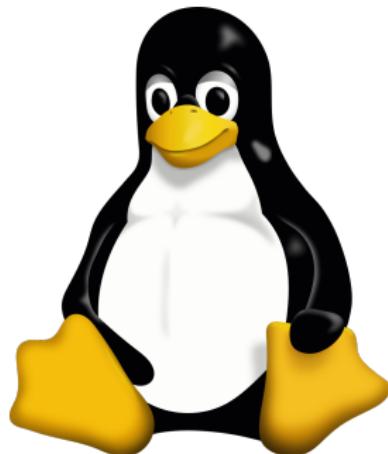


Linux – The Operating System of Freedom

Zoltán Szabó, Department of Statistics, LSE (May 19, 2023)



Contents

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

My journey

Win
start

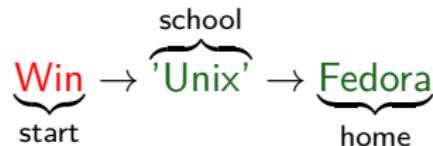


My journey

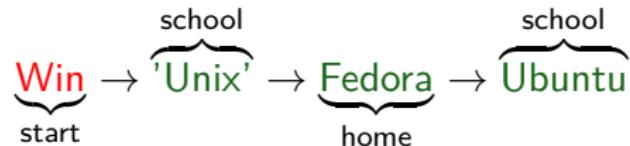
 **Win** → **'Unix'**
start



My journey



My journey

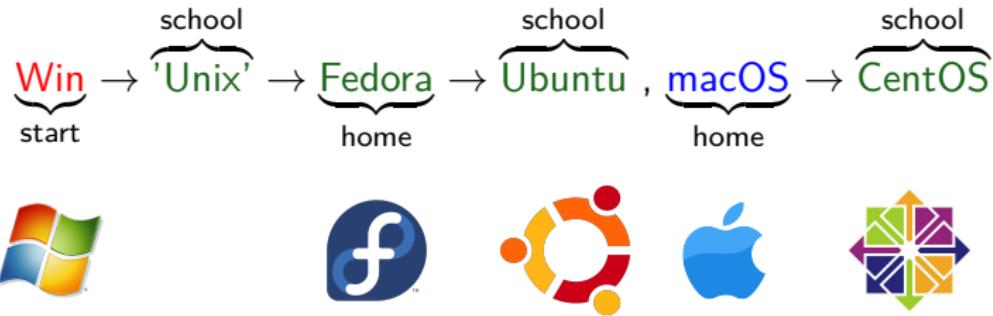


My journey

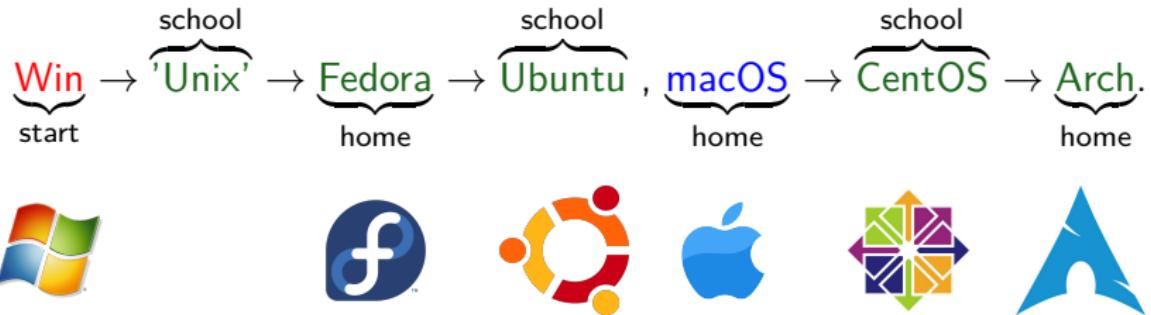
```
graph LR; Win[Win] --> Unix['Unix']; Unix --> Fedora[Fedora]; Fedora --> Ubuntu[Ubuntu]; Ubuntu --> macOS[macOS]
```



My journey



My journey



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...)

Some fun: 30th birthday = 2 years ago. Notation: [link](#).

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...)



#30YearsofLinux



Some fun: 30th birthday = 2 years ago. Notation: [link](#).

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...)



#30YearsofLinux



Linus Torvalds (~now):



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



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



- At the heart of > 3 billion Android devices,

- 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, . . .

- 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)
 - ① to use,
 - ② to study,
 - ③ to modify, and
 - ④ 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)
 - ① to use,
 - ② to study,
 - ③ to modify, and
 - ④ 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)
 - ① to use,
 - ② to study,
 - ③ to modify, and
 - ④ 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)
 - ① to use,
 - ② to study,
 - ③ to modify, and
 - ④ 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;)



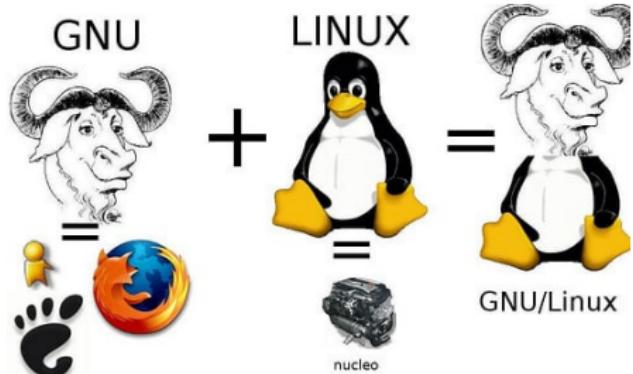
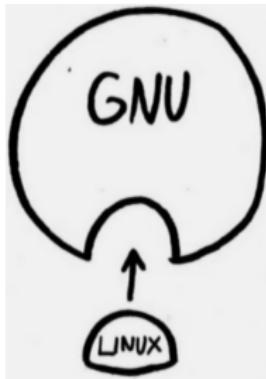
- Founder of the **Free Software Foundation** ('85) → **resources**,
 - non-profit organization to support the free software movement.
- Author of the **GNU GPL license** ('89): for the GNU project,



1st **copyleft** license

any derivative work must be distributed under the same terms.

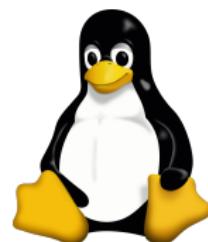
- Creator of **GNU Emacs**: 'text editor' (**LISP** interpreter).



⇒

- 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.

to *not constrain* the user

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



Free vs right to repair

- idea in 60s_{vid[O]}: \$12 ≪ \$1500 (MacBook Pro),
- my experience: battery replacement in Surface Pro = 600€,



Free vs right to repair

- idea in 60s_{vid[O]}: \$12 ≪ \$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_{vid[O]}: \$12 ≪ \$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.



- **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).

- **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 (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

point	rolling
always up-to-date	+
(new software features, bug fixes, security patches)	
supports even very new hardware	+
more secure	+
no need to reinstall it	+
requires semi-decent internet	-
less suited for servers (where stability is max-ed)	-

My choice: Arch (released in 2002)

- ① rolling release.
- ② great package manager (pacman),
 - fast,
 - allows parallel downloading.



[Package managers handle dependencies.]

My choice: Arch (released in 2002)

- ① rolling release.
- ② great package manager (pacman),

- fast,
- allows parallel downloading.



[Package managers handle dependencies.]

- ③ superb documentation ([Arch Wiki](#)):
 - base reference even for other distributions;)
 - [readable](#)[O] and [searchable](#)[O] offline.

My choice: Arch (released in 2002)

- ① rolling release.
- ② great package manager (pacman),

- fast,
- allows parallel downloading.



[Package managers handle dependencies.]

- ③ superb documentation ([Arch Wiki](#)):
 - base reference even for other distributions;)
 - [readable](#)[O] and [searchable](#)[O] offline.
- ④ excellent software availability:
 - [main](#): 14K

My choice: Arch (released in 2002)

- ① rolling release.
- ② great package manager (pacman),

- fast,
- allows parallel downloading.



[Package managers handle dependencies.]

- ③ superb documentation ([Arch Wiki](#)):
 - base reference even for other distributions;)
 - [readable](#)[O] and [searchable](#)[O] offline.
- ④ excellent software availability:
 - [main](#): 14K, [AUR](#): 90K packages.

Both are searchable.

Installation

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



Installation

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



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

Installation

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



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

Installation

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



- ➋ Enter into the 'BIOS' (by pressing Esc, F1/F2/...):
 - disable Secure boot,
 - choose the boot medium to be the USB stick.
- ➌ Boot from the USB stick.
- ➍ Follow the instructions.

Downloading note

- .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.

Downloading note

- .iso size:

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

Downloading note

- .iso size:
 - 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:

- 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 (\Leftarrow 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:

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).

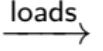
In practice:

- **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.

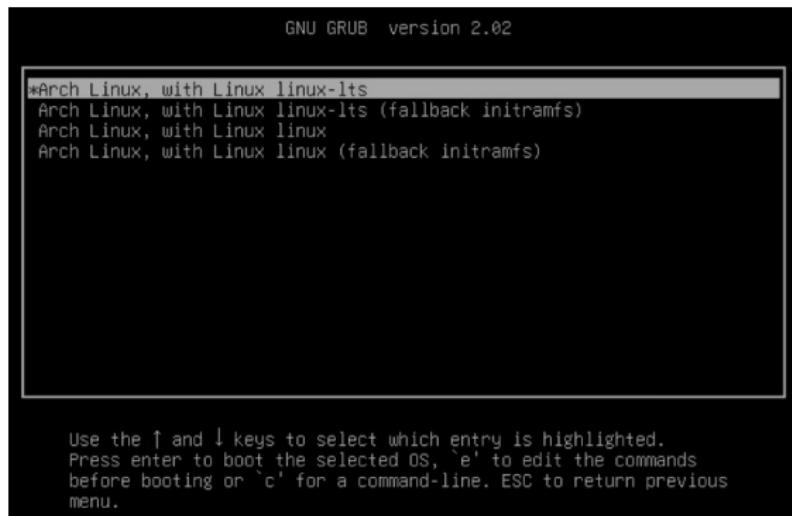
Notes on the boot process — a 'bit' technical

- 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.



Instructions: for Arch – scary;)

- ① Step-by-step text guide (official one).
- ② Video guide:
 - vid₁[O]: UEFI; check the YouTube comments as well!
 - vid₂: BIOS, UEFI, UEFI-LVM-LUKS.

Instructions: for Arch – scary;)

- ① Step-by-step text guide (official one).
- ② Video guide:
 - vid₁[O]: UEFI; check the YouTube comments as well!
 - vid₂: BIOS, UEFI, UEFI-LVM-LUKS.

Definitions

- firmware: BIOS (very old machine), UEFI (semi-new computer).
- partition table: BIOS ⇒ MBR (a.k.a. DOS, MS-DOS); UEFI ⇒ GPT.
- LVM: adjustable layout, LUKS: encryption.
- LUKS: your data can't be read even if your laptop is stolen.

Instructions: BIOS or UEFI

BIOS:



UEFI:



Installation hints

- ① use ethernet: faster.

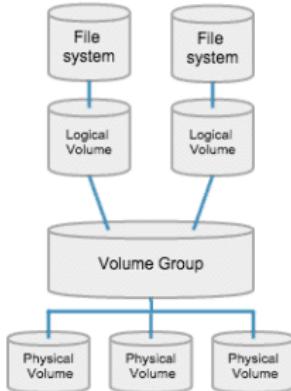


Installation hints

- 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.

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,



- **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!

Installation hints – continued

- 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.

Installation hints – continued

- 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, ×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 (~ 11s):
 - SSD matters: Samsung 970 EVO Plus ← my choice (for laptop).



- ⑦ Use a spare drive (to avoid the wrestling of the op. systems).

Installation hints – continued

- ⑧ Create a normal user (beyond the root; \in wheel; sudo).

Installation hints – continued

- ⑧ Create a normal user (beyond the root; \in wheel; sudo).
- ⑨ Log your installation, usage, information sources (e.g. by Vimwiki)!

Installation hints – continued

- 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.

Applications: categorized; some handy ones

Notations: **M** = 'in main', **A** = 'in AUR', **W** = web client, **P** = pip,
✓ = installed by default.

- Web & mail:

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

Applications: categorized; some handy ones

Notations: **M** = 'in main', **A** = 'in AUR', **W** = web client, **P** = pip,
✓ = installed by default.

- Web & mail:

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

- Media:

- image viewer: `feh (M)`, `gthumb (M)`, `geequery (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)`.

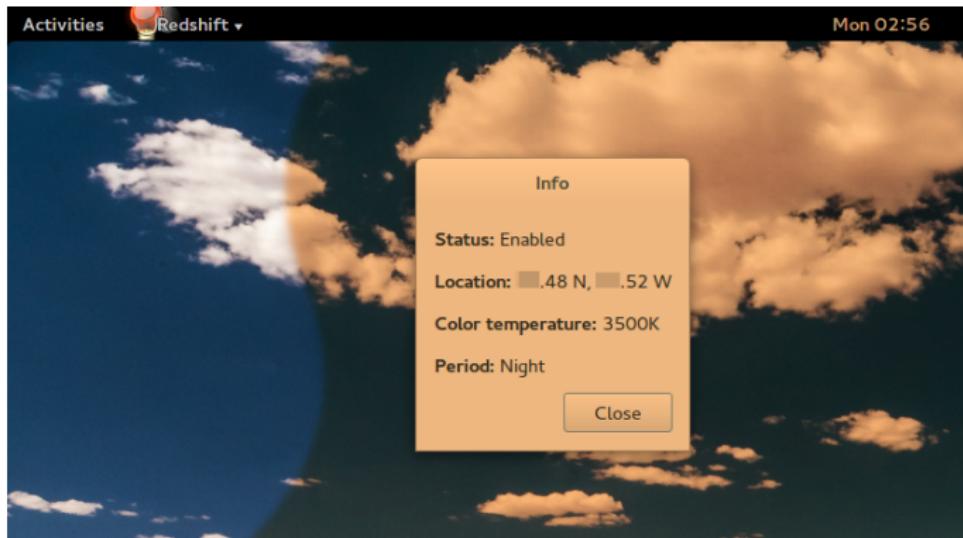
- Text:
 - document viewer: `xdvi` (✓, ∈ texlive), `xpdf` (M),
 - .pdf annotation: `xournalapp` (M),
 - text editing: `texlive-most` (L^AT_EX, group, M), `kile` (M) (≈ WinEdt), `vim` (M), `libreoffice-still` (M), `notepadqq` (M).

Applications+

- Text:
 - document viewer: `xdvi` (✓, ∈ texlive), `xpdf` (M),
 - .pdf annotation: `xournalapp` (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),

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),

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:
 - **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

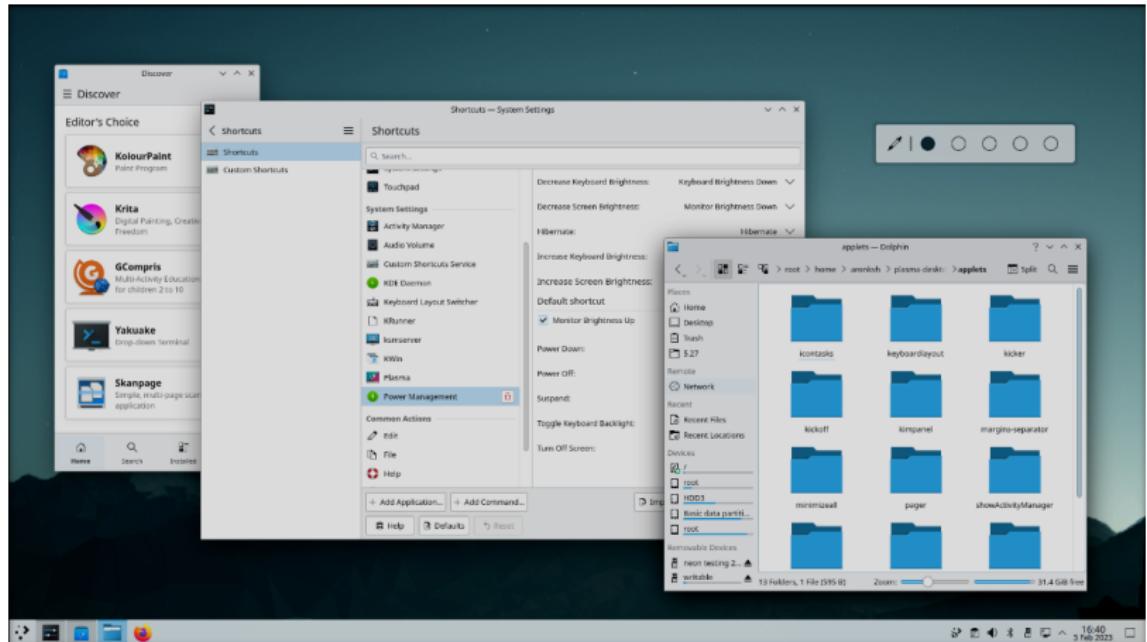
DE: GNOME



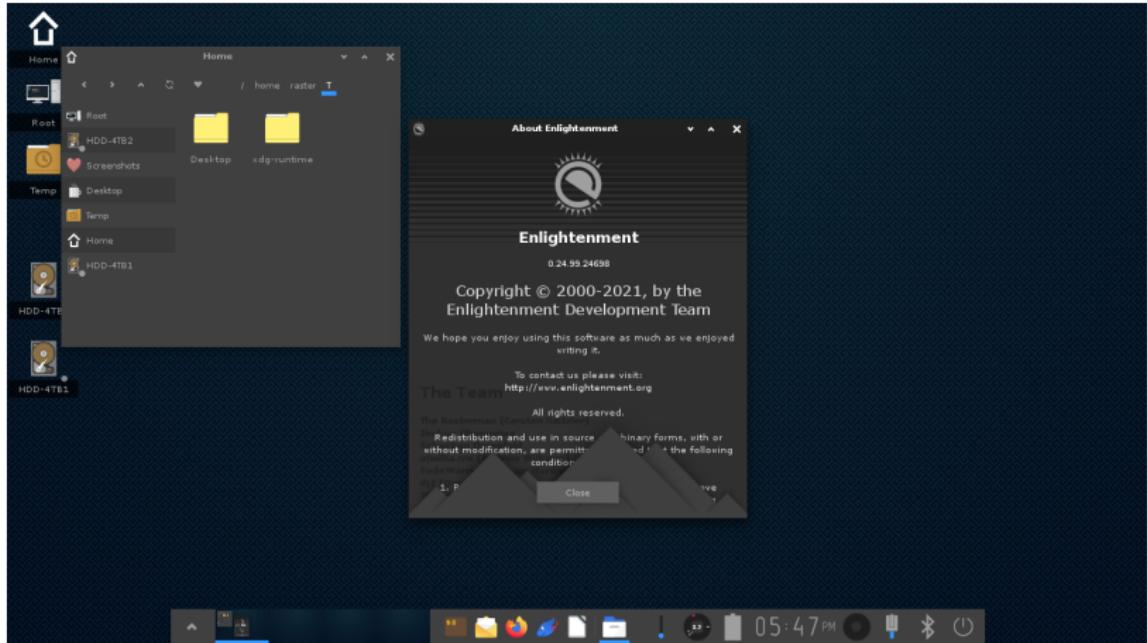
DE: GNOME – my desktop



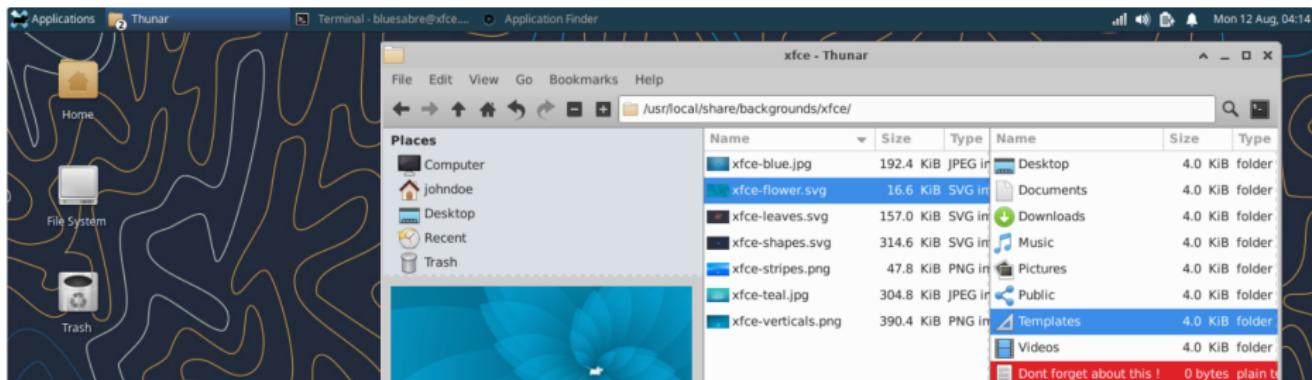
DE: KDE Plasma



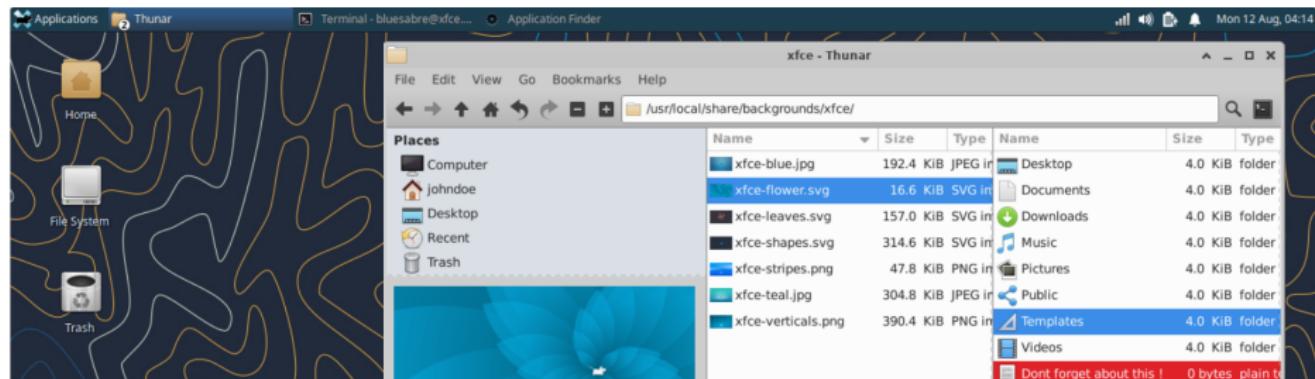
DE: Enlightenment



DE: Xfce



DE: Xfce



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 → GNOME,
 - KWin → KDE,
 - Xfwm → Xfce,
 - Enlightenment → Enlightenment.
- Tiling:
 - i3.
- Dynamic:
 - Qtile:
 - it uses Python;)
 - various (69) widgets.

Examples follow

WM: i3

```
main.c ("/i3/src") - VIM
}
/* Set up i3 specific atoms like I3_SOCKET_PATH and I3_CONFIG_PATH */
x_set_i3_atoms();

struct ev_io *xcb_watcher = malloc(sizeof(struct ev_io));
struct ev_io *xkb = malloc(sizeof(struct ev_io));
struct ev_check *xcb_check = malloc(sizeof(struct ev_check));
struct ev_prepare *xcb_prepare = malloc(sizeof(struct ev_prepare));

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

if (xkb_supported) {
    ev_io_init(xkb, xkb_get_event, ConnectionNumber(xkbfd), EV_READ);
    ev_io_start(main_loop, xkb);
}

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

ev_check_init(xcb_check, xcb_check_db);
ev_check_start(main_loop, xcb_check);

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

xcb_flush(conn);

manage_existing_windows(root);

if ((disable_signal_handler)
    setuse_signal_handler();

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

/* Autostarting exec-lines */
if (autostart) {
    struct Autostart *exec;
    TAILQ_FOREACH(exec, autostarts, autostarts) {
        LOG("auto-starting %s\n", exec->command);
        start_application(exec->command);
    }
}

/* Autostarting exec_always_lines */
struct Autostart *exec_always;
TAILQ_FOREACH(exec_always, autostarts_always, autostarts_always) {
    LOG("auto-starting (always!) %s\n", exec_always->command);
    start_application(exec_always->command);
}

ev_loops(main_loop, 0);
src/main.c
464,9      992
```

x200: xplayer 902/E05.avi

MP4Player



x200: git log

```
commit b8774212b3802badb615418f45d85566fcfc149b
Author: Michael Steapelberg <michael@stapelberg.de>
Date: Sun Jul 17 22:08:00 2011 +0200
```

 Add missing function prototype for strndup on Darwin (Thanks Marcus)

```
commit 3da3e691063f7c47f09461bb948f48abeb6fc
Author: Michael Steapelberg <michael@stapelberg.de>
Date: Sun Jul 17 15:21:57 2011 +0200
```

 i3-config-wizard: use fgetln on Darwin, use strndup from FreeBSD on Darwin (Thanks Marcus)

```
commit 02dfb3e091a759d2152919779f6d00929a71cc
Author: Michael Steapelberg <michael@stapelberg.de>
Date: Sun Jul 17 15:18:45 2011 +0200
```

 use memmem and strndup from FreeBSD on Darwin (Thanks Marcus)

```
commit fe563ad6995eff95e9019bf98ba0a72e21d133b
Author: Michael Steapelberg <michael@stapelberg.de>
Date: Sun Jul 17 15:17:24 2011 +0200
```

 makefile: link -liconv on Darwin (Thanks Marcus)

```
commit 7512f633a79c290f4e60287fb2bcd689f025b42
Author: Michael Steapelberg <michael@stapelberg.de>
Date: Fri Jul 15 15:21:39 2011 +0200
[]
```

1 2 3 4

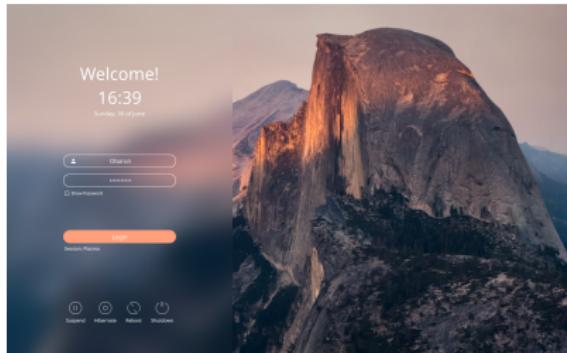
2001:4d80:100e:23:2f:16ff:fe1a:f9b0 | 2.3 GB | DHCP no | VPNs no | Us down | Et 192.168.1.42 (1000 Mbit/s) | MTU 74, ISS 1#] | 01s | 0,03 | 2011-07-22 15:02:32

WM: Qtile



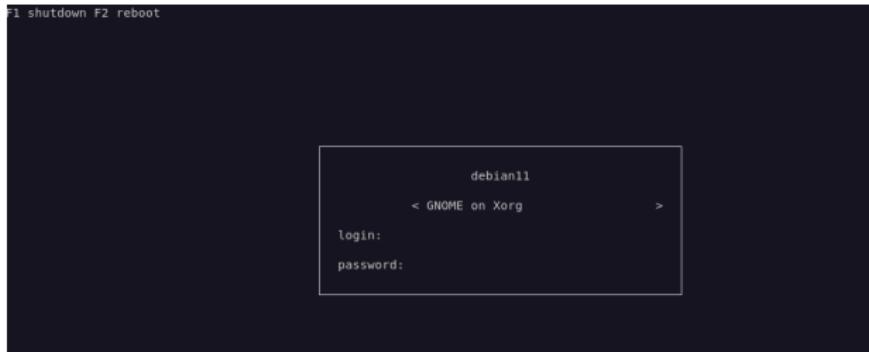
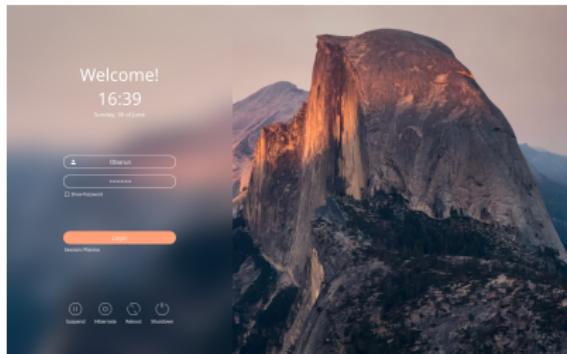
Login/display manager

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



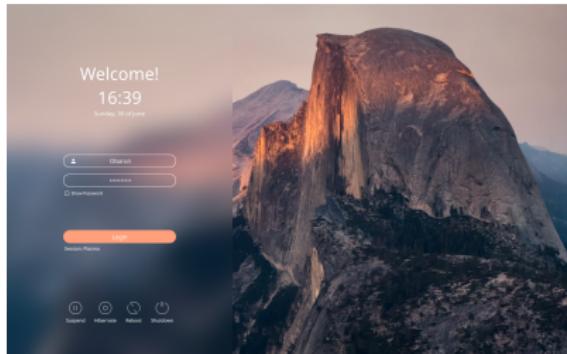
Login/display manager

- 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

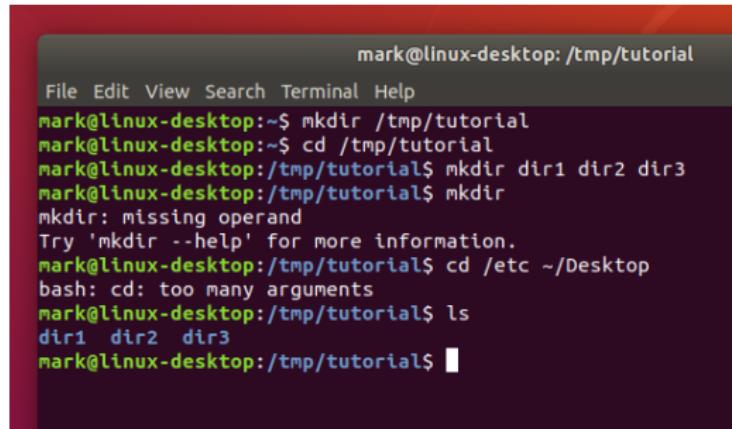
- It gives graphical login – if you prefer not using/starting from tty:)
- Popular choices: (i) SDDM: [tutorial\[O\]](#) ([further inspiration](#)), (ii) Ly: [tutorial\[O\]](#), (iii) GDM: [tutorial\[O\]](#) ([further inspiration](#)).



Terminal: used for instance @ Arch install

Command shell:

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



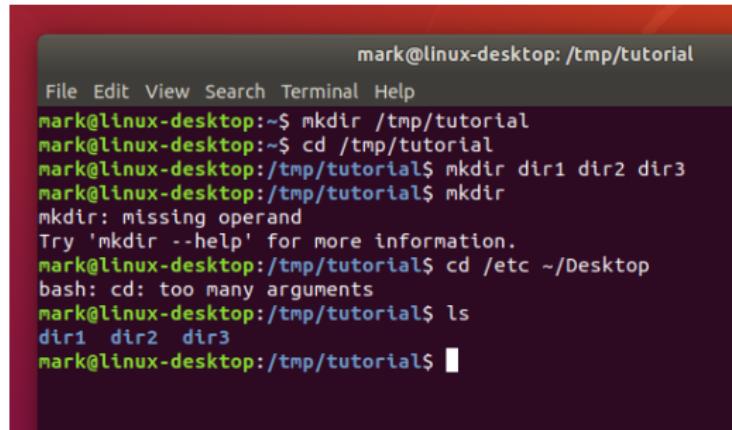
A screenshot of a terminal window titled "mark@linux-desktop: /tmp/tutorial". The window has a dark background and a light gray title bar. The terminal itself has a dark background with white text. It shows a series of commands being entered and executed:

```
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 ~/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,



The screenshot shows a terminal window with a dark background and a red header bar. The header bar contains the text "mark@linux-desktop: /tmp/tutorial". Below the header is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows the following command-line session:

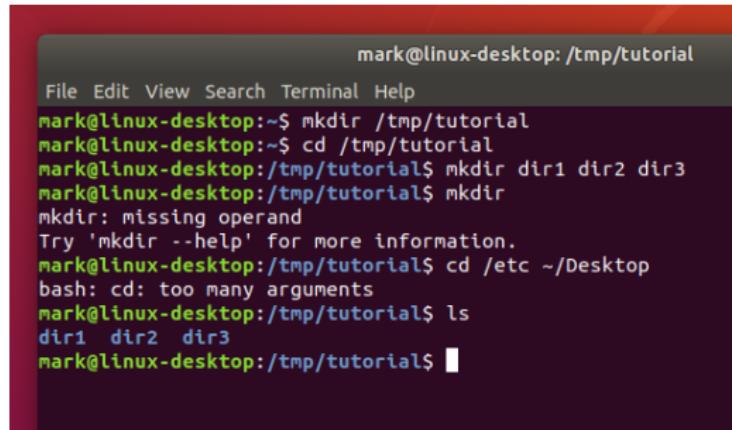
```
mark@linux-desktop:~$ mkdir /tmp/tutorial
mark@linux-desktop:~$ cd /tmp/tutorial
mark@linux-desktop:/tmp/tutorial$ mkdir dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
mark@linux-desktop:/tmp/tutorial$ cd ~/Desktop
bash: cd: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

- Google Colab: !shellcommand,

Terminal: used for instance @ Arch install

Command shell:

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



A screenshot of a terminal window titled "mark@linux-desktop: /tmp/tutorial". The window has a dark background and a light-colored title bar. The terminal shows the following command-line session:

```
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 ~/Desktop
bash: cd: too many arguments
mark@linux-desktop:/tmp/tutorial$ ls
dir1 dir2 dir3
mark@linux-desktop:/tmp/tutorial$ █
```

- Google Colab: !shellcommand,
- **virtual console** = text terminal + login prompt (`ttyX` = `Ctrl+Alt+FX`, $X \in [7]$).

Command line (CLI) – continued

- 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.

Command line (CLI) – continued

- 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

Command line (CLI) – continued

- 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.

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 ← most difficult;),
`:w` : write (save),
`h,j,k,l` : arrows (left, down, up, right),



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 ← 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` = input mode, `Esc` = command mode,
- highly customizable & efficient,
- keyboard-driven, language-like.

Example:

<code>:q</code>	: quit ← most difficult;),
<code>:w</code>	: write (save),
<code>h,j,k,l</code>	: arrows (left, down, up, right),
<code>dw</code>	: delete word,
<code>cw</code>	: change word,



Superb text editor: Vim

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

Example:

<code>:q</code>	: quit ← most difficult;),
<code>:w</code>	: write (save),
<code>h,j,k,l</code>	: arrows (left, down, up, right),
<code>dw</code>	: delete word,
<code>cw</code>	: change word,
<code>d3w</code>	: delete 3 words,



Superb text editor: Vim

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

Example:

<code>:q</code>	: quit ← most difficult;),
<code>:w</code>	: write (save),
<code>h,j,k,l</code>	: arrows (left, down, up, right),
<code>dw</code>	: delete word,
<code>cw</code>	: change word,
<code>d3w</code>	: delete 3 words,
<code>di(</code>	: delete inside parentheses,



Superb text editor: Vim

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

Example:

<code>:q</code>	: quit ← most difficult;),
<code>:w</code>	: write (save),
<code>h,j,k,l</code>	: arrows (left, down, up, right),
<code>dw</code>	: delete word,
<code>cw</code>	: change word,
<code>d3w</code>	: delete 3 words,
<code>di(</code>	: delete inside parentheses,
<code>dd</code>	: delete line,



Superb text editor: Vim

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

Example:

<code>:q</code>	: quit ← most difficult;),
<code>:w</code>	: write (save),
<code>h,j,k,l</code>	: arrows (left, down, up, right),
<code>dw</code>	: delete word,
<code>cw</code>	: change word,
<code>d3w</code>	: delete 3 words,
<code>di(</code>	: delete inside parentheses,
<code>dd</code>	: delete line,
<code>p</code>	: paste,



Superb text editor: Vim

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

Example:

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



Superb text editor: Vim

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

Example:

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



Superb text editor: Vim

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

Example:

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

-
- cross-platform.



Vim – continued (free ⇒)

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



- personal Wiki: vimwiki,
- tutorials: \$ vimtutor and

Name	Vids
ThePrimeagen	link₁ , link₂ , link₃ , link₄ , link₅ , link₆
Missing Semester	link[O]
DistroTube	link₁[O] , link₂[O]
Ben Awad	link

Hint: How to ask on forums?

- Netiquette[O]; discussion → guide.
- DIY mentality:
 - the community is friendly and helps *if* you put in effort,
 - ⇒ **read & do your research first!**



Odysee (with LBRY) / YouTube channels: good educators

Name	Odysee	YouTube	
DistroTube	link	link	
Learn Linux TV	—	link	
Eric Murphy	link	link	
Brodie Robertson	link	link	
EF - Linux Made Simple	link	link	(less active nowadays [†])
OldTechBloke	link	link	(less active nowadays [†])
Mental Outlaw	link	link	(Linux, privacy)
Luke Smith	link	link	(less active nowadays [†])
VeronicaExplains	—	link	(less active [†])
MobileTechReview	—	link	laptop & mobile reviews
Naomi Brockwell: NBTV	link	link	privacy
Louis Rossmann	link	link	right to repair

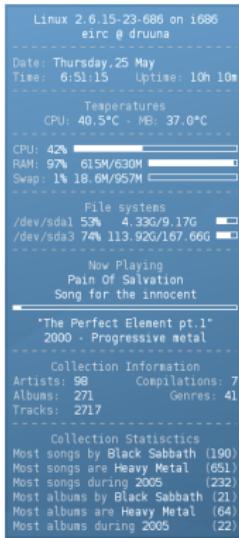
[†] but her/his past videos are nice.

Ricing: Conky

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

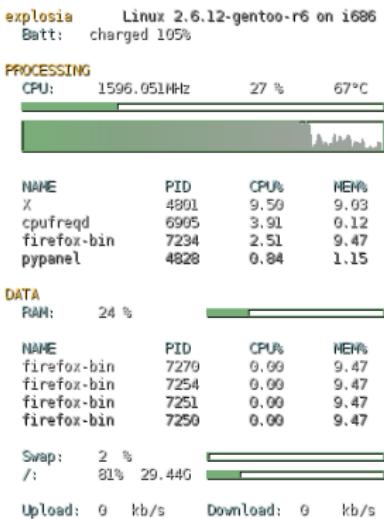
Ricing: Conky

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



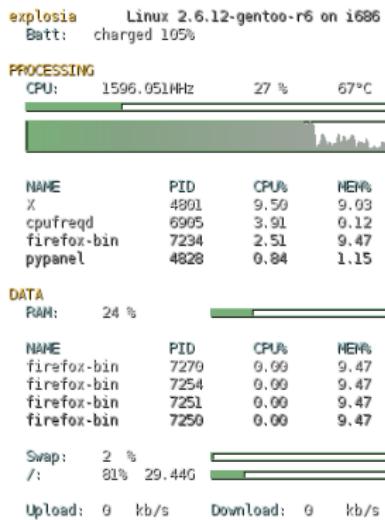
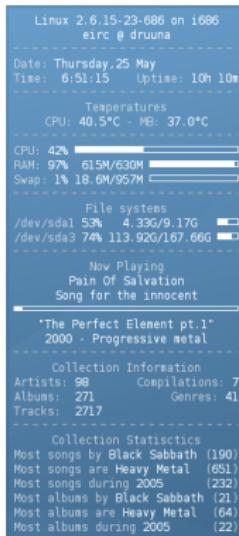
Ricing: Conky

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



Ricing: Conky

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



Ricing: Conky on desktop



Ricing: Conky on desktop



Ricing: Polybar ([link₁](#), [link₂](#))



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

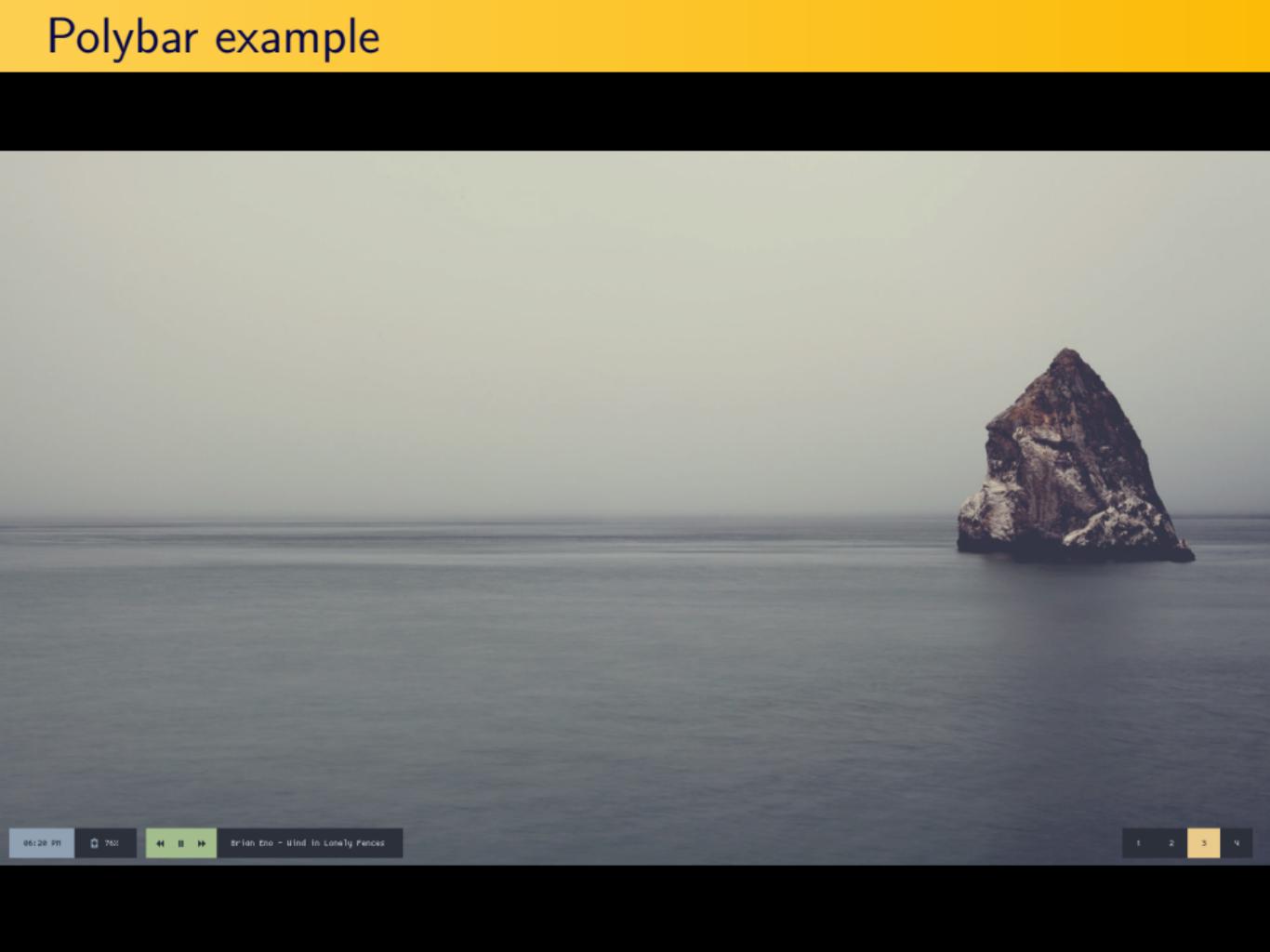
Ricing: Polybar ([link₁](#), [link₂](#))



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

Example follows.

Polybar example



06:28 PM

76%



brian zho - Wind in Lonely Fences

1 2 3 4

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

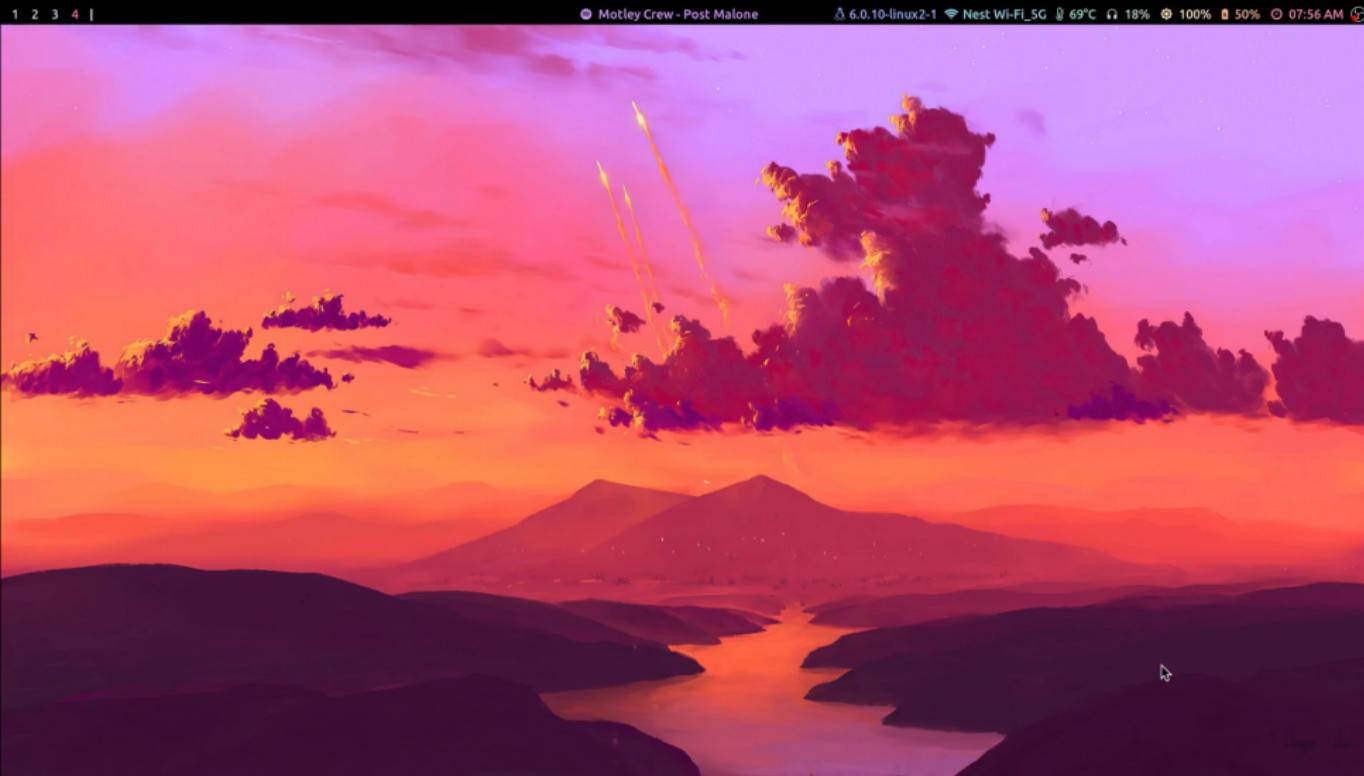
- They can
 - add effects like transparency, animations or blur,
 - be standalone or built into the DE / WM.
- Example: Picom, Hyprland, Wayfire. Related fun[O] with Gnome extensions.

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.
- Example: [Picom](#), [Hyprland](#), [Wayfire](#). Related fun[0] with Gnome extensions.
- For Hyprland: [website](#), [wiki](#); [demo](#) (screenshot):



Composer: Hyperlnd_{vid}



Compositor: Wayfire_{vid}

Simple •



Linux phones (security & privacy; beta!)

① Librem 5:

- by Purism, running PureOS.



Linux phones

② PinePhone, PinePhone Pro:

- by Pine64,
- PinePhone Pro: [Wiki](#); various op. systems \ni 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

