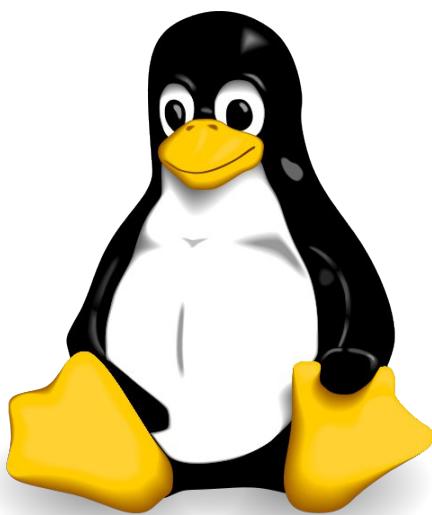


Instalace SW na Linuxu? - Typy a triky pro technicky znalé



Instalace SW na Linuxu? - Typy a triky pro technicky znalé.....	1
Jak se instaluje Software na Linuxu?.....	2
Manažer balíčků.....	2
Seznam populárních Manažerů Balíčků.....	2
Systémové Repozitáře.....	3
Konfigurace systémových repozitářů	3
Externí Repozitáře.....	4
Univerzální Repozitáře (Flatpak a Snap)	5
Flatpak.....	5
Flatpak Příkazy.....	5
Flatpak GUI	5
Flatpak - Výhody a Nevýhody	5
Snap.....	6
Snap vs Flatpak.....	6
Kdy Použít Flatpak nebo snap?	6
Manuální Instalace SW mimo repozitář.....	7
Balíčkový soubory	7
Binární Soubory.....	7
.Appimage soubor	8
Jednoduše Nastavit Pravomoce.....	8
Kompilovat SW ze zdrojového kódu.....	8
Recepty na komplikaci SW	8
Skripty .sh.....	9

Jak se instaluje Software na Linuxu?

- Instalace softwaru na Linuxu funguje výrazně jinak než na Windows.
- Místo stahování .exe souborů z internetu se většina programů instaluje z ověřených repozitářů.
- V repozitáři jsou **balíčky** které obsahují programy.

Manažer balíčků

- **Programy, Aplikace a Knihovny se na Linuxu balí do Balíčků.**
- Pro stahování balíčků z repozitářů se používá Manažer Balíčků.
- **Manažer Balíčků** může balíčky:
 - Stahovat a Instalovat *
 - Odstraňovat
 - Aktualizovat seznam balíčků
 - Aktualizovat balíčky
 - Zjišťovat stav(Verze, Dostupnost, Potřebné knihovny atd...)
 - Opravovat

*Manažer balíčků prvně stáhne balíček, pak ho rozbalením instaluje do systému. Balíček může také zapínat služby.

Seznam populárních Manažerů Balíčků

- Manažer balíčků existuje několik.
- Každá rodina Distribucí používá svoji.
- **Na to aby jsi mohl měnit balíčky na svém systému, musíš si dát opravení, pomocí sudo.**
např. `sudo apt install balíček` (Nedávaj sudo když jsi uživatel root)

Distribuce	Název	Instalace	Odstanení
Ubuntu	<code>apt</code> (<code>dpkg</code>)	<code>apt install balíček</code>	<code>apt remove balíček</code> <code>apt purge balíček</code> *
SUSE OpenSUSE	<code>ZYpp</code> (<code>rpm</code>)	<code>zypper install balíček</code>	<code>zypper remove balíček</code>
Fedora RHEL	<code>dnf</code> (<code>rpm</code>)	<code>dnf install balíček</code>	<code>dnf remove balíček</code>
Archlinux	<code>pacman</code>	<code>pacman -S balíček</code>	<code>pacman -R balíček</code>
VoidLinux	<code>xbps</code>	<code>xbps-install balíček</code>	<code>xbps-remove balíček</code>
Alpine Chimera	<code>apk</code>	<code>apk add balíček</code>	<code>apk del balíček</code>

*apt purge odstraní i konfigurační soubory spojené s daným balíčkem.

Distribuce	Aktualizace seznamu balíčků	Aktualizace balíčků	Více Info
Ubuntu Debian Linux Mint	<code>apt update</code>	<code>apt upgrade</code>	https://manpages.ubuntu.com/manpages/resolute/man8/apt.8.html
SUSE OpenSUSE	<code>zypper refresh</code>	<code>zypper update</code> Tumbleweed: <code>zypper dup</code>	https://manpages.opensuse.org/Tumbleweed/zypper/zypper.8.en.html
Fedora RHEL	<code>dnf update</code>	<code>dnf upgrade</code>	https://www.man7.org/linux/man-pages/man8/dnf.8.html
Archlinux	<code>pacman -Sy</code>	<code>pacman -Syy</code>	https://man.archlinux.org/man/pacman.8.en
Voidlinux	<code>xbps-install -S</code>	<code>xbps-install -u</code>	https://man voidlinux.org/xbps
Alpine Chimera	<code>apk update</code>	<code>apk upgrade</code>	https://www.mankier.com/8/apk

Výhody manažera balíčků:

- **Vyšší bezpečnost, ověřený zdroj SW**
- Aktualizace systému aktualizuje i aplikace
- Žádné „klikací instalátory“

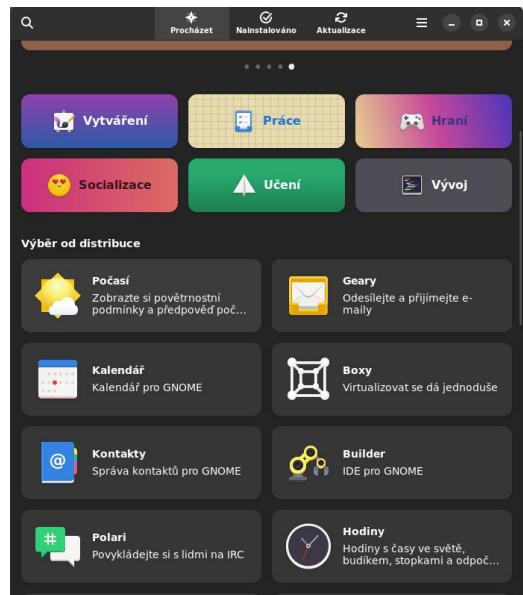
**Existují „Obchody“ pro snazší instalaci
Obchod je grafická verze manažer balíčků**

Pozor:

**Často nebývá všechno v obchodech.
Věci jako knihovny programů tam nemusí být.**

Přirovnání k Windows:

Repozitáře ≈ Microsoft Store (ale bez reklam)



Systémové Repozitáře

- Oficiální zdroje softwaru, které dodává distribuce.
- Obsahuje ověřené programy kde software je zkoušený a aktualizovaný

Jeden Linux systém může mít více repozitářů najednou.

➤ **Systémové repozitáře se často dělí podle Licence nebo architektury:**

- **Hlavní** - Většina balíčků
- **Ne-svobodné** – Linux balíčky které nejsou Open Source. Často to bývá Firmware a řadiče na HW.
- **Multilib** nebo **32bit** – Některé aplikace na linuxu jsou stále jen 32 bitové, tohle je problém na 64 bitovém systému. Užitečné např pro Steam a 32 bitové programy v Wine.
- **Zkušební/Externí** – Programy které nejsou tak dobře vyzkoušené a ještě nejsou v hlavním repo.

*repo se často používá jako zkratka pro repozitář(repository)

Konfigurace systémových repozitářů

- Konfigurace repozitářů bývá v konfiguračním souboru manažera balíčků.
- Příklady:

Debian

Soubor /etc/apt/sources.list

```
#deb cdrom:[Debian GNU/Linux 13.1.0 _Trixie_ - Official amd64 NETINST with firmware 20250906-1]
deb http://deb.debian.org/debian/ trixie main non-free-firmware
deb-src http://deb.debian.org/debian/ trixie main non-free-firmware

deb http://security.debian.org/debian-security trixie-security main non-free-firmware
deb-src http://security.debian.org/debian-security trixie-security main non-free-firmware

# trixie-updates, to get updates before a point release is made;
# see https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_updates_and_backports
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware
deb-src http://deb.debian.org/debian/ trixie-updates main non-free-firmware

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
```

Voidlinux

Soubory v /etc/share/xbps.d/ (Každý repo má svůj vlastní soubor)

```
[oscar@void-LP xbps.d]$ pwd  
/usr/share/xbps.d  
[oscar@void-LP xbps.d]$ ls -1  
00-repository-main.conf  
10-repository-multilib-nonfree.conf  
10-repository-multilib.conf  
10-repository-nonfree.conf  
void-virtualpkgs.conf  
xbps-arch.conf  
xbps.conf  
[oscar@void-LP xbps.d]$ cat 00* 10*  
repository=https://repo-default.voidlinux.org/current  
repository=https://repo-default.voidlinux.org/current/multilib/nonfree  
repository=https://repo-default.voidlinux.org/current/multilib  
repository=https://repo-default.voidlinux.org/current/nonfree
```

Nepiš do těchto Souborů! Přetři si postup na stránkách tvé linux distribuce!

Často to bývá tak že si instaluješ repozitář, např. něco ve stylu `xbps-install void-repo-nonfree`. Jen některé distribuce doporučují aby jsi přidával repozitáře skrze konfiguračních souborů (Archlinux).

Externí Repozitáře

- Někdy program není v oficiálních repozitářích nebo je tam starší verze.
- Řešením je přidání externího repozitáře výrobce SW.

Příklady:

- Google Chrome
- Visual Studio Code
- Docker (Pro nové verze)

Mimo přidávání repozitářů přímo do systémů, existují bezpečnější systémy který fungují odděleně:

Distribuce	Název	Co to je	Poznámka
Ubuntu Linux Mint	PPA (Personal Package Archive)	Externí repozitáře sdílené na https://launchpad.net/	Velmi jednoduché, oblíbené u začátečníků
Fedora	COPR (Cool Other Package Repo)	Komunitní repozitáře pro Fedora balíčky	Novější verze SW, méně zkoušené
openSUSE	OBS (Open Build Service)	Oficiální build systém SUSE Nabízí i pro jiné distribuce	Velmi kvalitní a testované
Univerzální	HomeBrew	Univerzální Repozitář pro MacOS a Linux	Ne tak kvalitní jako repozitáře navržené přímo pro Linux. Pro Programátory.

Univerzální Repozitáře (Flatpak a Snap)

Flatpak

- Běží v izolovaném prostředí (sandbox).
- Stejný balíček pro všechny distribuce, tohle umožňuje velmi dobré zkoušení kvality.
- Nejpopulárnější repozitář je Flathub ale existují i další (Můžeš si i vyrobit vlastní).
- Můžeš prohlížet nabídku aplikací z Flathub na: <https://flathub.org/cs>
- Stáhni si Flathub pro tvoji distribuci: <https://flathub.org/cs/setup>

Flatpak Příkazy

Jako příklad balíčku jsem použil VLC přehrávač

Přidání Flathub repo	<code>flatpak remote-add --if-not-exists flathub https://dl.flathub.org/repo/flathub.flatpakrepo</code>
Instalace balíček	<code>flatpak install flathub org.videolan.VLC</code>
Od-instalace balíčku	<code>flatpak remove org.videolan.VLC</code>
Opravit balíček	<code>flatpak repair org.videolan.VLC</code>
Ukázat oprávnění balíčku	<code>flatpak permission-show org.videolan.VLC</code>

Flatpak oprávnění

- Flatpak umožňuje si nastavit oprávnění ke každé aplikaci (úložiště, síť, zvuk atd...).
- Jednoduchá doporučená aplikace pro nastavení pravomocí:
<https://flathub.org/cs/apps/com.github.tchx84.Flatseal>
- Rozsáhlejší aplikace pro spravování flatpak aplikací a jejich data:
<https://flathub.org/cs/apps/io.github.flattool.Warehouse>

Flatpak GUI

- Flatpak má velmi vyspělé GUI obchody.
 - Existuje Gnome Software a KDE Discover pro Gnome a KDE plasma protředí.
 - Tyto často nejsou jen pro Flatpak ale i pro snap, apt a rpm.
- Ale když nemáš Gnome nebo KDE, tak bych doporučoval jiné Obchody

Název	Popis	Odkaz
EasyFlatpak	Jednoduché a Přehledné Autor Doporučuje	https://flathub.org/cs/apps/org.dupot.easyflatpak
Bazaar	Rychlé díky tomu že stahuje balíčky naráz. Díky tomu nefunguje na všechny distribuce.	https://flathub.org/cs/apps/io.github.kolunmi.Bazaar

Flatpak - Výhody a Nevýhody

Výhody	Nevýhody
Jistá instalace. Většinou se nic se nepokazí.	Flatpak Knihovny jsou jednolitější a zabírají více místa než jejich systémové varianty. Často knihovny zabírá více místa než samotná aplikace.
Žádné potíže s neshodou v knihovnách.	Aplikace zabírají ještě více místa protože občas 2 aplikace potřebují jiné verze stejné knihovny.
Nový Software rychle.	Příklad systému s 24 aplikacemi, kde celkový počet Flatpaků je 65...

Snap

- Vyvíjený firmou Canonical.
- Automatické aktualizace.
- Běží izolovaně.

Snap vs Flatpak

- Snap má jenom jeden Repozitář který ovládá Canonical.
- Snap server není svobodný Software.(Nemůžeš si vyrábět vlastní repozitáře)
- Snap nefunguje na všech Linux distribucích (Bez systemD).
- Snap má nevýhody Flatpak navíc má svoje nevýhody.
- Snap je technicky nezralý v porovnání s Flatpak, to je proč je častokrát pomalejší než Flatpak.
- Snap má ale velké plány, jako udělat z Linux jádra snap...?
- Na ubuntu když dás `sudo apt install` tak se ti u vybraných aplikací změní na `snap install..` Tohle způsobuje např:

Firefox se místo 30 vteřin stahování a 3 vteřin otevření, Se snap verze stahuje 2 minuty a otevírá 10 vteřin. Canonical na Ubuntu nutí uživatele použít objektivně horší snap...

➤ **Jestli chceš Ubuntu bez snap, použij Linux Mint. (Linux Mint je v zásadě Ubuntu bez snap)**

Nebo se můžeš pokusit odebrat snap z Ubuntu: <https://www.baeldung.com/linux/snap-remove-disable>

Kdy Použít Flatpak nebo snap?

- chceš nejnovější verzi aplikace
- aplikace není v repozitáři tvé distribuce
- chceš jednoduchou instalaci

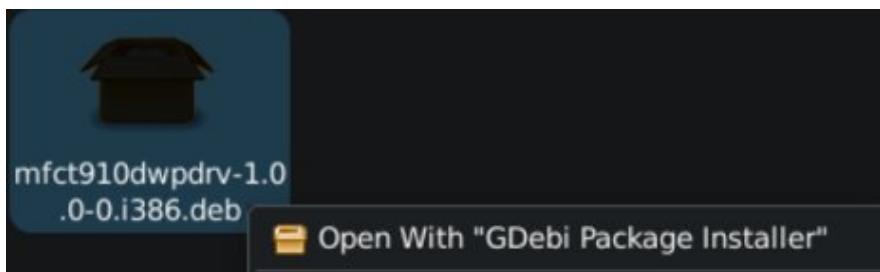
Manuální Instalace SW mimo repozitář

- Když ale požadovaný SW není v repozitáři, musí se nainstalovat manuálně.
- Tohle nejvíc připomíná Windows .exe
- Tohle není nejlepší metod, protože se SW neaktualizuje.

Balíčkový soubory

- Balíčkový soubory jsou pro manažery balíčků
- Příklad souborů: .deb pro Debian a .rpm pro RHEL a SUSE

Většinou se dají .deb nebo .rpm soubory instalovat pomocí GUI:



- Občas ale buď GUI není nebo nefunguje.
- Jako se všim v linuxu, jde to taky i v terminálu:

Debian Ubuntu Linux Mint	<code>sudo apt install ./balicek.deb</code> <code>sudo dpkg -i balicek.deb</code> *
RHEL Fedora	<code>sudo dnf install balicek.rpm</code> <code>sudo rpm -i balicek.rpm</code> *
SUSE OpenSUSE	<code>sudo zypper install balicek.rpm</code> <code>sudo rpm -i balicek.rpm</code> *

*Pomocí dpkg nebo rpm, instalace balíčků proběhne bez stahování knihoven a závislostí z repozitáře.
Použij jen v případě když apt, dnf nebo zypper není schopen nainstalovat balíček.

Binární Soubory

- Binární soubory které připomínají .exe na windows
- Binární soubory na Linuxu většinou nemají příponu .exe
- Na to aby se binární soubor mohl spustit, musí se mu dát pravomoce k spouštění:
 - `chmod +x binárka`
- Následně se spustí pomocí:
 - `./binárka`

Příklad Hello world! programu v jazyce C:

```
[oscar@void-LP Debug]$ ls
Ahoj-Světe
[oscar@void-LP Debug]$ ./Ahoj-Světe
bash: ./Ahoj-Světe: Permission denied
[oscar@void-LP Debug]$ chmod +x Ahoj-Světe
[oscar@void-LP Debug]$ ls
Ahoj-Světe
[oscar@void-LP Debug]$ ./Ahoj-Světe
Hello world!
```

[oscar@void-LP Debug]\$ – uživatel a adresář

Když dáš pravomoce pomocí chmod +x, tak se ti může soubor zjevit zeleně.

.Appimage soubor

- Univerzální standard pro binární soubory. *
- Jeden soubor který se spustí přímo a neinstaluje se.
- Appimage má v sobě zabudované všechny knihovny (+ pro uživatele).
- .Appimage souborům se musí dávat pravomoce stejně jako u jiných binárních souborů.
- Doporučuji tuto aplikaci pro spravování .appimage souborů:
<https://flathub.org/cs/apps/it.mijorus.gearlever>

*Většinou nefunguje na MUSL distribucích jako Alpine, Chimera a Void musl edice

Jednoduše Nastavit Pravomoce

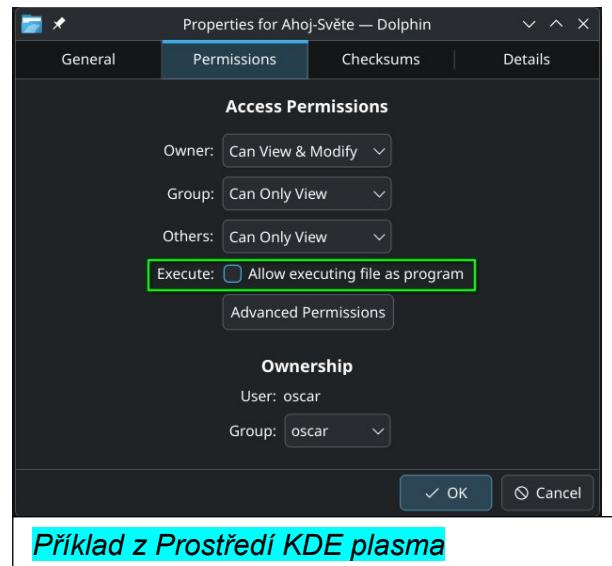
- Většinou jdou pravomoce nastavit v GUI:

Správce souborů ->

Vlastnosti souboru ->

Oprávnění ->

Nastavit aby soubor byl spustitelný jako program.



Tohle funguje na všechny soubory

Kompilovat SW ze zdrojového kódu

- Nejpokročilejší způsob instalace který se Používá se když ostatní metody nejdou.
- Často vyžaduje znalost programování nebo dobrou praxi.
- Instrukce od výrobce pomáhají.

Recepty na komplikaci sw

- Existují ale recepty na komplikování balíčků.
- K receptům je často vázán nástroj který recept přečte a zkompiluje.
- Populární jsou:

Distribuce	Název sbírky receptů	Nástroj pro komplikaci	Popis
Arch Linux	AUR (Arch User Repository)	<code>makepkg</code>	Hodně SW, Málo zkoušené, Křehké
Void Linux	<code>void-packages</code>	<code>void-src</code>	Zkoušené, málo SW. Jednodušeji se vyrábějí vlastní recepty.
Gentoo Linux	<code>portage</code>	<code>emerge</code>	Hlavní způsob jak instalovat balíčky na Gentoo. Velmi funkční a kvalitní.

Skripty .sh

- Soubor který spustí více příkazů za sebou.
- Podobné .bat z Windows.
- Skripty používají existující systémové příkazy.
- .sh skriptu se musí dát pravomoce se spouštět: `chmod +x skript.sh`
- Následně se spustí pomocí: `./skript.sh`

Příklad: Skript který nastaví systémový čas aby odpovídal času v BIOS:

(HW-clock-For-Windows-Dualboot.sh)

```
[oscar@void-LP Složka]$ ls
HW-clock-For-Windows-Dualboot.sh
[oscar@void-LP Složka]$ ./HW-clock-For-Windows-Dualboot.sh
bash: ./HW-clock-For-Windows-Dualboot.sh: Permission denied
[oscar@void-LP Složka]$ chmod +x HW-clock-For-Windows-Dualboot.sh
[oscar@void-LP Složka]$ ls
HW-clock-For-Windows-Dualboot.sh
[oscar@void-LP Složka]$ ./HW-clock-For-Windows-Dualboot.sh
This script sets the System clock to HW clock, the same as windows
Fixes time difference when dualbooting with windows
[oscar@void-LP Složka]$ █
```

Když dás pravomoce pomocí chmod +x, tak se ti může soubor zjevit zeleně.

Obsah HW-clock-For-Windows-Dualboot.sh skriptu:

```
#!/bin/bash
echo "This script sets the System clock to HW clock, the same as windows"
echo "Fixes time difference when dualbooting with windows"
sudo hwclock --systohc --localtime
```

- `#!/bin/bash` – Říká Linuxu ať použije Bash jako interpreta.
- `echo "Text"` – Vytiskne do terminálu text obsažen v uvozovkách
- `sudo hwclock --systohc --localtime` – příkaz který se spustí, stejně jako by jsi ho napsal ty.
(Nastavuje čas systému)

V skriptů můžou být i : Proměné, If, Do while, Case, funkce. (Občas to připomíná programování v C)

Skriptování je silný nástroj. Můžeš jednoduše vyrábět vlastní skripty pro svoji potřebu.

Příklad autorovy doporučených skriptů:

- Záchranný SW pro připojení se do terminálu rozbitého Linuxu:
<https://github.com/BitJam/chroot-rescue> (**Silně doporučuji**)
- SW na správu herního serveru pro hru Minecraft: <https://github.com/squidnose/MCserverTUI>
- Sbírka skriptů pro nastavování Voidlinuxu: <https://github.com/squidnose/Voidlinux-Post-Install-TUI>

Jestli se chceš naučit skriptovat, tento kurz může pomoci: <https://course.ysap.sh/>

- **Předtím než si nějaký skript spustíš, otevři ho a přečti si ho.**
- Tohle se dá udělat pomocí příkazu cat:

```
[oscar@void-LP Složka]$ cat HW-clock-For-Windows-Dualboot.sh
#!/bin/bash
echo "This script sets the System clock to HW clock, the same as windows"
echo "Fixes time difference when dualbooting with windows"
sudo hwclock --systohc --localtime
```

Více Manuálů tohoto typu můžeš najít na:

- <https://github.com/squidnose/Linux-Navody-CZ>

Zajímavé programy můžeš najít na:

- <https://github.com/squidnose>
- <https://codeberg.org/squidnose-code>

Tento Manuál Vytvořil Oscar R Fiala. ozfiala@gmail.com.

Tento dokument je šířen pod licencí GNU General Public License, verze 3 nebo jakákoli pozdější verze.

Více informací naleznete na: <https://www.gnu.org/licenses/gpl.html>