(\leftarrow\) my choice (for laptop).

7. Use a spare drive (to avoid the wrestling of the op. systems).
8. Create a normal user (beyond the root; ∈ wheel; sudo).

9. Log your installation, usage, information sources (e.g. by Vimwiki).

10. Post-installation: think in terms of tasks not software, and use the native applications. A weekly system update can be healthy.
Create a normal user (beyond the root; $\in$ wheel; sudo).

Log your installation, usage, information sources (e.g. by Vimwiki)!
Create a normal user (beyond the root; ∈ wheel; sudo).

Log your installation, usage, information sources (e.g. by Vimwiki)!

Post-installation:
- think in terms of tasks not software, and use the native applications.
- a weekly system update can be healthy.
Applications: categorized; some handy ones

Notations: \( M = ' \in \text{main}' \), \( A = ' \in \text{AUR}' \), \( W = \text{web client} \), \( P = \text{pip} \), ✓ = installed by default.

- **Web & mail:**
  - browser: firefox (M), tor-browser (A).
  - e-mail: ProtonMail (W), thunderbird (M).

---

Zoltán Szabó  |  Linux
Applications: categorized; some handy ones

Notations: \( M = ' \in \text{main}' \), \( A = ' \in \text{AUR}' \), \( W = \text{web client} \), \( P = \text{pip} \), \( \checkmark = \text{installed by default} \).

- Web & mail:
  - browser: firefox (\( M \)), tor-browser (\( A \)).
  - e-mail: ProtonMail (\( W \)), thunderbird (\( M \)).

- Media:
  - image viewer: feh (\( M \)), gthumb (\( M \)), geeqie (\( M \)).
  - image editor: gimp (\( M \)).
  - video player: mplayer (\( M \)), vlc (\( M \)), celluloid (\( M \)).
  - video editor: kdenlive (\( M \)).
  - video downloader: youtube-dl (\( M \)), yt-dlp (\( M \)).
  - spotify:
    - player: spotify (\( A \)),
    - downloader: spotdl (\( P \)).
  - audio editor: tenacity (\( A \)).
Text:

- document viewer: xdvi (✓, ∈ texlive), xpdf (M),
- .pdf annotation: xournalpp (M),
- text editing: texlive-most (LaTeX, group, M), kile (M) (≈ WinEdt), vim (M), libreoffice-still (M), notepadqq (M).
Applications

- **Text:**
  - document viewer: xdvi (✓, ∈ texlive), xpdf (M),
  - .pdf annotation: xournalpp (M),
  - text editing: texlive-most (LaTeX, group, M), kile (M) (≈ WinEdt), vim (M), libreoffice-still (M), notepadqq (M).

- **Chat & collaboration:**
  - chat: skypeforlinux-stable-bin (A), zoom (A), qtox (M),
  - version control: git (M),
  - calendar & reminder: remind (M).
Applications

- eye protection: **redshift** (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
Applications+

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
Applications+

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimal dock: plank (M),
eye protection: redshift (M),
file manager: thunar (M),
launcher: dmenu (M), rofi (M),
programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
pwd manager: keepassxc (M),
RSS/Atom feed reader: newsboat (M),
screen locker: slock (M),
minimal dock: plank (M),
remote support: teamviewer (A),
Applications+

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimal dock: plank (M),
- remote support: teamviewer (A),
- terminal: tilix (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimal dock: plank (M),
- remote support: teamviewer (A),
- terminal: tilix (M),
- firewall: ufw (M),
Applications

- eye protection: redshift (M),
- file manager: thunar (M),
- launcher: dmenu (M), rofi (M),
- programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
- pwd manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimal dock: plank (M),
- remote support: teamviewer (A),
- terminal: tilix (M),
- firewall: ufw (M),
- desktop environment: gnome (M, group), qtile (M, WM).
Desktop environments (DE)

- Desktop environments:
  - **window manager**, and
  - a bundle of applications (calendar, image viewer, file manager, ...).
Desktop environments (DE)

Desktop environments:
- windows manager, and
- a bundle of applications (calendar, image viewer, file manager, ...).

You
- are free to choose it, and can have multiple ones,
- get workspaces.
Desktop environments (DE)

- Desktop environments:
  - *windows manager*, and
  - a bundle of applications (calendar, image viewer, file manager, ...).
- You
  - are free to choose it, and can have multiple ones,
  - get workspaces.
- Some popular choices: GNOME, KDE Plasma, Xfce, Enlightenment.
Desktop environments (DE)

- Desktop environments:
  - windows manager, and
  - a bundle of applications (calendar, image viewer, file manager, ...).

- You
  - are free to choose it, and can have multiple ones,
  - get workspaces.

- Some popular choices: GNOME, KDE Plasma, Xfce, Enlightenment.

Examples follow
DE: Enlightenment
Extra DE inspiration

unixporn (screenshots).
Window managers (WM)

- It allows handling windows (open, close, min/max-ze, move, resize, . . .).
- It can be part of a DE or standalone.
- Idea: WMs can be even snappier than DEs.
Window managers (WM)

- It allows handling windows (open, close, min/max-ze, move, resize, ...).
- It can be part of a DE or standalone.
- Idea: WMs can be even snappier than DEs.
- 3 types:
  1. stacking (a.k.a. floating),
  2. tiling: non-overlapping windows,
  3. dynamic: allows switching between tiling and floating layout.
Stacking:

- Mutter → GNOME,
- KWin → KDE,
- Xfwm → Xfce,
- Enlightenment → Enlightenment.
Stacking:
- Mutter → GNOME,
- KWin → KDE,
- Xfwm → Xfce,
- Enlightenment → Enlightenment.

Tiling:
- i3.
Stacking:
- Mutter → GNOME,
- KWin → KDE,
- Xfwm → Xfce,
- Enlightenment → Enlightenment.

Tiling:
- i3.

Dynamic:
- Qtile:
  - it uses Python;
  - various (69) widgets.
Stacking:
- Mutter $\rightarrow$ GNOME,
- KWin $\rightarrow$ KDE,
- Xfwm $\rightarrow$ Xfce,
- Enlightenment $\rightarrow$ Enlightenment.

Tiling:
- i3.

Dynamic:
- Qtile:
  - it uses Python;)
  - various (69) widgets.

Examples follow
WM: i3

```c
/* Set up I3 specific atomos like I3_SOCKET_PATH and I3_CONFIG_PATH */
x_set_atom(name());

struct ev_io xcb_watcher = xcb_socket(sizeof(struct ev_io));
struct ev_io xcb_xkb = xcb_socket(sizeof(struct ev_io));
struct ev_check xcb_check = xcb_socket(sizeof(struct ev_check));
struct ev_prepare xcb_prepare = xcb_socket(sizeof(struct ev_prepare));

ev_io_init(xcb_watcher, xcb_get_event, xcb_get_file_dequeuer(conn), EV_READ);
ev_io_start(main_loop, xcb_watcher);

if (xcb_supported) {
  xcb_check_init(xcb_check, xcb_check_xkb);
  xcb_check_start(main_loop, xcb_check);

  /* Flush the buffer so that libev can properly get new events */
  xcb_flush();
}

ev_prepare_init(xcb_prepare, xcb_prepare_xkb);
ev_prepare_start(main_loop, xcb_prepare);

ev_flush(conn);

/* Remove existing windows (root); */
if (disable_signalhandler)
  use_signal_handler();

/* Ignore SIGPIPE to survive errors when an IPC client disconnects */
/* while we are handling a message */
if (signal(SIGPIPE, SIG_IGN))
  use_signal_handler();

/* Autotesting exec-lines */
if (autostart) {
  TAILF_REACH(exec, autostarts, autostarts) {
    LOG("auto-starting %s", exec->command);
    start_application(exec->command);
  }
}

/* Autotesting exec-always Lines */
if (autostart_always) {
  TAILF_REACH(exec_always, autostarts_always, autostarts_always) {
    LOG("auto-starting (always) %s", exec_always->command);
    start_application(exec_always->command);
  }
}
```
WM: Qtile
Login/display manager

- It gives graphical login – if you prefer not using/starting from tty:)
- Popular choices: (i) SDDM: tutorial[O] (further inspiration)
Login/display manager

- It gives graphical login – if you prefer not using/starting from tty:)
- Popular choices: (i) SDDM: tutorial[O] (further inspiration), (ii) Ly: tutorial[O]
Login/display manager

- It gives graphical login – if you prefer not using/starting from tty:)
Terminal: used for instance @ Arch install

Command shell:

- like Jupyter notebook,
- interaction with the operating system,

```
mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd /etc ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ 
```
Terminal: used for instance @ Arch install

Command shell:
- like Jupyter notebook,
- interaction with the operating system,

Google Colab: `!shellcommand`,
Terminal: used for instance @ Arch install

Command shell:
- like Jupyter notebook,
- interaction with the operating system,

Google Colab: `!shellcommand`,

**virtual console** = text terminal + login prompt (tty\(X = \text{Ctrl}+\text{Alt}+\text{FX}, \ X \in [7])$.}
Examples:

- `$ cd` : change the current working directory,
- `$ ls` : list directory content,
- `$ pwd` : print the name of the current directory,
- `$ cp` : copy files & directories,
- `$ mv` : move or rename files and directories,
- `$ touch` : create file,
- `$ mkdir` : create directory,
- `$ man` : manual page of a command.
Examples:

$ cd : change the current working directory,
$ ls : list directory content,
$ pwd : print the name of the current directory,
$ cp : copy files & directories,
$ mv : move or rename files and directories,
$ touch : create file,
$ mkdir : create directory,
$ man : manual page of a command.

shell := command line interpreter $\xrightarrow{\text{example}}$ Bash
Examples:

- `$ cd`: change the current working directory,
- `$ ls`: list directory content,
- `$ pwd`: print the name of the current directory,
- `$ cp`: copy files & directories,
- `$ mv`: move or rename files and directories,
- `$ touch`: create file,
- `$ mkdir`: create directory,
- `$ man`: manual page of a command.

Shell := command line interpreter \(\xrightarrow{\text{example}}\) Bash

⇒
- shell/bash scripting.
- lot of automation possibilities.

Zoltán Szabó

Linux
Superb text editor: **Vim**

- modal editor $i \xrightarrow{\text{change}} \mathbf{E} \text{sc} = \text{command mode}$,  
- highly customizable & efficient,  
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult ;)</td>
</tr>
<tr>
<td>:w</td>
<td>write (save)</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right)</td>
</tr>
</tbody>
</table>
Superb text editor: **Vim**

- modal editor $i = \text{input mode, Esc} = \text{command mode}$,
- highly customizable & efficient,
- keyboard-driven, language-like.

**Example:**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
</tbody>
</table>
Superb text editor: **Vim**

- modal editor $\xrightarrow{\text{change}} i = \text{input mode}$, \texttt{Esc} = \text{command mode},
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit $\leftarrow$ most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
<tr>
<td>cw</td>
<td>change word,</td>
</tr>
</tbody>
</table>
Superb text editor: Vim

- modal editor $\xrightarrow{\text{change}} i = \text{input mode}$, Esc $= \text{command mode}$,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
<tr>
<td>cw</td>
<td>change word,</td>
</tr>
<tr>
<td>d3w</td>
<td>delete 3 words,</td>
</tr>
</tbody>
</table>

Zoltán Szabó
Linux
Superb text editor: Vim

- modal editor \(\text{i} = \text{input mode}, \text{Esc} = \text{command mode}\),
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

```
:q : quit ← most difficult;),
:w : write (save),
h,j,k,l : arrows (left, down, up, right),
dw : delete word,
cw : change word,
d3w : delete 3 words,
di( : delete inside parentheses,
```
Superb text editor: **Vim**

- modal editor $i \rightarrow$ input mode, Esc = command mode,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

```plaintext
:q    : quit ← most difficult;),
:w    : write (save),
h,j,k,l: arrows (left, down, up, right),
dw    : delete word,
cw    : change word,
d3w    : delete 3 words,
di(    : delete inside parentheses,
:      : delete line,
```
Superb text editor: **Vim**

- modal editor $\xrightarrow{\text{change}} i = \text{input mode}$, $\text{Esc} = \text{command mode}$,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dW</td>
<td>delete word,</td>
</tr>
<tr>
<td>cW</td>
<td>change word,</td>
</tr>
<tr>
<td>d3W</td>
<td>delete 3 words,</td>
</tr>
<tr>
<td>di(</td>
<td>delete inside parentheses,</td>
</tr>
<tr>
<td>dd</td>
<td>delete line,</td>
</tr>
<tr>
<td>p</td>
<td>paste,</td>
</tr>
</tbody>
</table>
Superb text editor: **Vim**

- modal editor $i \xrightarrow{\text{change}} \text{Esc} = \text{command mode}$, input mode,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit (most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
<tr>
<td>cw</td>
<td>change word,</td>
</tr>
<tr>
<td>d3w</td>
<td>delete 3 words,</td>
</tr>
<tr>
<td>di(</td>
<td>delete inside parentheses,</td>
</tr>
<tr>
<td>dd</td>
<td>delete line,</td>
</tr>
<tr>
<td>p</td>
<td>paste,</td>
</tr>
<tr>
<td>y</td>
<td>yank (copy),</td>
</tr>
</tbody>
</table>

Zoltán Szabó

Linux
Superb text editor: Vim

- modal editor change $i = input$ mode, $Esc = command$ mode,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
<tr>
<td>cw</td>
<td>change word,</td>
</tr>
<tr>
<td>d3w</td>
<td>delete 3 words,</td>
</tr>
<tr>
<td>di(</td>
<td>delete inside parentheses,</td>
</tr>
<tr>
<td>dd</td>
<td>delete line,</td>
</tr>
<tr>
<td>p</td>
<td>paste,</td>
</tr>
<tr>
<td>y</td>
<td>yank (copy),</td>
</tr>
<tr>
<td>yy</td>
<td>yank line, . . .</td>
</tr>
</tbody>
</table>
Superb text editor: Vim

- modal editor $\text{change} \rightarrow i = \text{input mode}$, $\text{Esc} = \text{command mode}$,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:q</td>
<td>quit ← most difficult;),</td>
</tr>
<tr>
<td>:w</td>
<td>write (save),</td>
</tr>
<tr>
<td>h,j,k,l</td>
<td>arrows (left, down, up, right),</td>
</tr>
<tr>
<td>dw</td>
<td>delete word,</td>
</tr>
<tr>
<td>cw</td>
<td>change word,</td>
</tr>
<tr>
<td>d3w</td>
<td>delete 3 words,</td>
</tr>
<tr>
<td>di(</td>
<td>delete inside parentheses,</td>
</tr>
<tr>
<td>dd</td>
<td>delete line,</td>
</tr>
<tr>
<td>p</td>
<td>paste,</td>
</tr>
<tr>
<td>y</td>
<td>yank (copy),</td>
</tr>
<tr>
<td>yy</td>
<td>yank line, . . .</td>
</tr>
</tbody>
</table>

- cross-platform.

Zoltán Szabó | Linux
Vim – continued (free ⇒)

- integration to browser, Jupyter notebook, ...
- evolution: vi → Vim → Neovim (community-developed),

personal Wiki: vimwiki,

- tutorials: $ vimtutor and

<table>
<thead>
<tr>
<th>Name</th>
<th>Vids</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThePrimeagen</td>
<td>link₁, link₂, link₃, link₄, link₅, link₆</td>
</tr>
<tr>
<td>Missing Semester</td>
<td>link[O]</td>
</tr>
<tr>
<td>DistroTube</td>
<td>link₁[O], link₂[O]</td>
</tr>
<tr>
<td>Ben Awad</td>
<td>link</td>
</tr>
</tbody>
</table>
Hint: How to ask on forums?

- Netiquette; discussion → guide.
- DIY mentality:
  - the community is friendly and helps *if* you put in effort,
  - ⇒ read & do your research first!
<table>
<thead>
<tr>
<th>Name</th>
<th>Odysee</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistroTube</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>Learn Linux TV</td>
<td>–</td>
<td>link</td>
</tr>
<tr>
<td>Eric Murphy</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>Brodie Robertson</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>EF - Linux Made Simple</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>OldTechBloke</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>Mental Outlaw</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>Luke Smith</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>VeronicaExplains</td>
<td>–</td>
<td>link</td>
</tr>
<tr>
<td>MobileTechReview</td>
<td>–</td>
<td>link</td>
</tr>
<tr>
<td>Naomi Brockwell: NBTV</td>
<td>link</td>
<td>link</td>
</tr>
<tr>
<td>Louis Rossmann</td>
<td>link</td>
<td>link</td>
</tr>
</tbody>
</table>

† but her/his past videos are nice.
Ricing: Conky

- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...
Ricing: Conky

- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...

- Example (further inspiration):

```
Linux 2.6.15-29-686 on i686
eirc @ druuna

Date: Thursday, 25 May
Time: 6:51:15  Uptime: 10h 10m

Temperatures
  CPU: 40.5°C - MB: 37.0°C

CPU: 42%
  R&M: 615M/630M
  Swap: 1% 186M/187M

File systems
  /dev/sda1  53%  4.33G/9.17G
  /dev/sda3  74%  113.92G/187.55G

Now Playing
  Pain of Salvation
  Song for the innocent

"The Perfect Element pt.1"
  2000 - Progressive metal

Collection Information
  Artists: 998
  Compilations: 7
  Albums: 291
  Genres: 41
  Tracks: 2217

Collection Statistics
  Most songs by Black Sabbath (199)
  Most songs are Heavy Metal (691)
  Most songs during 2005 (232)
  Most albums by Black Sabbath (21)
  Most albums are Heavy Metal (64)
  Most albums during 2005 (22)
```
- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, . . .
- Example (further inspiration):

```
explosia  Linux 2.6.12-gentoo-r6 on i686
Batt:  charged 100%

**PROCESSING**

<table>
<thead>
<tr>
<th>NAME</th>
<th>PID</th>
<th>CPU%</th>
<th>MEH%</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>4821</td>
<td>9.50</td>
<td>9.93</td>
</tr>
<tr>
<td>cpufreq</td>
<td>6065</td>
<td>3.01</td>
<td>6.12</td>
</tr>
<tr>
<td>firefox-bin</td>
<td>7234</td>
<td>2.51</td>
<td>9.47</td>
</tr>
<tr>
<td>pypanel</td>
<td>4828</td>
<td>0.84</td>
<td>1.15</td>
</tr>
</tbody>
</table>

**DATA**

<table>
<thead>
<tr>
<th>NAME</th>
<th>PID</th>
<th>CPU%</th>
<th>MEH%</th>
</tr>
</thead>
<tbody>
<tr>
<td>firefox-bin</td>
<td>7270</td>
<td>0.00</td>
<td>9.47</td>
</tr>
<tr>
<td>firefox-bin</td>
<td>7254</td>
<td>0.00</td>
<td>9.47</td>
</tr>
<tr>
<td>firefox-bin</td>
<td>7251</td>
<td>0.00</td>
<td>9.47</td>
</tr>
<tr>
<td>firefox-bin</td>
<td>7250</td>
<td>0.00</td>
<td>9.47</td>
</tr>
</tbody>
</table>

Swap: 2 %
/
Upload: 0 kb/s  Download: 0 kb/s
```
Ricing: Conky

- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...

- Example (further inspiration):

```
explosia  Linux 2.6.12-gentoo-r6 on i686
Batt: charged 100%

PROCESSING
CPU: 1596.06MHz  27%  67°C

NAME   PID   CPU%  MEM%
%     4281   9.50   9.03
cpufreq 6095  3.01   6.12
firefox-bin 7234  2.51  9.47
pypanel 4828  0.84   1.15

DATA
RAM: 24%

NAME   PID   CPU%  MEM%
firefox-bin 7270  0.00   9.47
firefox-bin 7254  0.00   9.47
firefox-bin 7231  0.00   9.47
firefox-bin 7250  0.00   9.47

Swap: 2%

/   81%  29.44G

Upload: 0 kb/s  Download: 0 kb/s
```
Ricing: Conky on desktop

13:52

Monday, 01 February 2016

RAM: 64%

firefox: 31.07
thunderbird: 7.94
spyder: 5.10
dropbox: 4.65

CPU: 15%

Load: 0.77
Processes: 0/227
firefox: 11.25
Xorg: 0.58
lyx: 0.33
plugin-container: 0.25

TO-DO LIST (2)
- Computer Methods Assignments
- Meet the DDF guide
Ricing: Conky on desktop
fast replacement of the status bar,
- date, time, keyboard layout, backlight, volume, MPD, network, CPU, . . .
Ricing: Polybar ($\text{link}_1$, $\text{link}_2$)

- fast replacement of the status bar,
- date, time, keyboard layout, backlight, volume, MPD, network, CPU, . . .

Example follows.
Compositors: for Xorg (a.k.a. X), Wayland

- They can
  - add effects like transparency, animations or blur,
  - be standalone or built into the DE / WM.
Compositors: for Xorg (a.k.a. X), Wayland

- They can
  - add effects like transparency, animations or blur,
  - be standalone or built into the DE / WM.

Compositors: for Xorg (a.k.a. X), Wayland

- They can
  - add effects like transparency, animations or blur,
  - be standalone or built into the DE / WM.
- For Hyprland: website, wiki; demo (screenshot):
Librem 5: by Purism, running PureOS.
Linux phones

2 PinePhone, PinePhone Pro:
- by Pine64,
- PinePhone Pro: Wiki; various op. systems ∈ Arch;
- recipe: vid₁, vid₂, vid₃, vid₄, vid₅.

My choice (more stable and transparent communication)
Summary

- Linux history, user freedom
Summary

- Linux history, user freedom,
- distributions, installation, applications
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE ← WM ← CLI; Vim
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE ← WM ← CLI; Vim,
- educational channels
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE ← WM ← CLI; Vim,
- educational channels,
- ricing: conky, polybar, compositor
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE ← WM ← CLI; Vim,
- educational channels,
- ricing: conky, polybar, compositor,
- Linux phones.
Are you ready to own your computer