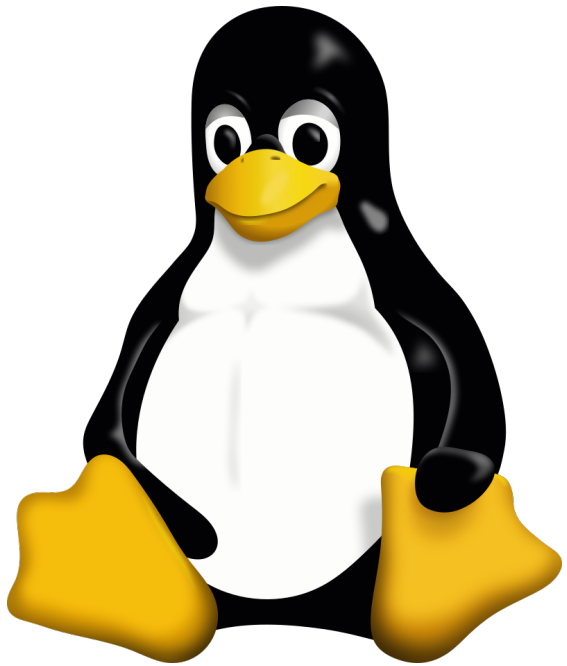
Linux – The Operating System of Freedom

Zoltán Szabó @ Department of Statistics, LSE (Sept. 27, 2024)



Contents

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

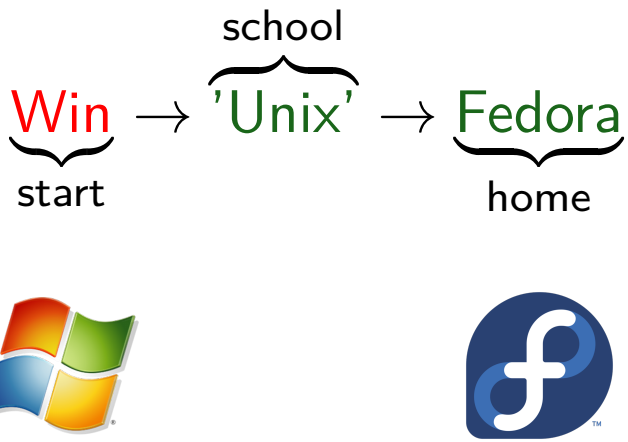
My journey

Win
start

My journey

$$\underbrace{\text{Win}}_{\text{start}} \rightarrow \overbrace{\text{'Unix'}}^{\text{school}}$$

My journey



My journey

$$\underbrace{\text{Win}}_{\text{start}} \rightarrow \overbrace{\text{'Unix'}}^{\text{school}} \rightarrow \underbrace{\text{Fedora}}_{\text{home}} \rightarrow \overbrace{\text{Ubuntu}}^{\text{school}}$$

My journey

$$\underbrace{\text{Win}}_{\text{start}} \rightarrow \overbrace{\text{'Unix'}}^{\text{school}} \rightarrow \underbrace{\text{Fedora}}_{\text{home}} \rightarrow \overbrace{\text{Ubuntu}}^{\text{school}}, \underbrace{\text{macOS}}_{\text{home}}$$




My journey

$$\underbrace{\text{Win}}_{\text{start}} \rightarrow \overbrace{\text{'Unix'}}^{\text{school}} \rightarrow \underbrace{\text{Fedora}}_{\text{home}} \rightarrow \overbrace{\text{Ubuntu}}^{\text{school}}, \underbrace{\text{macOS}}_{\text{home}} \rightarrow \overbrace{\text{CentOS}}^{\text{school}}$$





My journey

Win (start) → 'Unix' (school) → Fedora (home) → Ubuntu (school), macOS (home) → CentOS (school) → Arch (home).






Some fun (a 21Y Finish CS student)

```
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 = 3 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...

Some fun: 30th birthday = 3 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...
```

Linus Torvalds (∼now):

Linux today; OD := Odysee, YT := YouTube (as fallback)

- Supercomputer world: 100% market share,
- Runs: from old laptops [OD_1, YT_1; 1980=YT_2] to top 500 supercomputers, even on a RISC-V laptop [YT]

Linux today; OD := Odysee, YT := YouTube (as fallback)

- Supercomputer world: 100% market share,
- Runs: from old laptops [OD_1, YT_1; 1980=YT_2] to top 500 supercomputers, even on a RISC-V laptop [YT]

- At the heart of $>$ 3 billion Android devices

Linux today; OD := Odysee, YT := YouTube (as fallback)

- Supercomputer world: 100% market share,
- Runs: from old laptops [OD_1, YT_1; 1980=YT_2] to top 500 supercomputers, even on a RISC-V laptop [YT]

- 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; OD := Odysee, YT := YouTube (as fallback)

- Supercomputer world: 100% market share,
- Runs: from old laptops [OD_1, YT_1; 1980=YT_2] to top 500 supercomputers, even on a RISC-V laptop [YT]

- 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 $\Rightarrow$

- 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 $\Rightarrow$

- 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 $\neq$ gratis)

In fact, Linux = GNU/Linux: Linus used the GNU development tools for his kernel, ...

- Late 1970s: companies started to spread proprietary software $\Rightarrow$

- 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 $\neq$ 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 $\Rightarrow$

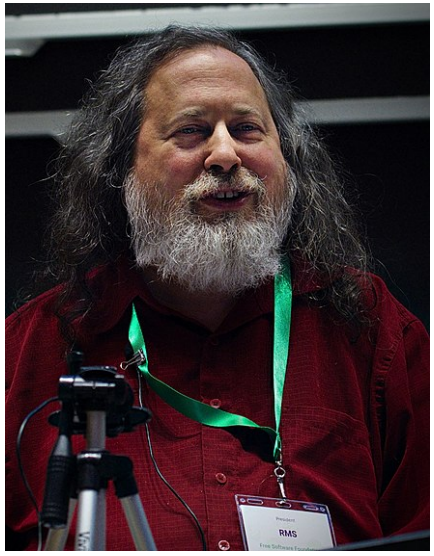
- GNU project = GNU is Not Unix:
  - Goal: write a UNIX-like operating system entirely of free software.
  - Users are legally free (GPL)

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

user freedom (to make choices; free $\neq$ gratis)

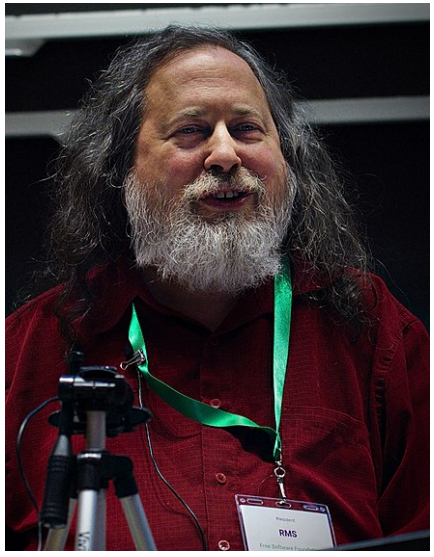
  - give computer users freedom and control in their use of their computers.
  - mass collaboration announced by Richard Stallman ('83; his website).

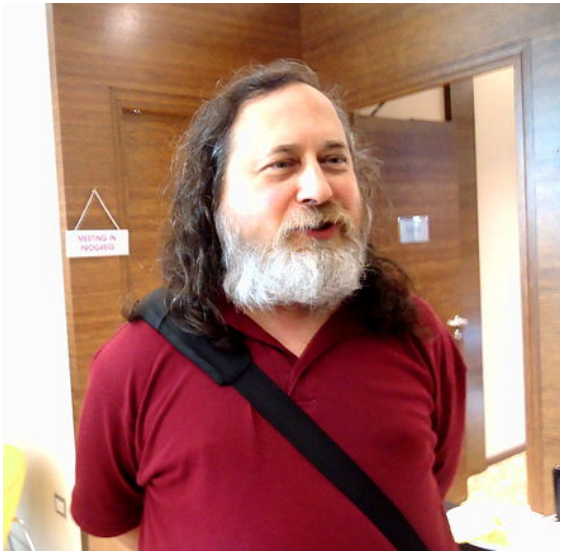
Richard (Matthew) Stallman: RMS – his hacker name;)




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

Richard (Matthew) Stallman: RMS – his hacker name;)





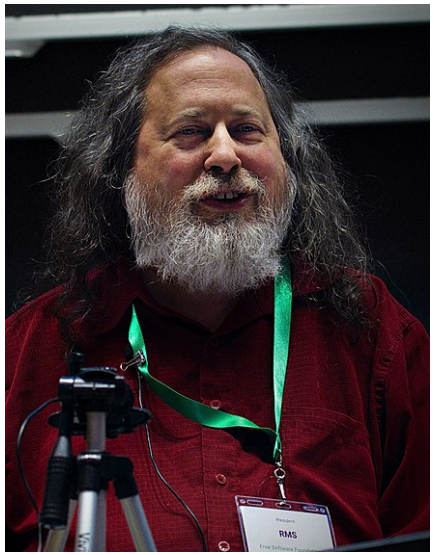

- Founder of the Free Software Foundation ('85) $\rightarrow$ resources,
  - non-profit organization to support the free software movement.
- Author of the GNU GPL license ('89): for the GNU project,

GPL v3
Free Software
Free as in Freedom

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,

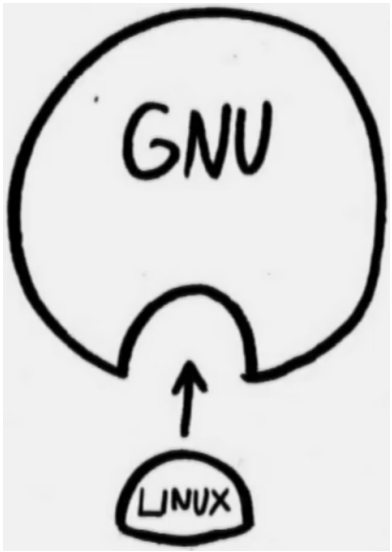
GPL v3 Free Software
Free as in Freedom

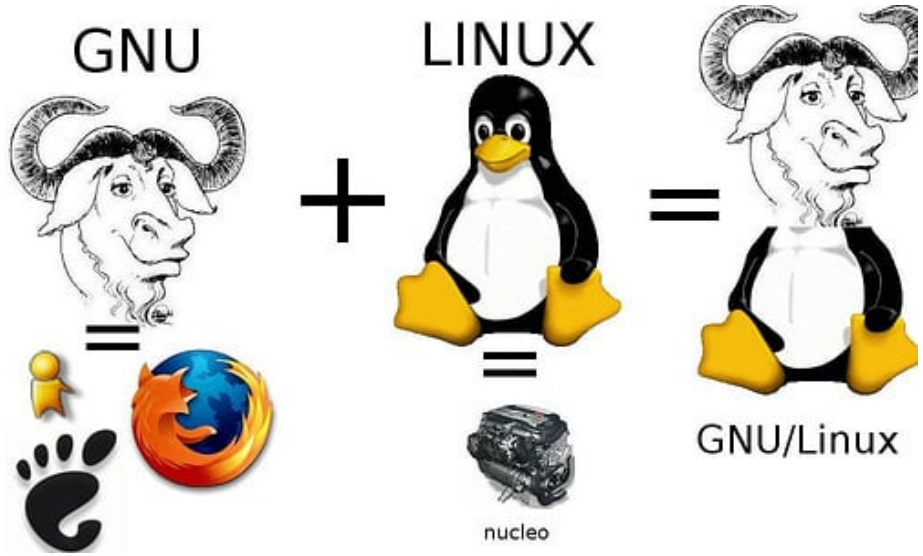
1st copyleft license

any derivative work must be distributed under the same terms.

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

GNU/Linux ('92-), shortly Linux



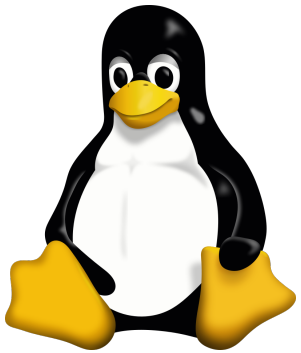


$\Rightarrow$

1. GNU utils: high-level utilities.
2. Kernel:
   - low-level 'stuff', written (mostly) in C, GPLv2,
   - manages the CPU, memory, device drivers, file system, ...

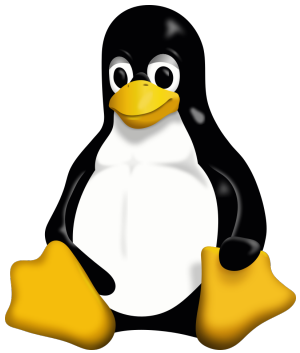
Linus+

- Tux (the mascot of the Linux kernel) := Torvalds UniX.



Linus+

- 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: $\sim$35 million (2023).

Free vs open

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

- goal: to respect user freedom and privacy.

to *not constrain* the user

Free vs open

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, video [OD,YT],
  - open-source code can 'spy' on you,

  privacy matters [OD,YT].

Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s [OD,YT]: \$12.9 $\ll$ \$1500 (repairing for 16Y@2024; $\sim$iPhone)

Free vs right to repair

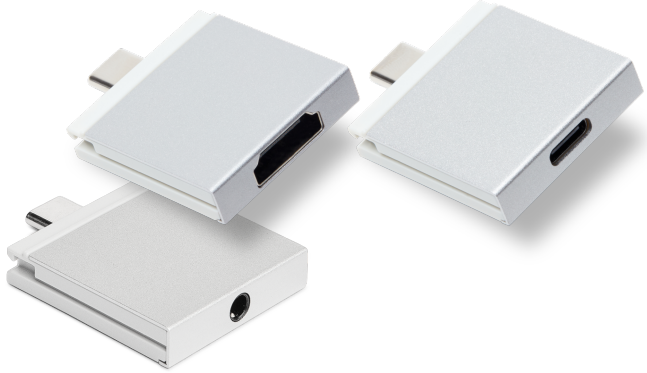
- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s [OD,YT]: \$12.9 ≪ \$1500 (repairing for 16Y@2024; ∼iPhone),
- Free future hints:
  - System76: repairable laptops, Launch keyboard, Pop!_OS, Virgo (design),

Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s [OD,YT]: \$12.9 ≪ \$1500 (repairing for 16Y@2024; ∼iPhone),
- Free future hints:
  - System76: repairable laptops, Launch keyboard, Pop!_OS, Virgo (design),
  - Tuxedo: customizable, repairable PCs.

Free vs right to repair

- my experience: battery replacement in Surface Pro = 600€,
- idea in 60s [OD,YT]: \$12.9 ≪ \$1500 (repairing for 16Y@2024; ∼iPhone),
- Free future hints:
  - System76: repairable laptops, Launch keyboard, Pop!_OS, Virgo (design),
  - Tuxedo: customizable, repairable PCs.
  - a laptop initiative: frame.work ≈ anti-MacBook ⇐ designed to be easy to upgrade & repair.



Linux: free $\Rightarrow$

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

Linux: free $\Rightarrow$

- community-driven, versatile, transparent, secure & private, modular, resource-efficient, sustainable.
- specifically:
  - no force to
  1. upgrade to the latest hardware

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)

Linux: free $\Rightarrow$

- 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. have your entire life history saved (e.g., Win 11: Recall $\in$ Copilot+ PCs [OD_1,YT_1; OD_2, YT_2; OD_3, YT_3]; Google & Apple $\xrightarrow{\text{push notifications}}$ governments [OD,YT], collect data even in incognito mode or with apps disabled)

Linux: free $\Rightarrow$

- 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. have your entire life history saved (e.g., Win 11: Recall $\in$ Copilot+ PCs [OD_1,YT_1; OD_2, YT_2; OD_3, YT_3]; Google & Apple $\xrightarrow{\text{push notifications}}$ governments [OD,YT], collect data even in incognito mode or with apps disabled),
4. create accounts or watch dummy ads on the UI

Linux: free $\Rightarrow$

- 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. have your entire life history saved (e.g., Win 11: Recall $\in$ Copilot+ PCs [OD$_1$,YT$_1$; OD$_2$, YT$_2$; OD$_3$, YT$_3$]; Google & Apple $\xrightarrow{\text{push notifications}}$ governments [OD,YT], collect data even in incognito mode or with apps disabled),
  4. create accounts or watch dummy ads on the UI,
  5. go with the trust us bro security/privacy guarantee.

Linux: free $\Rightarrow$

- specifically – continued:
  - significant storage reduction $\Leftarrow$ code sharing.

Linux: free $\Rightarrow$

- specifically – continued:
  - significant storage reduction $\Leftarrow$ code sharing.
  - no intentional slow down of your system (e.g., iPhone).

Linux: free $\Rightarrow$

- specifically – continued:
  - significant storage reduction $\Leftarrow$ code sharing.
  - no intentional slow down of your system (e.g., iPhone).
  - standard for computing clusters (example: slurm).

Versatility $\Rightarrow$

Various distros (tree):

- there have been $>$ 1000 distributions,
- currently (Sept. 26, 2024): 275 distributions

Versatility $\Rightarrow$

Various distros (tree):

- there have been $>$ 1000 distributions,
- currently (Sept. 26, 2024): 275 distributions,
- but minor differences.

Primary choice to make

point release vs rolling release.

Point release model: Windows

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

Point release model: Windows, 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, Sonoma, Sequoia.

Point release model: Windows, 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, Sonoma, Sequoia.
- Properties:
  1. occasional big changes,
  2. end-of-life date!

Point release: Linux distributions




(Debian $\rightarrow$) Ubuntu $\rightarrow$ Pop!_OS; Fedora; openSUSE.

Point release: Linux distributions





(Debian $\rightarrow$) Ubuntu $\rightarrow$ Pop!_OS; Fedora; openSUSE.

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

Point release: Linux distributions





(Debian $\rightarrow$) Ubuntu $\rightarrow$ Pop!_OS; Fedora; openSUSE.

- Ubuntu:
  - developer: Canonical Ltd.,
  - released every six months, LTS every 2 years.
- Pop!_OS:
  - $\approx$ Ubuntu with customized Gnome (COSMIC),
  - maintained by System76.

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.

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.

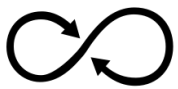
Rolling release: Linux distributions

- 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.



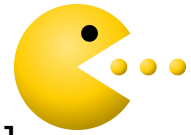
Tumbleweed

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)

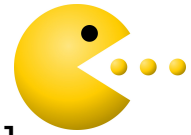
0. rolling release.
1. great package manager (pacman),
   - fast,
   - allows parallel downloading.



[Package managers handle dependencies.]

My choice: Arch (released in 2002)

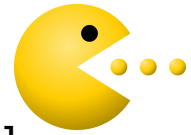
0. rolling release.
1. great package manager (pacman),
   - fast,
   - allows parallel downloading.

   [Package managers handle dependencies.]
2. superb documentation (Arch Wiki):
   - base reference even for other distributions;)
   - even offline readable [OD,YT] and searchable [OD,YT].

My choice: Arch (released in 2002)

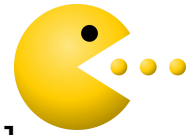
0. rolling release.
1. great package manager (pacman),
   - fast,
   - allows parallel downloading.

   [Package managers handle dependencies.]
2. superb documentation (Arch Wiki):
   - base reference even for other distributions;)
   - even offline readable [OD,YT] and searchable [OD,YT].
3. excellent software availability:
   - main: 14.75K

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 [OD,YT] and searchable [OD,YT].
3. excellent software availability:
   - main: 14.75K, AUR: 93.71K packages.

   Both are searchable.

Installation

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

Installation

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

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

Installation

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

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

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:
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

- .iso size:
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

- .iso size:
  - 5.8 GB (Ubuntu),
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

- .iso size:
  - 2.5 GB (Pop!_OS),
  - 5.8 GB (Ubuntu),
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

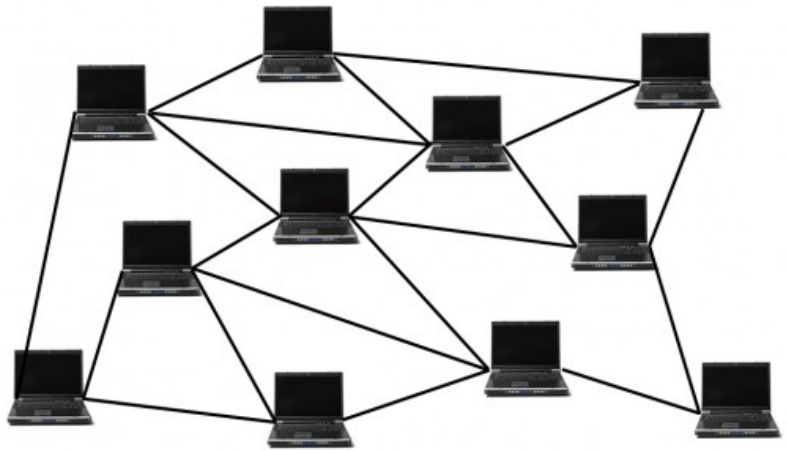
- .iso size:
  - 2.1 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 5.8 GB (Ubuntu),
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

- .iso size:
  - 1.1 GB (Arch),
  - 2.1 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 5.8 GB (Ubuntu),
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – for comparison.

Downloading note

- .iso size:
  - 1.1 GB (Arch),
  - 2.1 GB (Fedora),
  - 2.5 GB (Pop!_OS),
  - 5.8 GB (Ubuntu),
  - 6.3 GB (Windows 11) – for comparison.
  - 12.1 GB (MacOS Sequoia) – 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).

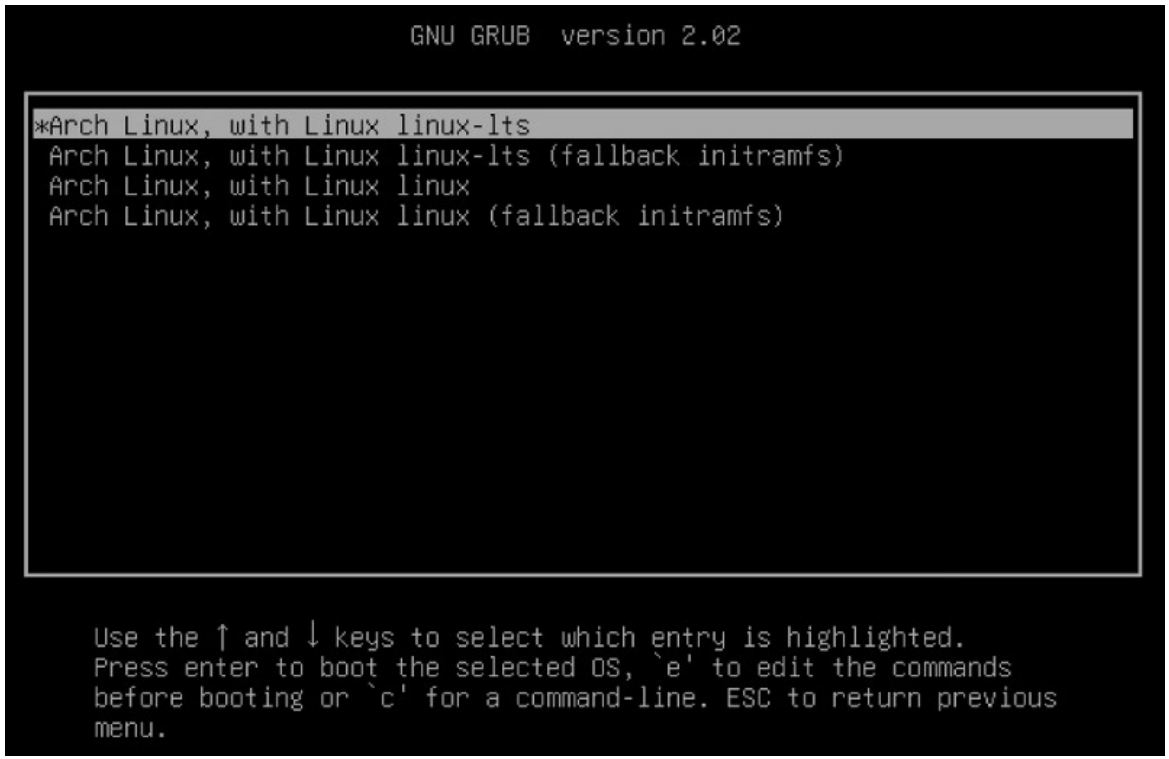
[†]no chat 😉, [‡]EFI System Partition.

Notes on the boot process — a bit technical

1. system's firmware (such as BIOS/UEFI/Coreboot/Libreboot) $\xrightarrow{\text{starts}}$
2. bootloader (such as GRUB $\Leftarrow$ GNU; features & others) $\xrightarrow{\text{loads}}$
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:
   - vid_1 [OD,YT]: UEFI; check the YouTube comments as well!
   - vid_2 [YT]: BIOS, UEFI, UEFI-LVM-LUKS.

Instructions: for Arch – scary;)

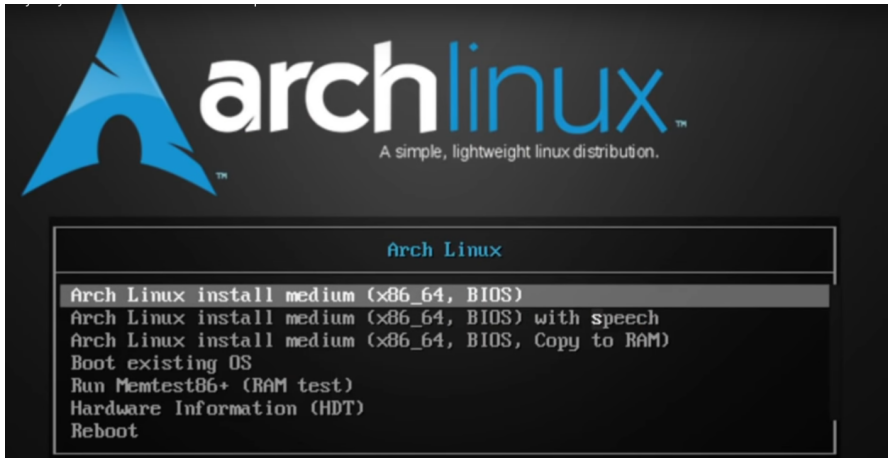
1. Step-by-step text guide (official one).
2. Video guide:
   - vid_1 [OD,YT]: UEFI; check the YouTube comments as well!
   - vid_2 [YT]: BIOS, UEFI, UEFI-LVM-LUKS.

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

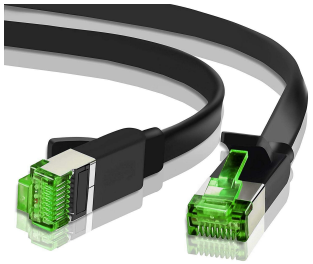
BIOS:



UEFI:

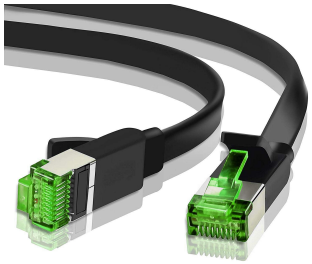
Installation hints

1. use ethernet: faster.

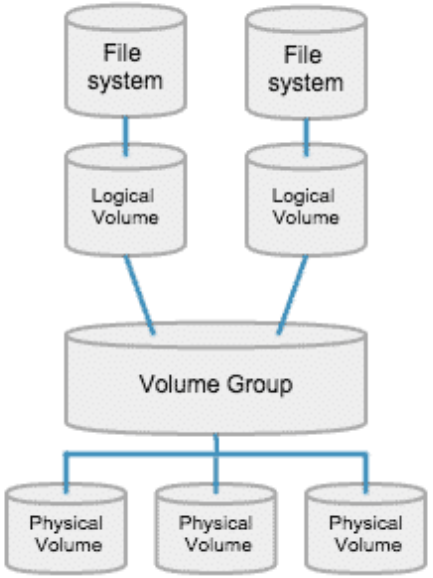


Installation hints

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.
   4. NixOS:
      - declarative approach based on Nix [YT] $\Rightarrow$ reproducibility,
      - can be pretty neat (in the future).

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.
   4. NixOS:
      - declarative approach based on Nix [YT] $\Rightarrow$ reproducibility,
      - can be pretty neat (in the future).

†Start with a DE before a WM.

Installation hints – continued

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

   - btrfs:
     - modern alternative,
     - supports compression $\Rightarrow$ less space, increased storage lifespan,
     - copy-on-write $\Rightarrow$ consistency even in case of power loss,
     - snapshot feature,
     - limited LUKS support.

     $\Rightarrow$ 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, $\times 1000$!); 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 ($\sim$ 11s):
   - SSD matters: Samsung 970 EVO Plus $\leftarrow$ my choice (for laptop).

Installation hints – continued

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

7. Use a spare drive (to avoid the wrestling of the op. systems),
   - example: (used) Thinkpad $\rightarrow$ T480 [YT]: low-budget, flexible.

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; $\in$ 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

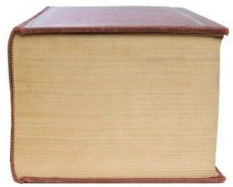
Notations: M = '$\in$ main', A = '$\in$ AUR', W = web client, ✓ = installed by default, $_p$= proprietary.

- Web & mail:
  - browser:
    - firefox (M), tor-browser-bin (A)

Applications: categorized; some handy ones

Notations: M = '$\in$ main', A = '$\in$ AUR', W = web client, ✓ = installed by default, $_p$= proprietary.

- Web & mail:
  - browser:
    - firefox (M), tor-browser-bin (A),
    - Chromium-based browsers: warning! $\Leftarrow$ Manifest v3 [OD,YT].



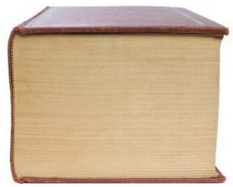
The Internet

The Internet when you block ads

Applications: categorized; some handy ones

Notations: M = '$\in$ main', A = '$\in$ AUR', W = web client, ✓ = installed by default, $_p$= proprietary.

- Web & mail:
  - browser:
    - firefox (M), tor-browser-bin (A),
    - Chromium-based browsers: warning! $\Leftarrow$ Manifest v3 [OD,YT].



The Internet

The Internet when you block ads

  - e-mail: ProtonMail (W), thunderbird (M).

ProtonMail: use this referral link to

get a free month on the Mail Plus plan!

Applications+

Media:

- image:
  - viewer: feh (M), gthumb (M), geeqie (M).
  - editor: gimp (M; bitmap), inkscape (M; vector).

Applications+

Media:

- image:
  - viewer: feh (M), gthumb (M), geeqie (M).
  - editor: gimp (M; bitmap), inkscape (M; vector).
- video (& audio):
  - player: mplayer (M), vlc (M). +: mpv (M), celluloid (M).
  - editor: kdenlive (M).
  - recording, live streaming: obs-studio (M).
  - downloader: yt-dlp (M).

Applications+

Media:

- image:
  - viewer: feh (M), gthumb (M), geeqie (M).
  - editor: gimp (M; bitmap), inkscape (M; vector).
- video (& audio):
  - player: mplayer (M), vlc (M). +: mpv (M), celluloid (M).
  - editor: kdenlive (M).
  - recording, live streaming: obs-studio (M).
  - downloader: yt-dlp (M).
- audio+:
  - spotify:
    - player: spotify_p(A),
    - downloader: spotdl (A).
  - editor: tenacity (M).

Applications+

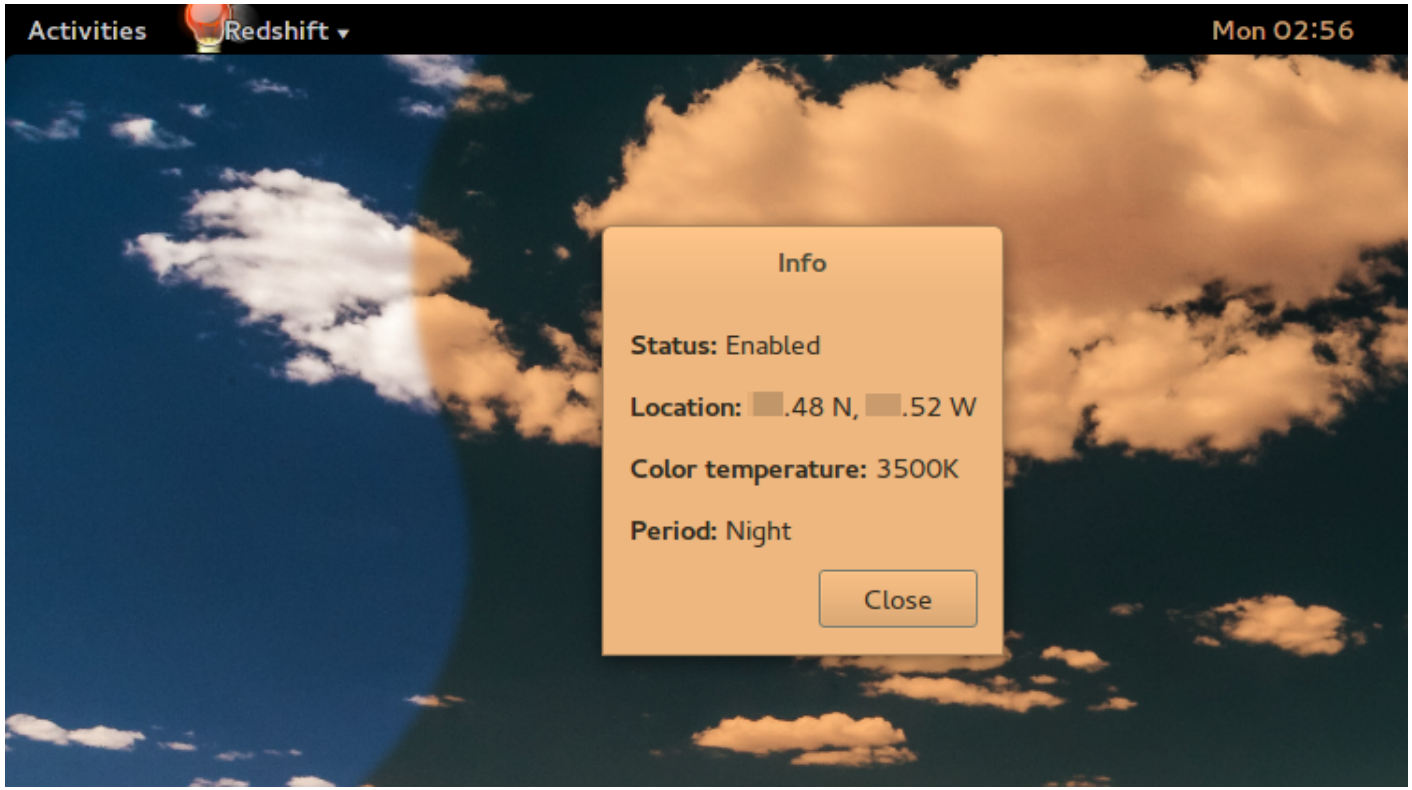
- Text:
  - document viewer: xdvi (✓, $\in$ texlive), xpdf (M), evince (M),
  - .pdf annotation: xournalpp (M),
  - text editing: texlive (LaTeX, M, group), kile (M) ($\approx$ WinEdt$_p$), vim (M), libreoffice-still (M), notepadqq (M).

Applications+

- Text:
  - document viewer: xdvi (✓, $\in$ texlive), xpdf (M), evince (M),
  - .pdf annotation: xournalpp (M),
  - text editing: texlive (LaTeX, M, group), kile (M) ($\approx$ WinEdt$_p$), vim (M), libreoffice-still (M), notepadqq (M).
- Chat & collaboration:
  - chat: skypeforlinux-bin$_p$(A), zoom$_p$(A), teams$_p$(A), BigBlueButton (W), Jitsi Meet (W),
  - 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),
- password 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),
- 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),
- remote access:
  - remote support: teamviewer$_p$(A),
  - VNC client & server: tigervnc (M),
  - SFTP client: filezilla (M), remmina (M; VNC client too)

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),
- remote access:
  - remote support: teamviewer$_p$(A),
  - VNC client & server: tigervnc (M),
  - SFTP client: filezilla (M), remmina (M; VNC client too).
- 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),
- remote access:
  - remote support: teamviewer$_p$(A),
  - VNC client & server: tigervnc (M),
  - SFTP client: filezilla (M), remmina (M; VNC client too).
- 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),
- password manager: keepassxc (M),
- RSS/Atom feed reader: newsboat (M),
- screen locker: slock (M),
- remote access:
  - remote support: teamviewer$_p$(A),
  - VNC client & server: tigervnc (M),
  - SFTP client: filezilla (M), remmina (M; VNC client too).
- terminal: tilix (M),
- firewall: ufw (M),
- desktop environment: gnome (M, group), plasma (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)

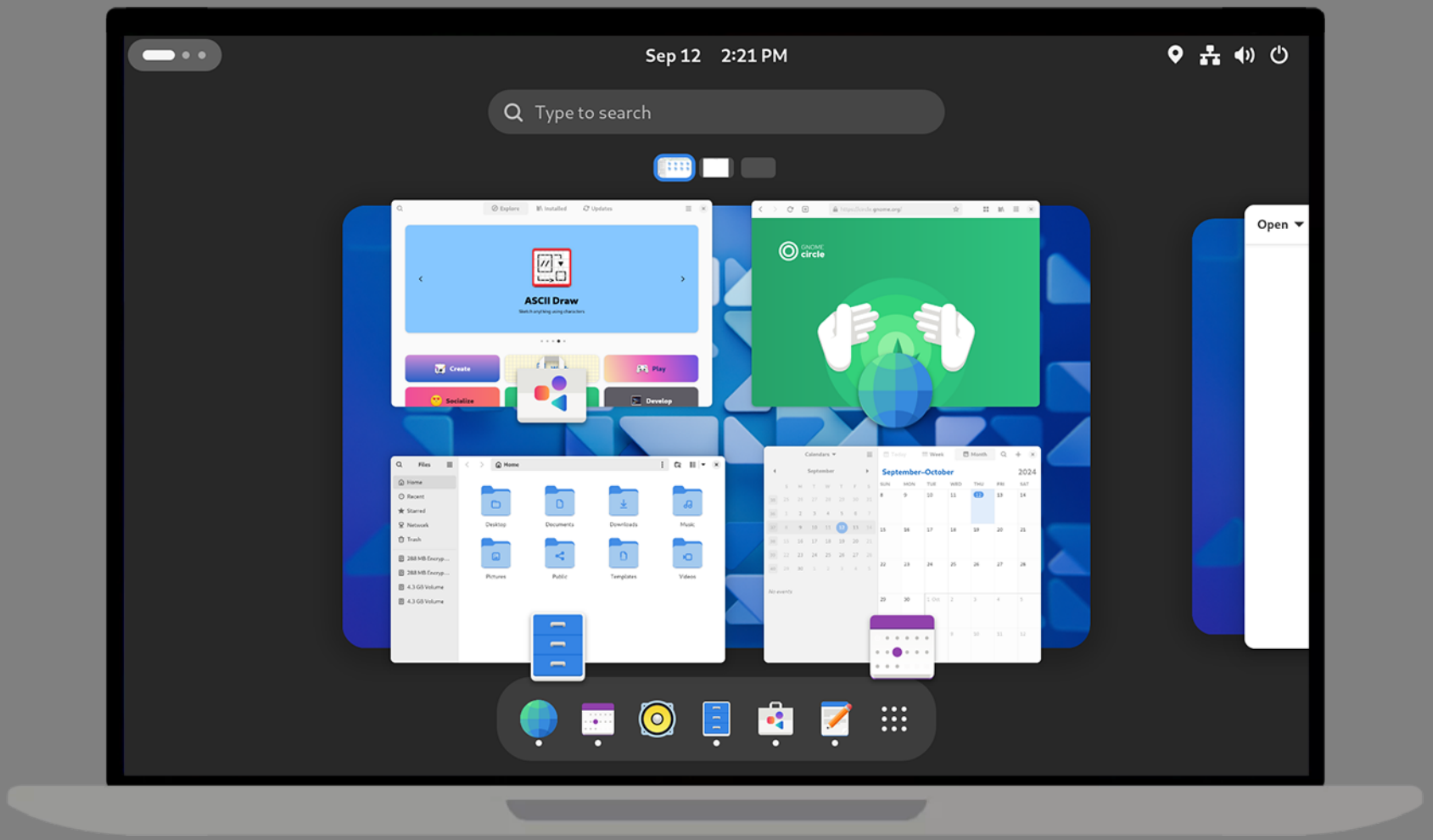
- Desktop environments:
  - windows manager, and
  - a bundle of applications (calendar, image viewer, file manager, ...).
- You
  - are free to choose it, and can have multiple ones,
  - get workspaces.
- Some popular choices: GNOME, KDE Plasma, Xfce, Enlightenment.

Desktop environments (DE)

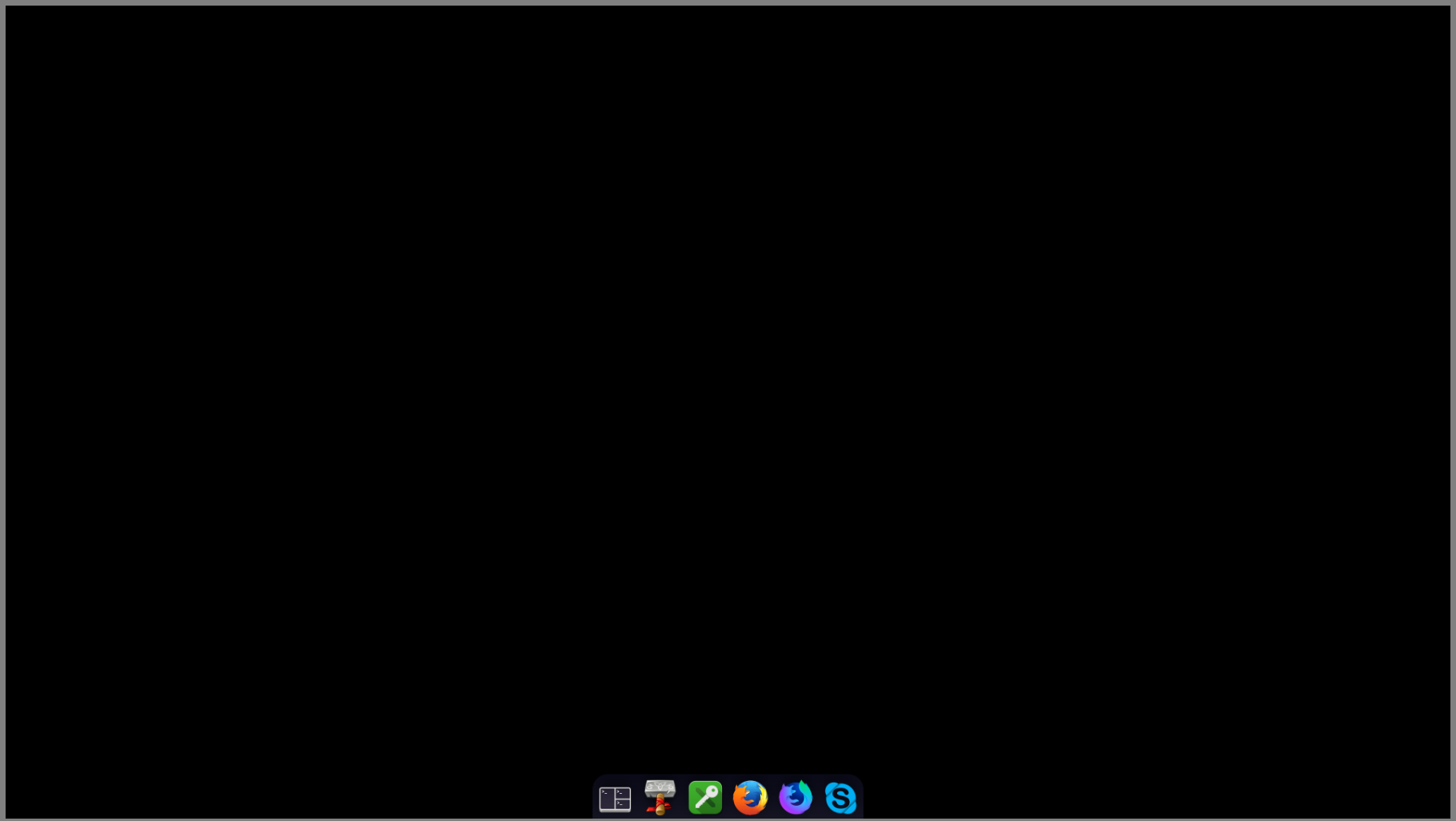
- Desktop environments:
  - windows manager, and
  - a bundle of applications (calendar, image viewer, file manager, ...).
- You
  - are free to choose it, and can have multiple ones,
  - get workspaces.
- Some popular choices: GNOME, KDE Plasma, Xfce, Enlightenment.

Examples follow

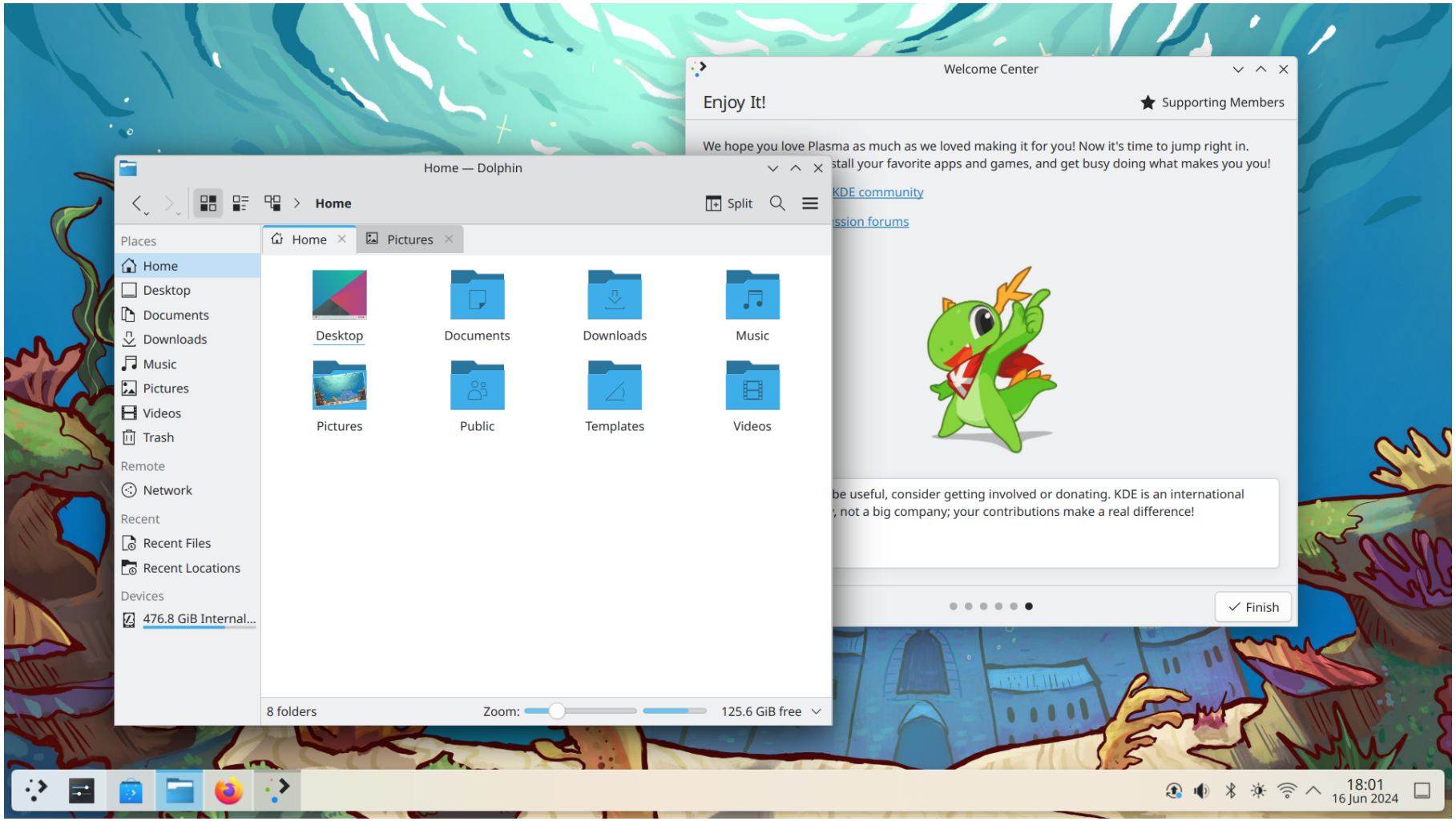
DE: GNOME


Sep 12 2:21 PM
Type to search
ASCII Draw
Create
Play
Socialize
Develop
GNOME circle
Open
September–October
2024

DE: GNOME – my ex-desktop

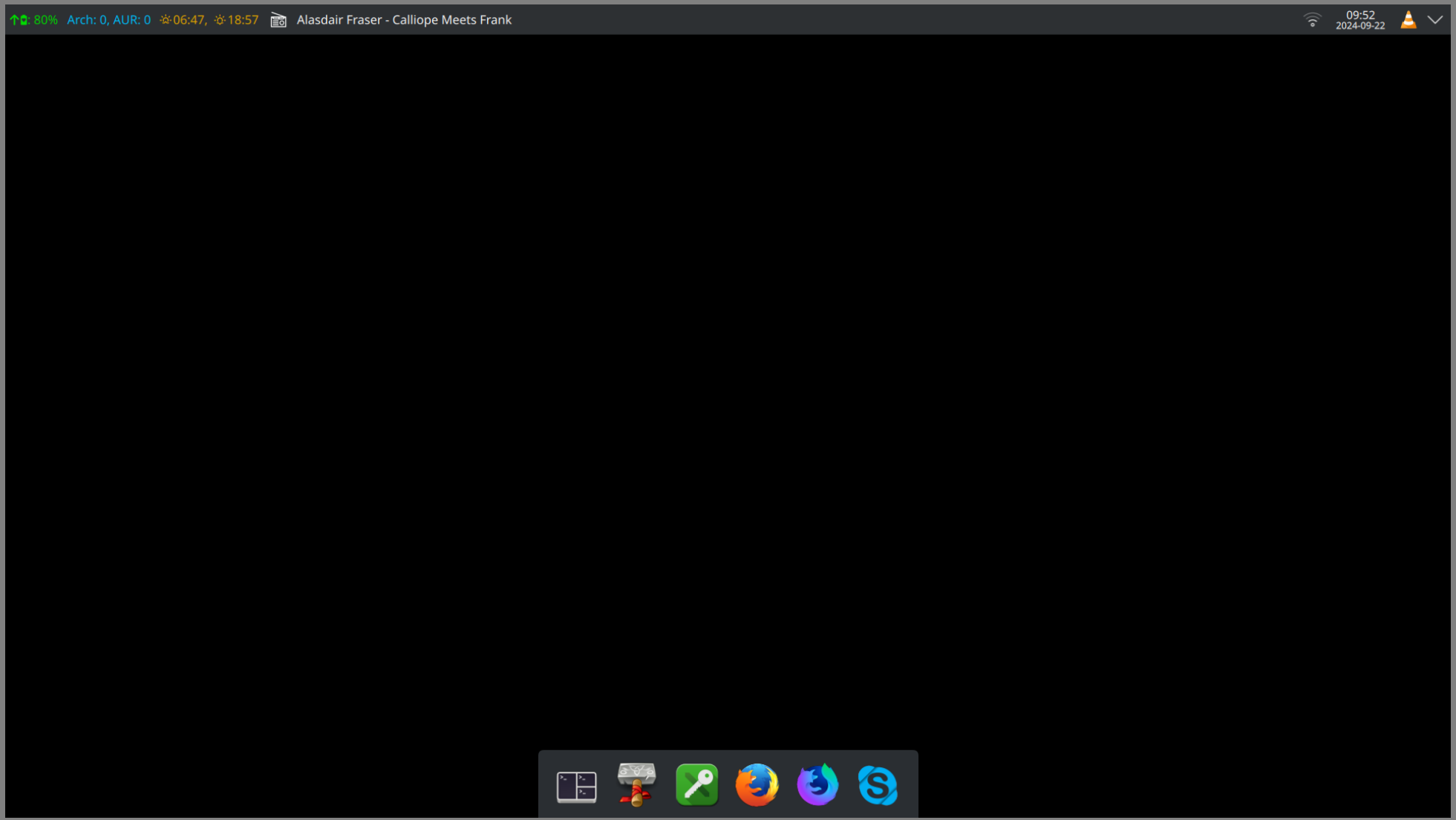


DE: KDE Plasma

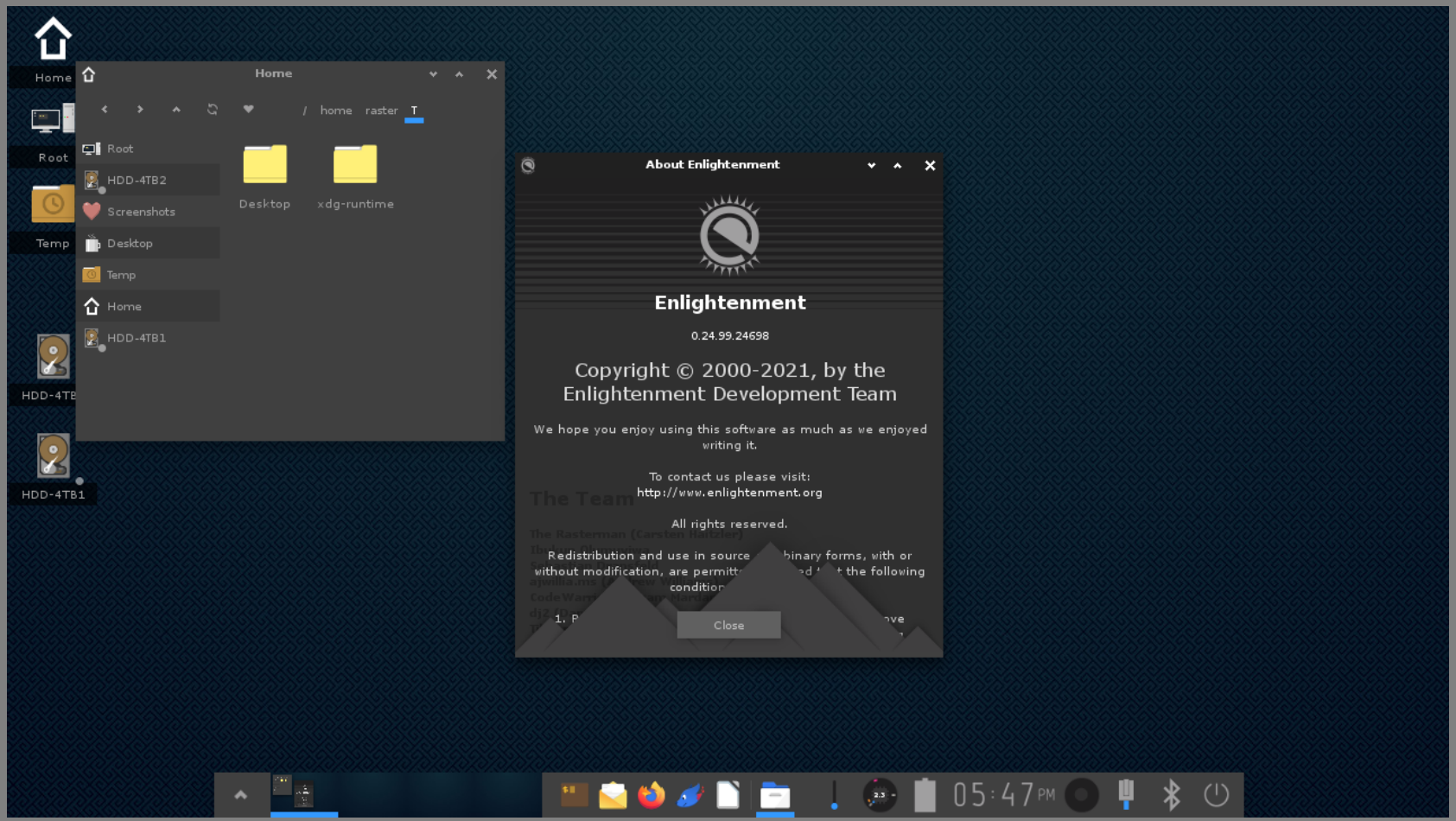


$\xrightarrow{\text{spec.}}$ Win XP [YT] & Win 11 mimicing [OD, YT].

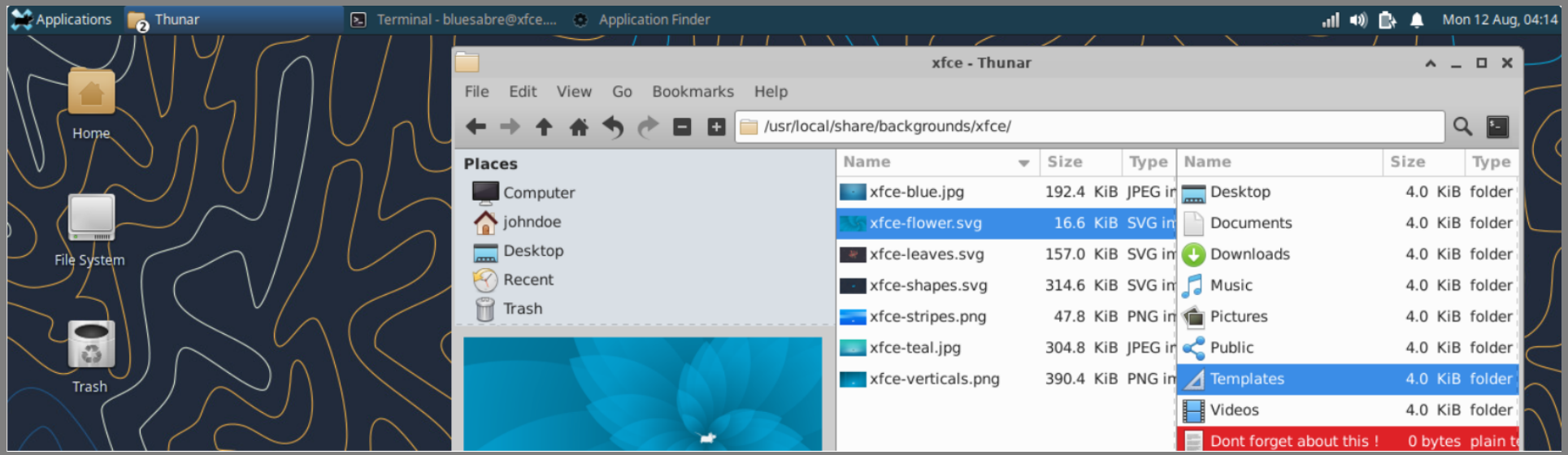
DE: KDE Plasma – my desktop



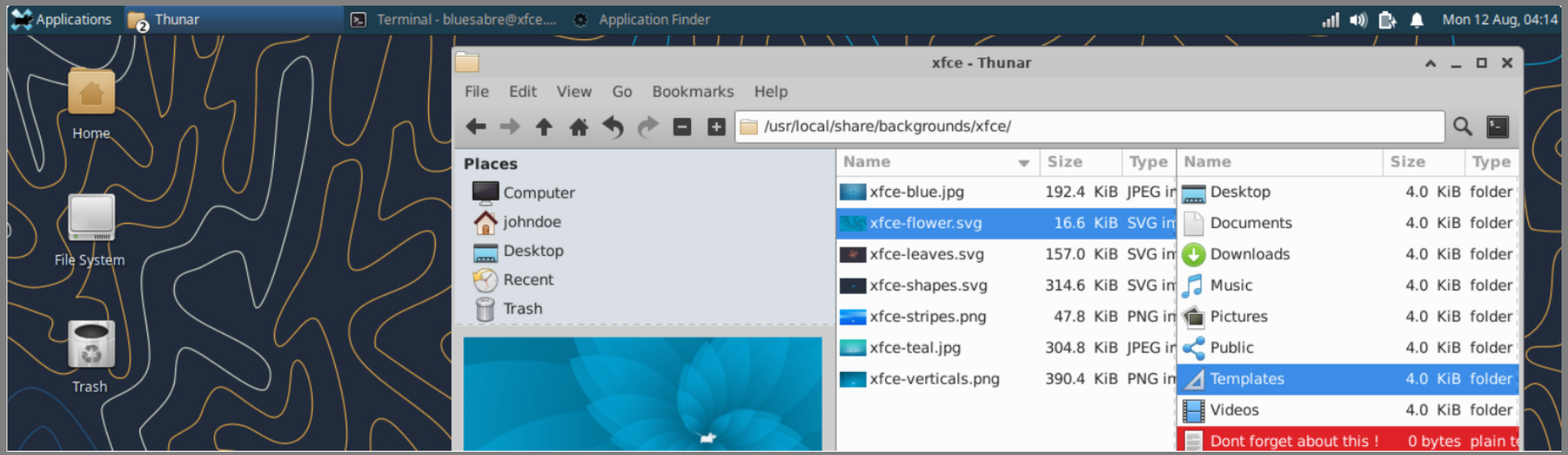
DE: Enlightenment



DE: Xfce


Applications
Thunar
Terminal - bluesabre@xfce....
Application Finder
Mon 12 Aug, 04:14
Home
File System
Trash
xfce - Thunar
File Edit View Go Bookmarks Help
/usr/local/share/backgrounds/xfce/
Places
Computer
johndoe
Desktop
Recent
Trash
Name Size Type
xfce-blue.jpg 192.4 KiB JPEG in
xfce-flower.svg 16.6 KiB SVG in
xfce-leaves.svg 157.0 KiB SVG in
xfce-shapes.svg 314.6 KiB SVG in
xfce-stripes.png 47.8 KiB PNG in
xfce-teal.jpg 304.8 KiB JPEG in
xfce-verticals.png 390.4 KiB PNG in
Name Size Type
Desktop 4.0 KiB folder
Documents 4.0 KiB folder
Downloads 4.0 KiB folder
Music 4.0 KiB folder
Pictures 4.0 KiB folder
Public 4.0 KiB folder
Templates 4.0 KiB folder
Videos 4.0 KiB folder
Dont forget about this ! 0 bytes

DE: Xfce



Extra DE inspiration

- unixporn (screenshots),
- COSMIC DE: worth keeping an eye on it! (Wayland-based)

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.

WM $\rightarrow$ DE

- Stacking:
  - Mutter $\rightarrow$ GNOME,
  - KWin $\rightarrow$ KDE,
  - Xfwm $\rightarrow$ Xfce,
  - Enlightenment $\rightarrow$ Enlightenment.

WM $\rightarrow$ DE

- Stacking:
  - Mutter $\rightarrow$ GNOME,
  - KWin $\rightarrow$ KDE,
  - Xfwm $\rightarrow$ Xfce,
  - Enlightenment $\rightarrow$ Enlightenment.
- Tiling:
  - i3.

WM → DE

- Stacking:
  - Mutter → GNOME,
  - KWin → KDE,
  - Xfwm → Xfce,
  - Enlightenment → Enlightenment.
- Tiling:
  - i3.
- Dynamic:
  - Qtile:
    - it uses Python;)
    - various (73) widgets.

WM → DE

- Stacking:
  - Mutter → GNOME,
  - KWin → KDE,
  - Xfwm → Xfce,
  - Enlightenment → Enlightenment.
- Tiling:
  - i3.
- Dynamic:
  - Qtile:
    - it uses Python;)
    - various (73) widgets.

Examples follow

WM: i3

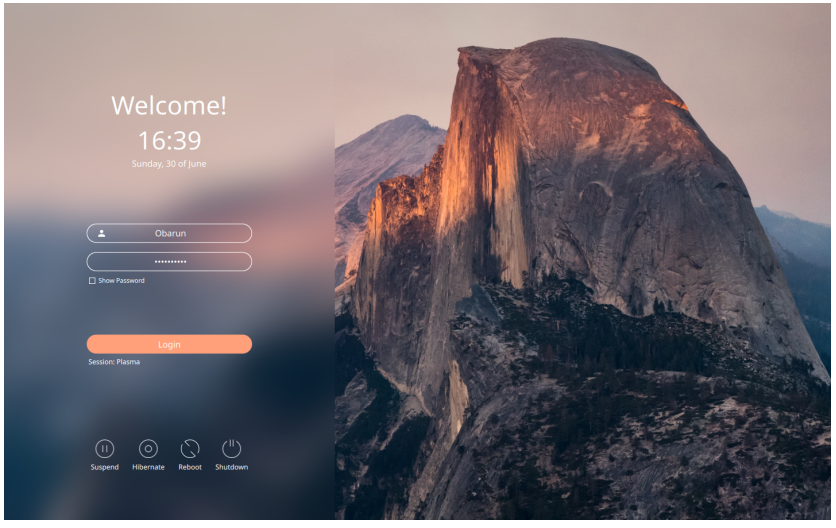

main.c (~/i3/src) - VIM
}
/* Set up i3 specific atoms like I3_SOCKET_PATH and I3_CONFIG_PATH */
x_set_i3_atoms();
struct ev_io *xcb_watcher = scalloc(sizeof(struct ev_io));
struct ev_io *xkb = scalloc(sizeof(struct ev_io));
struct ev_check *xcb_check = scalloc(sizeof(struct ev_check));
struct ev_prepare *xcb_prepare = scalloc(sizeof(struct ev_prepare));
ev_io_init(xcb_watcher, xcb_got_event, xcb_get_file_descriptor(conn), EV_READ);
ev_io_start(main_loop, xcb_watcher);
if (xkb_supported) {
ev_io_init(xkb, xkb_got_event, ConnectionNumber(xkbdpy), EV_READ);
ev_io_start(main_loop, xkb);
/* Flush the buffer so that libev can properly get new events */
XFlush(xkbdpy);
}
ev_check_init(xcb_check, xcb_check_cb);
ev_check_start(main_loop, xcb_check);
ev_prepare_init(xcb_prepare, xcb_prepare_cb);
ev_prepare_start(main_loop, xcb_prepare);
xcb_flush(conn);
manage_existing_windows(root);
if (!disable_signalhandler)
setup_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_loop(main_loop, 0);
src/main.c 464,9 99%
x200: mplayer S02/E05.avi
MPlayer
x200: git log
commit b87742172b3802badb615418f45d8556bfcf149b
Author: Michael Stapelberg <michael@stapelberg.de>
Date: Sun Jul 17 22:18:00 2011 +0200
Add missing function prototype for strndup on Darwin (Thanks Marcus)
commit 3da3a691063f7c4c7f09461bbe948fa48abeb5fd
Author: Michael Stapelberg <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 Mar
commit 02dfb8e891a759ed2152918778f6d0a928a71cac
Author: Michael Stapelberg <michael@stapelberg.de>
Date: Sun Jul 17 15:18:45 2011 +0200
use memmem and strndup from FreeBSD on Darwin (Thanks Marcus)
commit fc583adb9956ff95e9019bef8bb0e7c2e21d133b
Author: Michael Stapelberg <michael@stapelberg.de>
Date: Sun Jul 17 15:17:24 2011 +0200
makefile: link -liconv on Darwin (Thanks Marcus)
commit 7512f633a79b290f4e60287fb2b1cd689f025b42
Author: Michael Stapelberg <michael@stapelberg.de>
Date: Fri Jul 15 19:21:39 2011 +0200
:
1 2 3 4
2001:4d88:100e:23:21f:16ff:fe1a:f5b8 | 2,3 GB | DHCP: no | VPN: no | W: down | E: 192.168.1.42 (1000 Mbit/s) | BAT 74,15% [@] | 01% | 0,03 | 2011-07-22 15:02:32

WM: Qtile


monadtall
45.1%
739M/15419M
83.0°C
40G 10%
77%
?100%
1
14-11-2020 13:58
SUBSCRIBE

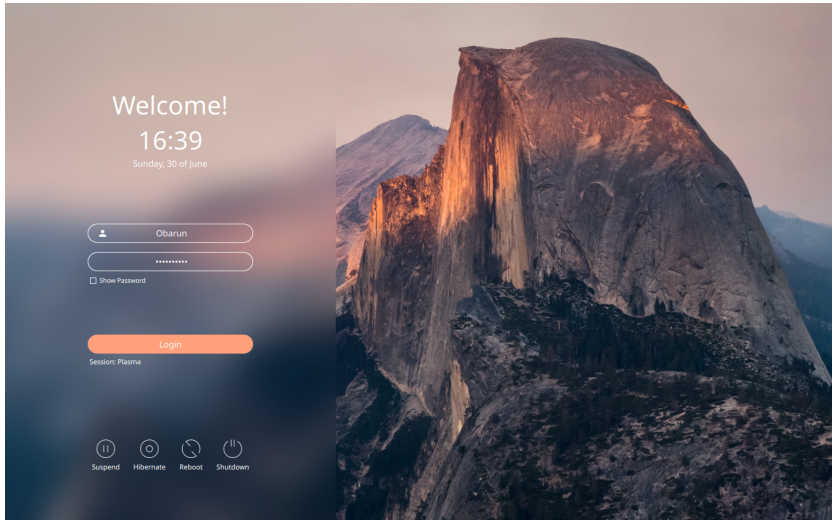
Login/display manager

- It gives graphical login – if you prefer not using/starting from tty;)
- Popular choices: (i) SDDM: tutorial [OD,YT] (inspiration+)



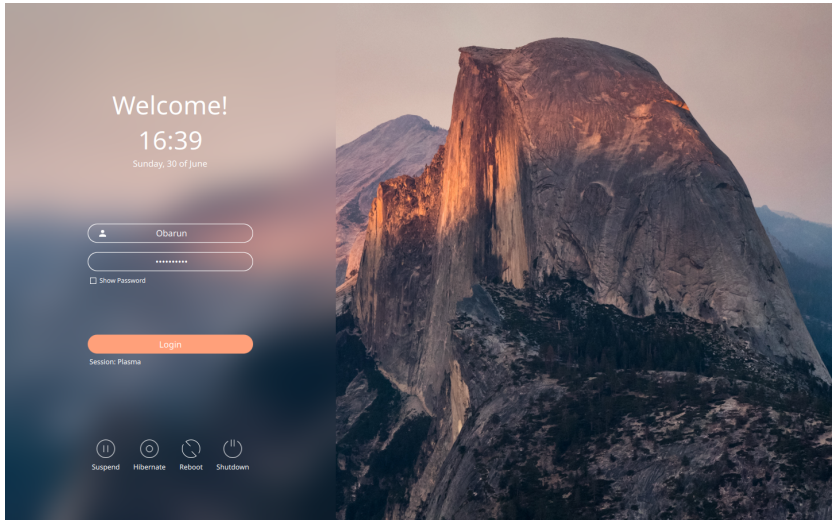
Login/display manager

- It gives graphical login – if you prefer not using/starting from tty;)
- Popular choices: (i) SDDM: tutorial [OD,YT] (inspiration+), (ii) Ly: tutorial [OD,YT]



Login/display manager

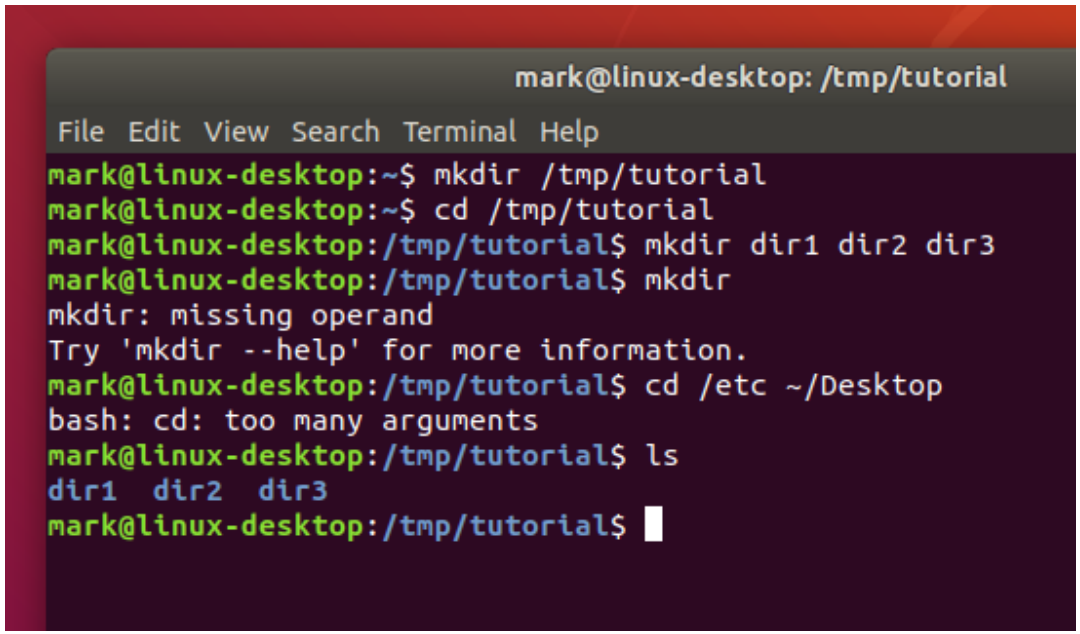
- It gives graphical login – if you prefer not using/starting from tty;)
- Popular choices: (i) SDDM: tutorial [OD,YT] (inspiration+), (ii) Ly: tutorial [OD,YT], (iii) GDM: tutorial [OD,YT] (inspiration+).



Terminal: used for instance @ Arch install

Command shell:

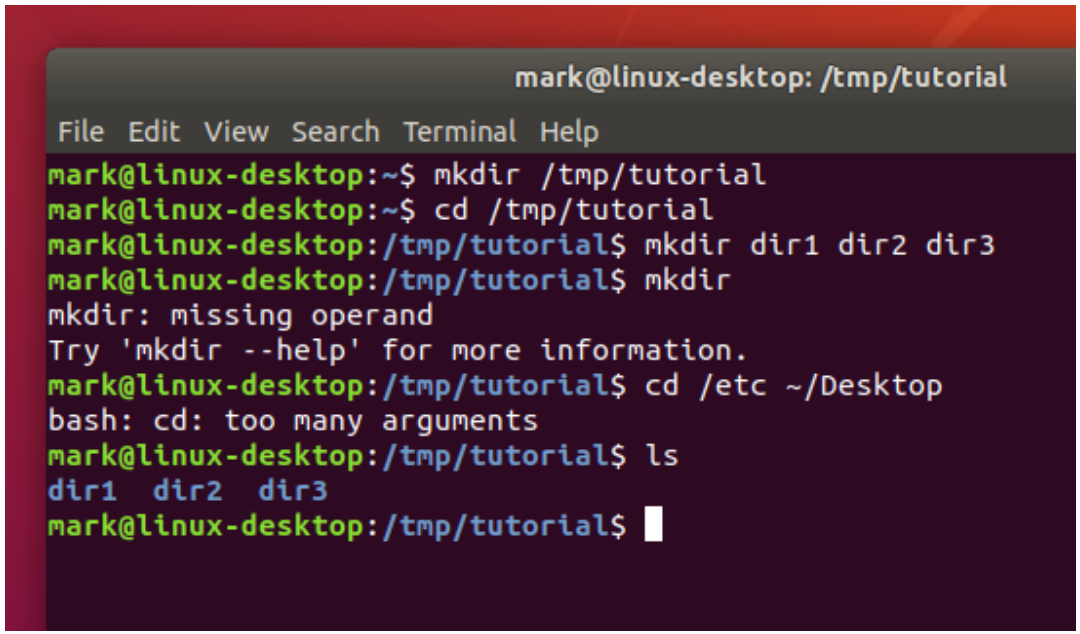
- like Jupyter notebook,
- interaction with the operating system,



Terminal: used for instance @ Arch install

Command shell:

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

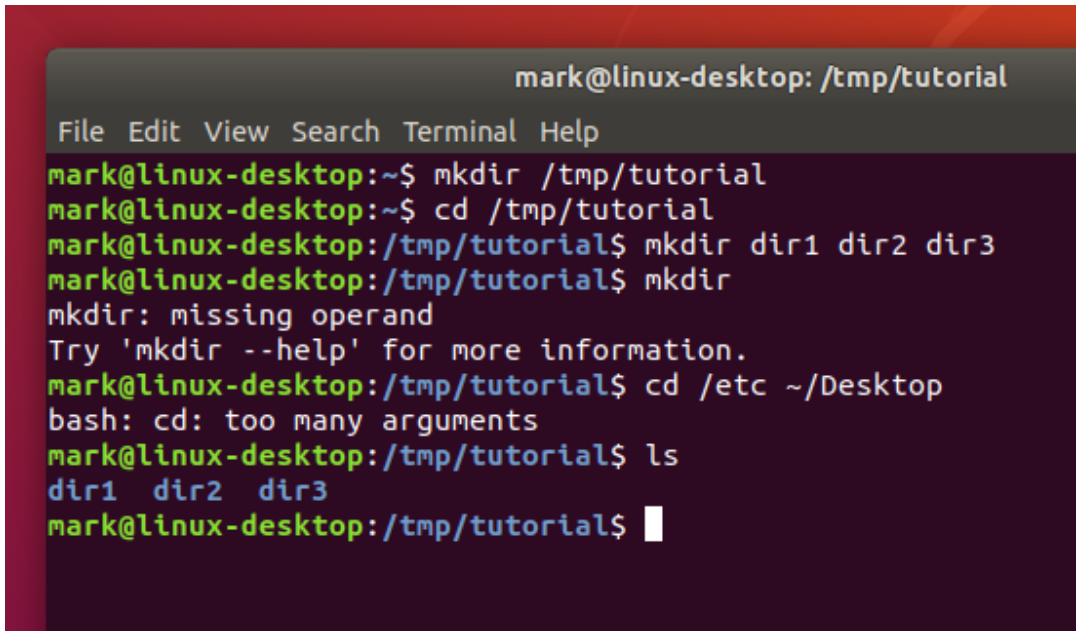


- Google Colab: `!shellcommand`

Terminal: used for instance @ Arch install

Command shell:

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



- Google Colab: `!shellcommand`,
- virtual console = text terminal + login prompt (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

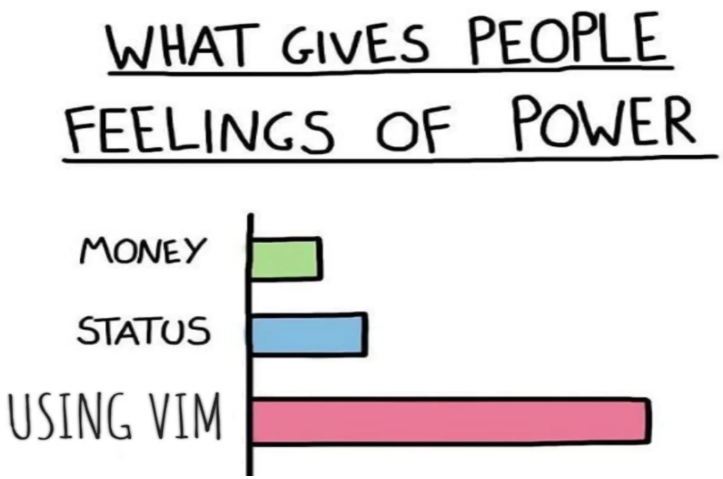
- Examples:
  - `$ cd` : change the current working directory,
  - `$ ls` : list directory content,
  - `$ pwd` : print the name of the current directory,
  - `$ cp` : copy files & directories,
  - `$ mv` : move or rename files and directories,
  - `$ touch` : create file,
  - `$ mkdir` : create directory,
  - `$ 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
  $\Rightarrow$
  - 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.
- cross-platform.



Vim: usage example

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

Vim: usage example

:q	:	quit $\leftarrow$ most difficult;),
:w	:	write (save),
h,j,k,l	:	arrows (left, down, up, right),
dw	:	delete word

Vim: usage example

:q	:	quit $\leftarrow$ most difficult;),
:w	:	write (save),
h,j,k,l	:	arrows (left, down, up, right),
dw	:	delete word,
cw	:	change word

Vim: usage 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

Vim: usage 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,
di(	:	delete inside parentheses

Vim: usage 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,
di(	:	delete inside parentheses,
dd	:	delete line

Vim: usage 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,
di(	:	delete inside parentheses,
dd	:	delete line,
p	:	paste

Vim: usage example

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

Vim: usage example

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

Vim – continued (free ⇒)

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

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

Name	Vids
ThePrimeagen	YT_1, YT_2, YT_3, YT_4, YT_5, YT_6
Missing Semester	OD, YT
DistroTube	OD_1, YT_1; OD_2, YT_2
Ben Awad	YT

Hint: How to ask on forums?

- Netiquette [OD,YT]; 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

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

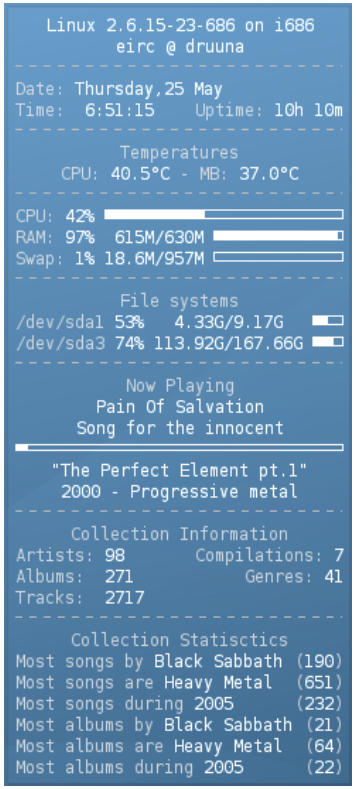
† but her/his past videos are nice.

Ricing: Conky

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

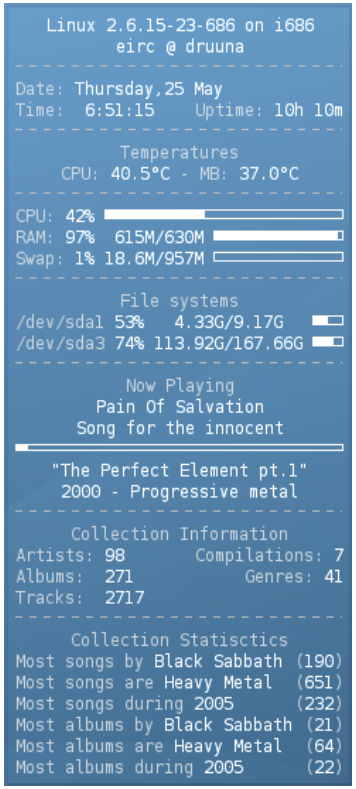
Ricing: Conky

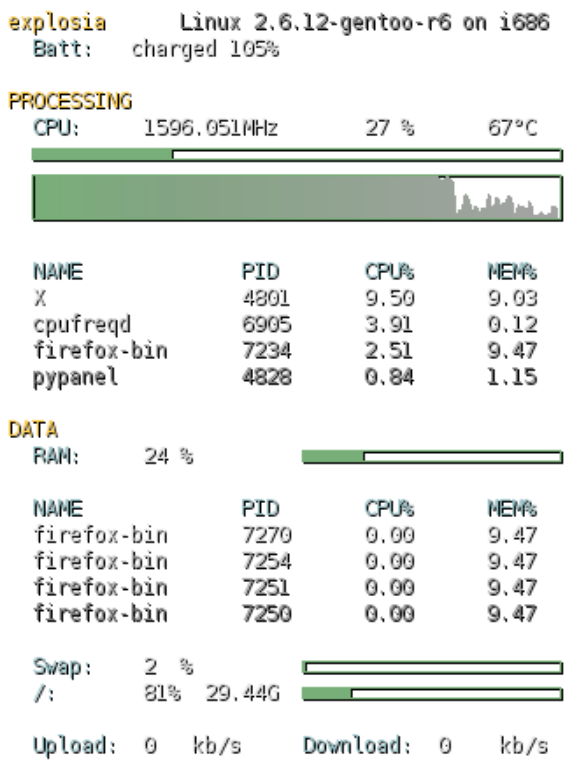
- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...
- Example (inspiration+):



Ricing: Conky

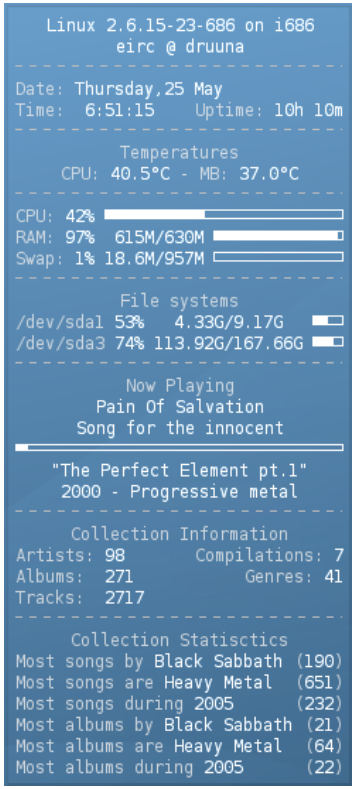
- system monitor: CPU, memory, swap space, disk storage, temperature, processes, network interfaces, battery power, system messages, e-mail, ...
- Example (inspiration+):

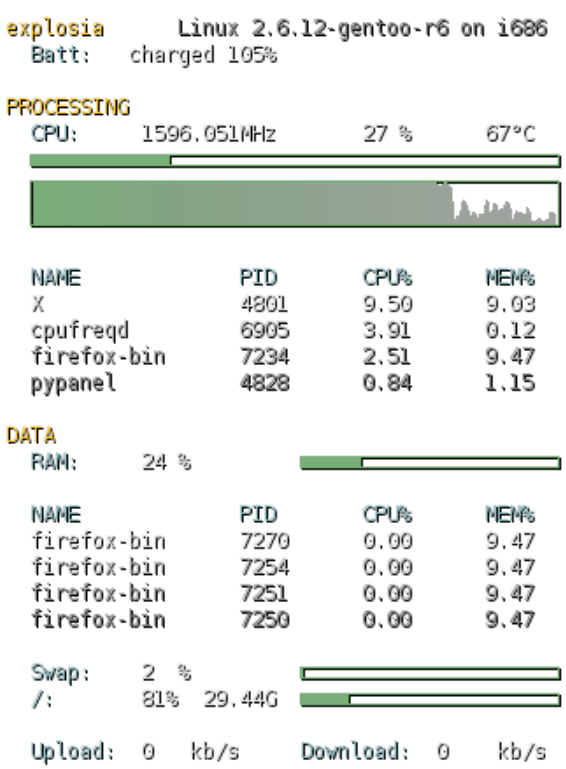




Ricing: Conky

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

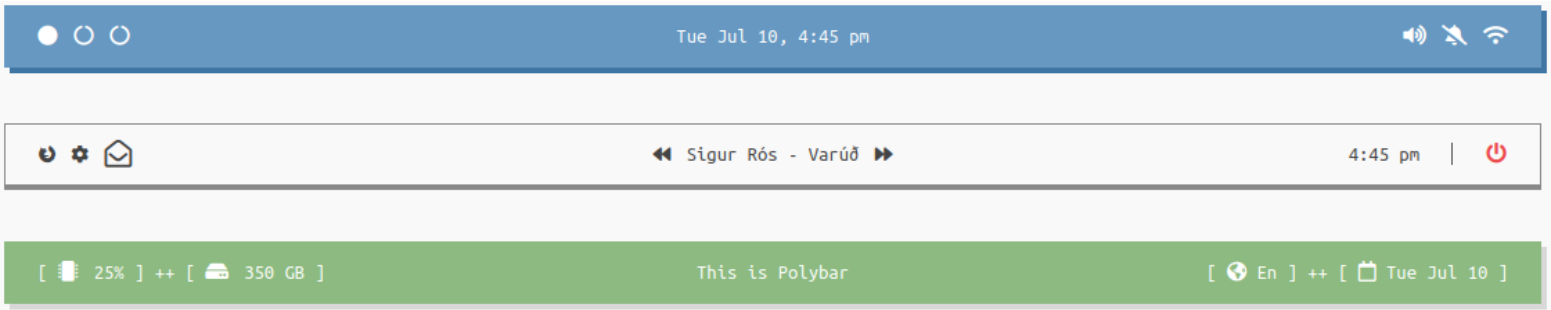




Ricing: Conky on desktop

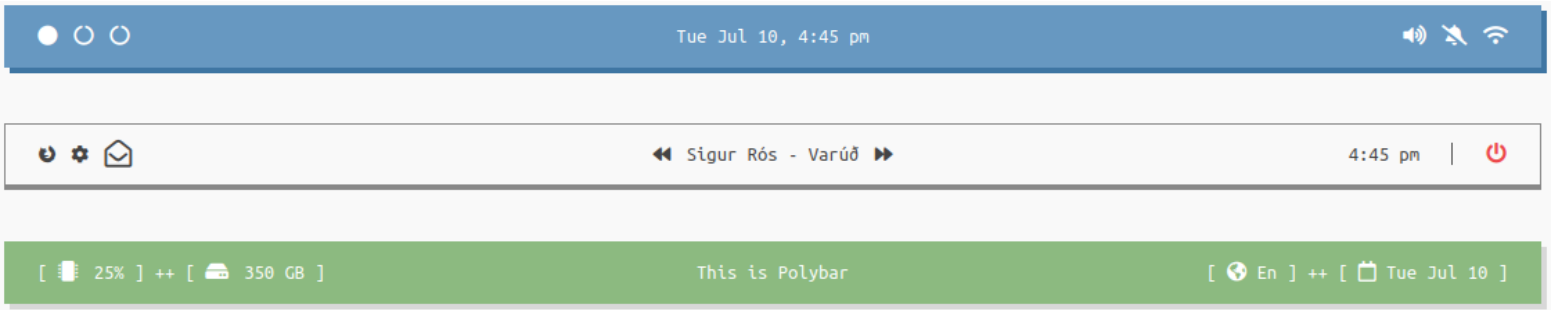
Ricing: Conky on desktop

Ricing: Polybar ($link_1$, $link_2$)



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

Ricing: Polybar ($link_1$, $link_2$)



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

Example follows.

Polybar example

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

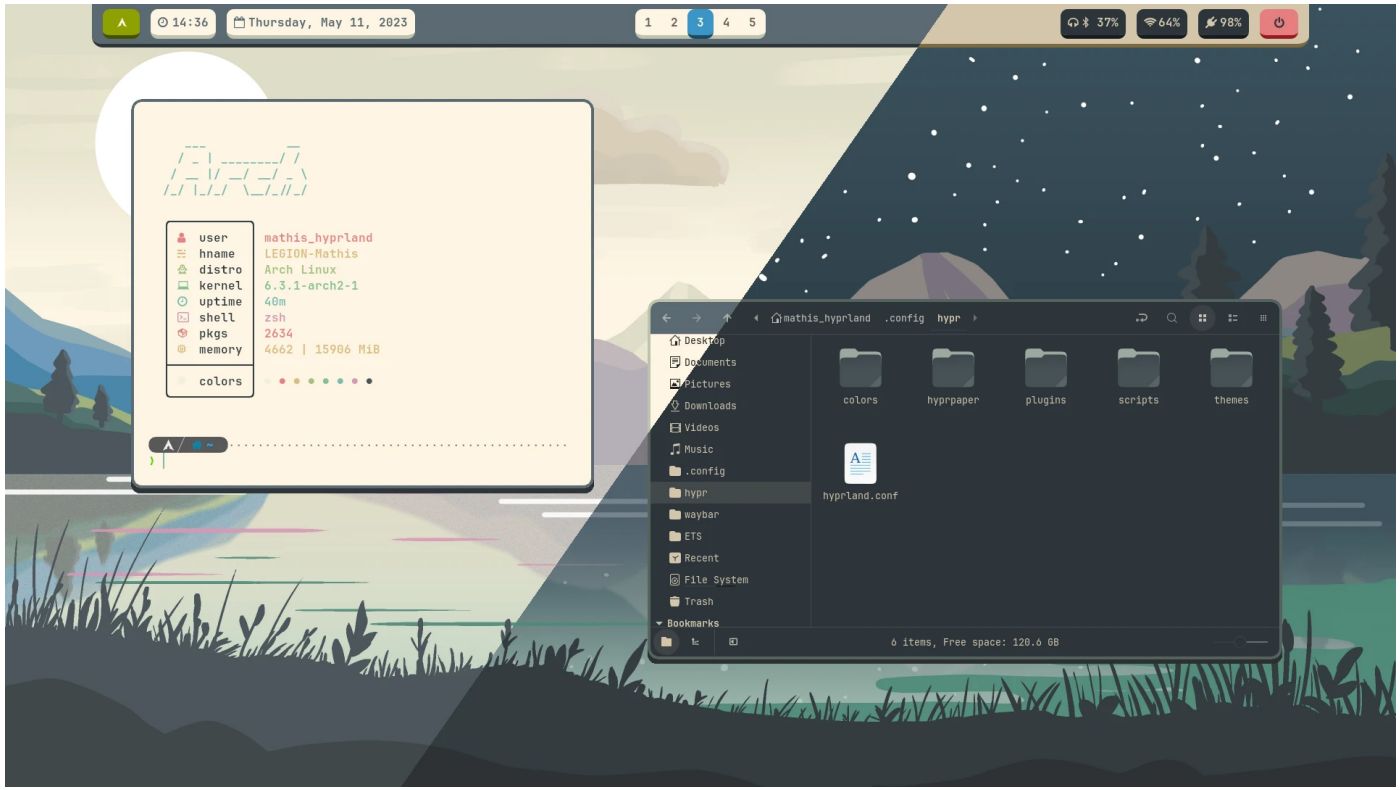
- They can
  - add effects like transparency, animations or blur,
  - be standalone or built into the DE / WM.

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

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

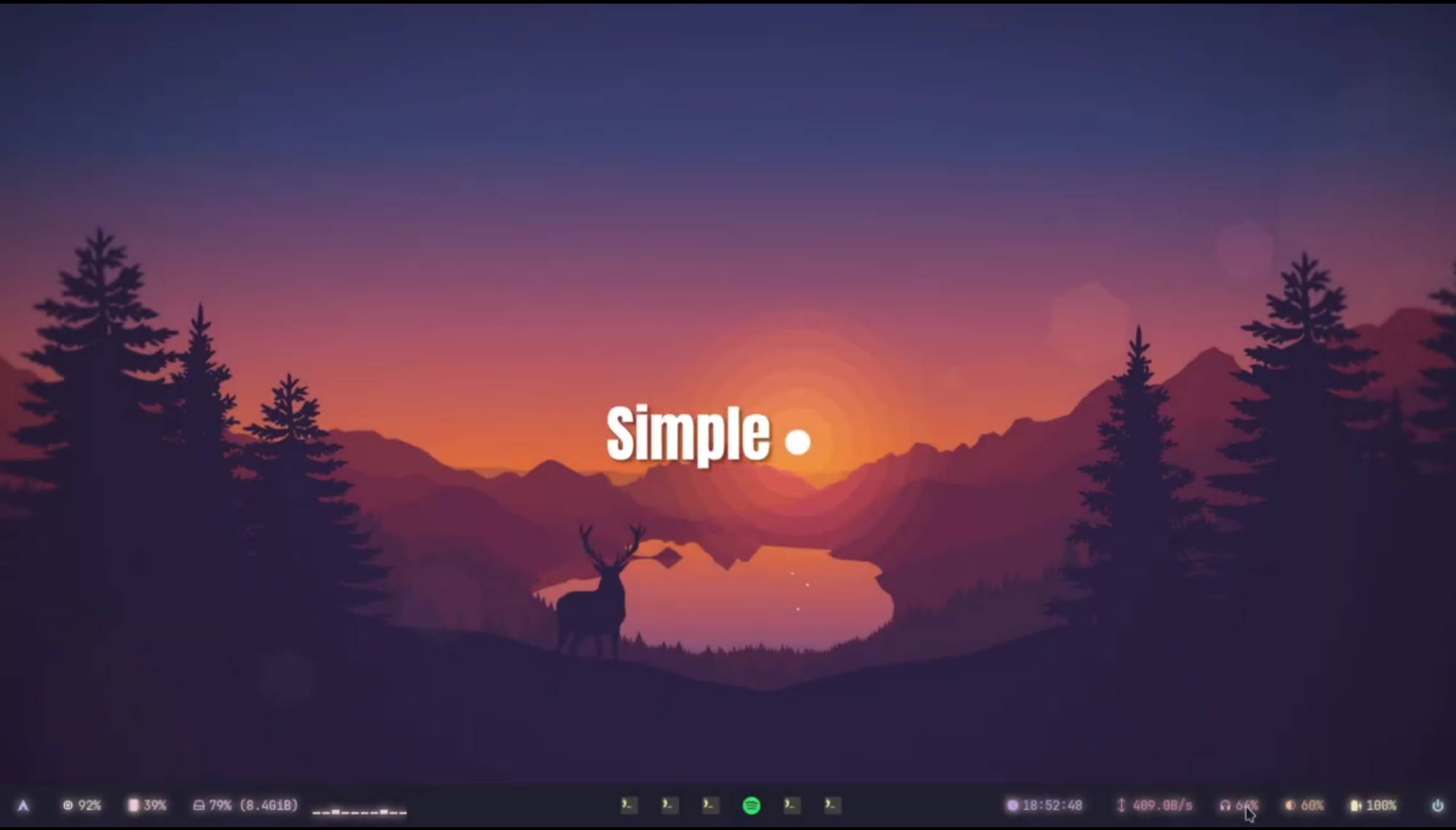
Compositors: for Xorg (a.k.a. X), for Wayland

- They can
  - add effects like transparency, animations or blur,
  - be standalone or built into the DE / WM.
- Example: Picom, Hyprland, Wayfire. Related fun [OD,YT] with Gnome extensions.
- For Hyprland: website, wiki; demo (screenshot):



Compositor: HyperInd$_{vid}$

Compositor: WayfireYT



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: YT_1, YT_2, YT_3, YT_4, YT_5.
   - exploring: Arch with Phosh (SSH, VNC, ... ✓); sxmo: looks exciting.

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 $\leftarrow$ WM $\leftarrow$ CLI; login manager; Vim

Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE $\leftarrow$ WM $\leftarrow$ CLI; login manager; Vim,
- educational channels

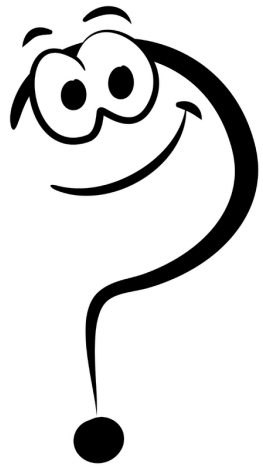
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE $\leftarrow$ WM $\leftarrow$ CLI; login manager; Vim,
- educational channels,
- ricing: conky, polybar, compositor

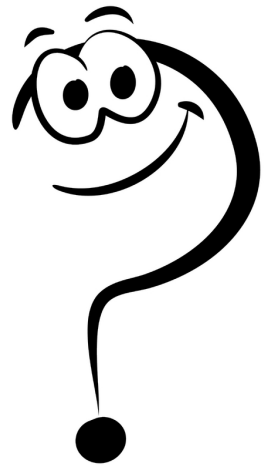
Summary

- Linux history, user freedom,
- distributions, installation, applications,
- DE $\leftarrow$ WM $\leftarrow$ CLI; login manager; Vim,
- educational channels,
- ricing: conky, polybar, compositor,
- Linux phones.

Are you ready to own your computer



Are you ready to own your computer



Feel free to share your

- adventure (how you liked Linux, new softwares/channels found),
- general constructive suggestions, . . .