Linux – The Operating System of Freedom

Zoltán Szabó © Department of Statistics, LSE (Sept. 22, 2023)
Contents

- A bit of Linux history.
- Linux distributions.
- Installation.
- Applications.
- Ricing and phones.
My journey
My journey

Win → \textit{'Unix'} → Fedora

Zoltán Szabó Linux
My journey

Win → 'Unix' → Fedora → Ubuntu

Zoltán Szabó Linux
My journey

Win → 'Unix' → Fedora → Ubuntu, macOS

start → school → home → school → home
My journey

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

start → school → home → school → home
My journey

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

Win start → school → 'Unix' → school → Ubuntu, school → CentOS → school → Arch.
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...
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,

One of the main secrets free and open source ⇒ knowledge sharing ⇒ creativity can kick in ⇒ versatility!
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, game consoles, smart TVs, smart watches, Amazon Kindle, 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, game consoles, smart TVs, smart watches, Amazon Kindle, international space stations, ... 

One of the main secrets

free and open source $\Rightarrow$ knowledge sharing $\Rightarrow$ creativity can kick in $\Rightarrow$ 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

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,

  ![GPL logo]

  **Free as in Freedom**

  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: ~35 million (2023).
Free vs open; $O :=$ Odysee

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

- goal: to respect user freedom and privacy.

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.

\( \text{to not constrain the user} \)

- free \( \supset \) open-source, \textbf{but}
  - free \( \neq \) open-source: text, vid\( [O] \),
  - open source code can 'spy' on you,

\textit{privacy matters}[O].
Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s: $12.9 \ll $1500 (MacBook Pro; repairing for 15Y@2023)
Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s (vid[$O$]): $12.9 \ll $1500 (MacBook Pro; repairing for 15Y@2023),
- I like to invest in our free future:
  - System76: repairable laptops, Launch keyboard, Pop!_OS,
Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s_{vid[O]}: $12.9 \llleq$ $1500$ (MacBook Pro; repairing for 15Y@2023),
- 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:

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

Various distros (tree):

- there have been $>1000$ distributions,
- currently (Sept. 22, 2023): 273 distributions
Various distros (tree):
- there have been $>1000$ distributions,
- currently (Sept. 22, 2023): 273 distributions,
- but minor differences.

Primary choice to make
point release vs rolling release.
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, Sonoma.
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, Sonoma.

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

(Debian → ) Ubuntu → Pop!_OS; Fedora; openSUSE.
Ubuntu:

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

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

(Debian → ) Ubuntu → Pop!_OS; Fedora; openSUSE.

- **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

- **Ubuntu**: developer: Canonical Ltd., released every six months, LTS every 2 years.
- **Pop!_OS**: \( \approx \) Ubuntu with customized Gnome (COSMIC), maintained by System76.
- **Fedora**: upstream source for Red Hat Enterprise Linux (developed by Red Hat), released every 6 months.
Point release: Linux distributions

(Debian → ) Ubuntu → Pop!_OS; Fedora; openSUSE.

- **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.

- **openSUSE**: Leap,  
  - released 1×/year.
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>point</th>
<th>rolling</th>
</tr>
</thead>
<tbody>
<tr>
<td>always up-to-date</td>
<td>+</td>
</tr>
<tr>
<td>(new software features, bug fixes, security patches)</td>
<td>+</td>
</tr>
<tr>
<td>supports even very new hardware</td>
<td>+</td>
</tr>
<tr>
<td>more secure</td>
<td>+</td>
</tr>
<tr>
<td>no need to reinstall it</td>
<td>+</td>
</tr>
<tr>
<td>requires semi-decent internet</td>
<td>−</td>
</tr>
<tr>
<td>less suited for servers (where stability is max-ed)</td>
<td>−</td>
</tr>
</tbody>
</table>
My choice: **Arch** (released in 2002)

- Rolling release.
- Great package manager (pacman),
  - Fast,
  - Allows parallel downloading.

[Package managers handle dependencies.]

---

Zoltán Szabó  Linux
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;)
   - even offline readable[O] and searchable[O].
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;)
   - even offline readable[O] and searchable[O].
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;)
   - even offline readable[O] and searchable[O].

3 excellent software availability:

Both are searchable.

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

4. Follow the instructions.
Downloading note

- .iso size:
  
  - 11 GB (MacOS Ventura) – for comparison.
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.
.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.
.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.
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) \(\xrightarrow{\text{starts}}\)
2. **bootloader** (such as GRUB \(\leftarrow\) GNU; features & others) \(\xrightarrow{\text{loads}}\)
3. the **kernel** (your operating system).

In practice:

---

Zoltán Szabó Linux
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† partition table to identify the ESP‡, and launches the target application (typically the bootloader).

† no chat 😊, ‡ 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.

![GRUB menu](image)
Instructions: for Arch – scary;)

1. **Step-by-step text guide** (official one).
2. **Video guide:**
   - $\text{vid}_1[O]$: 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;)

1. **Step-by-step text guide** (official one).
2. **Video guide**:
   - $\text{vid}_1[O]:$ EFI; 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: BIOS or UEFI

BIOS:

![BIOS Arch Linux](image)

UEFI:

![UEFI Arch Linux](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.
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 (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.

Zoltán Szabó Linux
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

3 partition formatting:
- **ext4**: more settled – my choice,

![Ext4 File System](image)

- **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 (\(\sim 11\)s):
   - SSD matters: Samsung 970 EVO Plus ← my choice (for laptop).

Zoltán Szabó

Linux
Installation hints – continued

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

7 Use a spare drive (to avoid the wrestling of the op. systems).
Create a normal user (beyond the root; ∈ wheel; sudo).
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, \( p = \text{proprietary} \).

- **Web & mail:**
  - browser: firefox (M), tor-browser (A).
  - e-mail: ProtonMail (W), Tutanota (W), thunderbird (M).
Applications: categorized; some handy ones

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

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

- **Media:**
  - image viewer: feh (M), gthumb (M), geeqie (M).
  - image editor: gimp (M).
  - video player: mplayer (M), vlc (M), mpv (M), celluloid (M).
  - video editor: kdenlive (M).
  - video downloader: yt-dlp (M).
  - spotify:
    - player: spotify\(_p\) (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), teams (A), BigBlueButton (W), Jitsi Meet (W), 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)
- password manager: keepassxc (M)
- RSS/Atom feed reader: newsboat (M)
- screen locker: slock (M)
- minimalistic dock: plank (M)
- remote support: teamviewer (A)
- terminal: tilix (M)
- firewall: ufw (M)
- desktop environment: gnome (M, group), qtile (M, WM).

Zoltán Szabó

Linux
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),
- password manager: **keepassxc** (M)

---

Zoltán Szabó

Linux
eye protection: redshift (M),
file manager: thunar (M),
launcher: dmenu (M), rofi (M),
programming: python (M), jupyter-notebook (M), spyder (M), pycharm-professional (A),
password 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),
- password 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),
- password manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimalistic dock: plank (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),
- password manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimalistic 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),
- password manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimalistic 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),
- password manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- minimalistic dock: plank (M),
- remote support: teamviewer (A),
- terminal: tilix (M),
- firewall: ufw (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),
password manager: keepassxc (M),
RSS/Atom feed reader: newsboat (M),
screen locker: slock (M),
minimalistic 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:
  - windows 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: GNOME
DE: GNOME – my desktop
DE: Xfce

Extra DE inspiration
unixporn (screenshots).

Zoltán Szabó
Linux
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 $\rightarrow$ GNOME,
- KWin $\rightarrow$ KDE,
- Xfwm $\rightarrow$ Xfce,
- Enlightenment $\rightarrow$ 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 → GNOME,
- KWin → KDE,
- Xfwm → Xfce,
- Enlightenment → Enlightenment.

Tiling:
- i3.

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

Examples follow
WM: i3

```c

/* Setup i3 specific atoms like I3_SOCKET_PATH and I3_CONFIG_PATH */
XInitAtomStr();

struct ev_io *xcb_watcher = sclass->sized(struct ev_io);
struct ev_io *xxio = sclass->sized(struct ev_io);
struct ev_check *xcb_check = sclass->sized(struct ev_check);
struct ev_prepare *xcb_prepare = sclass->sized(struct ev_prepare);

xxio_init(xcb_watcher, xcb_get_event, xcb_get_event, ev_file_notifier(conn), EV_READ);
xxio_start(main_loop, xcb_watcher);

if (xxio_supported) {
    ev_init(xcb, xcb_get_event, ConnectionNumber(xcb_connection), EV_READ);
    ev_start(main_loop, xcb_connection);
    /
    / Flush the buffer so that libev can properly get new events */
    xcb_flush(xcb_connection);
}

/* Ignore Signals to disable signal handler */
ex_signal_handler(NULL);

/* Autostarting exec-lines */
if (autostart) {
    TAILForeach(exec, autostarts, autostarts) {
        LOG("autostarting ", exec->execute);
        start_application(exec->execute, command);
    }
}

/* Autostarting exec_always-lines */
if (autostarts_always) {
    TAILForeach(exec_always, autostarts_always, autostarts_always) {
        LOG("autostarting (always) ", exec_always->execute);
        start_application(exec_always->execute, command);
    }
}
```

Zoltán Szabó  Linux
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]
- 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,
Terminal: used for instance @ Arch install

Command shell:

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

![Terminal screenshot with commands]

- Google Colab: `!shellcommand`,
Command shell:

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

Google Colab: `!shellcommand`,

**virtual console** = text terminal + login prompt (ttyX = Ctrl+Alt+FX, X ∈ [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 \ensuremath{=: \text{command line interpreter}} \rightarrow \text{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 $\rightarrow$ Bash $\rightarrow$

- shell/bash scripting.
- lot of automation possibilities.
Superb text editor: Vim

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

Example:

:q : quit $\leftarrow$ most difficult;),
:w : write (save),
h,j,k,l : arrows (left, down, up, right),
Superb text editor: Vim

- modal editor $\text{change} \rightarrow i = \text{input mode}$, Esc = 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,
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:

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

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

Example:

:q : quit $\leftarrow$ most difficult;
:w : write (save),
$h,j,k,l$ : arrows (left, down, up, right),
dw : delete word,
cw : change word,
d3w : delete 3 words,
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>
<tr>
<td>di(</td>
<td>delete inside parentheses,</td>
</tr>
</tbody>
</table>

[Image of Vim logo]
Superb text editor: **Vim**

- modal editor \( \text{change} \rightarrow \text{input mode, Esc = command mode,} \)
- 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 ( \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>
<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>
</tbody>
</table>
Superb text editor: Vim

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

Example:

\begin{itemize}
  \item [:q] : quit $\leftarrow$ most difficult;)
  \item [:w] : write (save),
  \item [h,j,k,l] : arrows (left, down, up, right),
  \item [dw] : delete word,
  \item [cw] : change word,
  \item [d3w] : delete 3 words,
  \item [di()] : delete inside parentheses,
  \item [dd] : delete line,
  \item [p] : paste,
\end{itemize}
Superb text editor: **Vim**

- modal editor $i \xrightarrow{\text{change}}$ 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>
</tbody>
</table>
Superb text editor: **Vim**

- modal editor **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>
<tr>
<td>y</td>
<td>yank (copy)</td>
</tr>
<tr>
<td>yy</td>
<td>yank line</td>
</tr>
</tbody>
</table>

Zoltán Szabó

**Linux**
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>d(</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
● 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>link$ _1$, link$ _2$, link$ _3$, link$ _4$, link$ _5$, link$ _6$</td>
</tr>
<tr>
<td>Missing Semester</td>
<td>link$ [O]$</td>
</tr>
<tr>
<td>DistroTube</td>
<td>link$ _1[O]$, link$ _2[O]$</td>
</tr>
<tr>
<td>Ben Awad</td>
<td>link$ [O]$</td>
</tr>
</tbody>
</table>
Hint: How to ask on forums?

- Netiquette\([O]\); discussion $\rightarrow$ guide.
- DIY mentality:
  - the community is friendly and helps *if* you put in effort,
  - $\Rightarrow$ read & do your research first!
## Odysee (with LBRY) / YouTube channels: good educators

<table>
<thead>
<tr>
<th>Name</th>
<th>Odysee</th>
<th>YouTube</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistroTube</td>
<td>link</td>
<td>link</td>
<td></td>
</tr>
<tr>
<td>Learn Linux TV</td>
<td>—</td>
<td>link</td>
<td></td>
</tr>
<tr>
<td>Eric Murphy</td>
<td>link</td>
<td>link</td>
<td></td>
</tr>
<tr>
<td>Brodie Robertson</td>
<td>link</td>
<td>link</td>
<td></td>
</tr>
<tr>
<td>EF - Linux Made Simple</td>
<td>link</td>
<td>link</td>
<td>(less active nowadays†)</td>
</tr>
<tr>
<td>OldTechBloke</td>
<td>link</td>
<td>link</td>
<td>(less active nowadays†)</td>
</tr>
<tr>
<td>Mental Outlaw</td>
<td>link</td>
<td>link</td>
<td>(Linux, privacy)</td>
</tr>
<tr>
<td>Luke Smith</td>
<td>link</td>
<td>link</td>
<td>(less active nowadays†)</td>
</tr>
<tr>
<td>VeronicaExplains</td>
<td>—</td>
<td>link</td>
<td>(less active†)</td>
</tr>
<tr>
<td>MobileTechReview</td>
<td>—</td>
<td>link</td>
<td>laptop &amp; mobile reviews</td>
</tr>
<tr>
<td>Naomi Brockwell: NBTV</td>
<td>link</td>
<td>link</td>
<td>privacy</td>
</tr>
<tr>
<td>Louis Rossmann</td>
<td>link</td>
<td>link</td>
<td>right to repair</td>
</tr>
</tbody>
</table>

† but her/his past videos are nice.
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 @ drwma

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

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

---
CPU: 42%
RAM: 97% 615M/630M
Swap: 7% 18.6M/357M

---
File systems
/dev/sda1 53% 4.335G/6.17G
/dev/sda3 74% 113.92G/137.66G

---
Now Playing
Pain of Salvation
Song for the innocent

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

---
Collection Information
Artists: 98    Compilations: 7
Albums: 271   Genres: 41
Tracks: 2217

---
Collection Statistics
Most songs by Black Sabbath (100)
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)
```

Zoltán Szabó  Linux
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: 1596.651MHz 27 % 67°C

NAME     PID  CPU% NR% M%8%
%       4821  9.50  9.33
cpureqd  6065  3.01  6.12
firefox-bin  7234  2.51  9.47
pypanel  4828  6.84  1.15

DATA
RAH:     24 %

NAME     PID  CPU% NR% M%8%
firefox-bin  7270  0.00  9.47
firefox-bin  7254  0.00  9.47
firefox-bin  7251  0.00  9.47
firefox-bin  7250  0.00  9.47

Swap:     2 %

/:
81%  29.446

Upload: 0 kb/s  Download: 0 kb/s
```

Zoltán Szabó  Linux
- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...
- Example (further inspiration):

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

PROCESSING
CPU: 1596.65MHz  27%   67°C

NAME     PID     CPU%   MEM%
%         4261    9.50    9.03
cpufreqd  6065    3.91    6.12
firefox-bin 7234    2.51    9.47
pypanel   4828    6.84    1.15

DATA
RAW: 24%

NAME     PID     CPU%   MEM%
firefox-bin 7270    0.00    9.47
firefox-bin 7254    0.00    9.47
firefox-bin 7251    0.00    9.47
firefox-bin 7250    0.00    9.47

Swap: 2%

/:
91%  29.44G

Upload: 0 KB/s  Download: 0 KB/s
```
Ricing: Conky on desktop
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.
Polybar example
Composers: 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):
Linux phones (security & privacy; beta!)

Librem 5:
- by Purism, running PureOS.
Linux phones

- PinePhone, PinePhone Pro:
  - by Pine64,
  - PinePhone Pro: Wiki; various op. systems \(\supseteq\) Arch;
  - recipe: vid$_1$, vid$_2$, vid$_3$, vid$_4$, vid$_5$.

My choice (more stable and transparent communication)
Linux history, user freedom
• 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