Pacman a AUR Helpers

V Arch Linuxu je "pacman" výchozím správcem balíčků. Je to nástroj příkazového řádku, který se používá k instalaci, aktualizaci a odstranění balíčků v Arch Linuxu a jeho derivátech. „AUR helpers“ jsou nástroje, které usnadňují práci s Arch User Repository (AUR), což je komunitně řízené úložiště, které obsahuje další softwarové balíčky, které nejsou součástí oficiálních repozitářů Arch Linuxu. Některé příklady pomocníků AUR zahrnují paru, yay, pikaur a aura. Tyto nástroje umožňují uživatelům snadno vyhledávat, instalovat a spravovat balíčky z AUR z příkazového řádku.

Pacman

správce balíčků

První krůčky

Pacman wiki home page
Správce balíků `pacman` je jednou z hlavních charakteristických vlastností **Arch Linuxu**. Kombinuje jednoduchý binární formát balíčku se snadno použitelným sestavovacím systémem (build system). Cílem pacmanu je umožnit snadnou správu balíčků, ať už jsou z oficiálních úložišť, nebo z vlastních sestavení uživatele.

Pacman udržuje systém aktuální synchronizací seznamů balíků s hlavním serverem. Tento model server/klient také umožňuje uživateli stahovat/instalovat balíčky jednoduchým příkazem, doplněný o všechny požadované závislosti.

Pacman je postaven na knihovně `libalpm` (Arch Linux Package Management), která poskytuje jednoduché API pro správu metadat balíčků a také podporu balíčků ve formátu balíčků Pacman (`.pkg.tar.xz`). Pacman se na tuto knihovnu spoléhá při instalaci, odstraňování a správě balíčků v systému.

Pacman je navržen tak, aby byl jednoduchý a snadno použitelný, s přímočarým rozhraním příkazového řádku, které vyžaduje malý zásah uživatele. Využívá jeden konfigurační soubor (`/etc/pacman.conf`), který se používá k určení umístění úložišť balíčků, stejně jako různé další možnosti, jako jsou podpisové klíče balíčků a možnosti synchronizace databáze.

Pacman funguje tak, že stáhne soubory balíčků z určeného úložiště balíčků a poté je extrahuje a nainstaluje do příslušných umístění v systému souborů. Spravuje také lokální databázi nainstalovaných balíčků, kterou používá ke sledování závislostí a řešení konfliktů balíčků.

Pacman obsahuje automatický systém aktualizace balíčků, který lze použít k aktualizaci všech nainstalovaných balíčků na nejnovější dostupnou verzi. Zahrnuje také podporu virtuálních balíčků, což jsou metabalíčky, které představují sadu souvisejících balíčků, které lze nainstalovat společně, aby poskytovaly konkrétní funkce.

Pacman je napsán v programovacím jazyce C a používá BSD tar formát pro balení.

Ukázka některých důležitých příkazů pro pacmana

```
sudo pacman -S awesome ## install package/meta-package
sudo pacman -Syu ## System update
sudo pacman -Syy ## sync database
pacman -Ss awesome ## search package with text 'gnome'
pacman -Qs awesome ## search installed packages
```

```
pacman -Si plasma-meta ## display extensive information
pacman -Qii awesome ## info + list of backup files
pacman -Qdt ## list packages no longer reqs (orphans)
sudo pacman -Sc ## clear packages in cached packages
sudo pacman -Scc ## all files in cache, strong aggresive, nothing leave in cache
sudo pacman -U /path/to/package/package_name-version.pkg.tar.zst #@ install local pckage, from AUR
sudo pacman -S --asdeps unzip ## install as dependency, can be removed as orphans
sudo pacman -Qe mc ## list version if it is explicitly installed
sudo pacman -D --asdeps unzip ## change the status to deps
sudo pacman -D --asexplicit unzip
sudo pacman -F pacman ## search files which are containing by package
sudo pacman --needed base-devel ## install if necessary
pactree awesome ## tree of depended packages
```

Pacman Configuration /etc/pacman.conf

Moje nastavení s barvami, přehlednou tabulkou stažení balíčků, **opravodovým** pacmanem a paralelním stahováním.

```
Color
CheckSpace
VerbosePkgLists
ILoveCandy
ParallelDownloads = 7
```

Bezpečnost a gpg podepisování: `SigLevel = Required DatabaseOptional`
Pacman podporuje podpisy balíčků, které do balíků přidávají další vrstvu zabezpečení. Výchozí konfigurace, `SigLevel = Required DatabaseOptional`, umožňuje ověření podpisu pro všechny balíčky na globální úrovni.

Nikdy nepoužívejte `SigLevel = Never` (jen zcela v případě velké nouze, nebo u člověka, kterého skutečně znáte a jeho závislost nelze obnovit kvůli nemoci například.)

Pacman Repositories multilib, extra, community a úroveň testing

Může se také stát, že úložiště obsahující balíček není ve vašem systému povoleno, např. Balíček může být v multilib úložišti, ale multilib není povolen ve vašem `pacman.conf`. Nutno povolit.

Můžete se stát i testerem pro Arch. Zapnout balíčky s `-testing`. Nutno pak ale updatovat celý systém, není jednoduchá změna! **Pozor**.

AUR

Arch User Repository

Arch User Repository (AUR)

Arch User Repository (AUR) je komunitní úložiště pro uživatele Arch. Obsahuje popisy balíků (**PKGBUILDs**), které vám umožňují sestavit balíček ze zdroje pomocí `makepkg` a poté jej nainstalovat pomocí `pacman`. AUR byla vytvořena za účelem organizace a sdílení nových balíčků z komunity a za účelem urychlení začlenění oblíbených balíčků do úložiště komunity.

Mnoho nových balíčků, které vstupují do oficiálních úložišť, začíná v AUR. V AUR mohou uživatelé přispívat svými vlastními sestavami balíků (PKGBUILDa související soubory). Komunita AUR má možnost hlasovat pro balíčky v AUR. Pokud se balíček stane dostatečně populárním - za předpokladu, že má **kompatibilní licenci a dobrou techniku balení** - může být vložen do komunitního úložiště (přímo přístupného pacmanem nebo abs). Například **Brave** licenci kompatibility nesplňuje, tedy nemůže být nikdy vložen do hlavních core repositářů, zůstane tedy v AUR.

Varování: Balíčky AUR jsou uživatelsky vytvářený obsah. Tyto PKGBUILD jsou zcela neoficiální a nebyli důkladně prověřeni. Jakékoli použití poskytnutých souborů je na vaše vlastní riziko. Proto vždy musí být před instalací zobrazen uživateli celý obsah PKGBUILD souboru a ten by měl ověřit jeho správnost. Pokud si nevíte rady, je dobré oslovit komunitu například na redditu a zeptat se na konkrétní balíček.

Pokud Vám nějaký balíček nejde nainstalovat, zaregistrujte se na AUR a do komentářů k danému balíčku napište problém, který máte. Maintainer balíčku Vám odpoví a nabídne řešení.

Yay, Paru, Aura pomocníci AUR

- JGuer/yay - Yet Another Yogurt - An AUR Helper napsán v Go. Byl navržen tak, aby byl rychlý, efektivní a uživatelsky přívětivý se zaměřením na jednoduchost a snadné použití. Yay je schopen provádět úkoly, jako je vyhledávání, instalace, aktualizace a správa balíčků z AUR.
- Morganamilo/paru - Paru AUR pomocník s mnoho features a minimální interakcí. Napsán Rust. Byl navržen tak, aby byl rychlý, lehký a snadno použitelný, se zaměřením na jednoduchost a minimalismus. Paru je schopen provádět úkoly, jako je vyhledávání, instalace, aktualizace a správa balíčků z AUR.
- fosskers/aura - Nadstandardní služby, zajímavé funkce, downgrades a zabezpečení upgrades a snapshots ovládání. Napsáno v Haskell. Byl navržen tak, aby byl jednoduchý, lehký a snadno použitelný, se zaměřením na automatizaci a spolehlivost. Aura je schopna provádět úkoly, jako je vyhledávání, instalace, aktualizace a správa balíčků z AUR.

Paru Instalace

```
sudo pacman -S --needed base-devel
git clone https://aur.archlinux.org/paru-bin.git
cd paru-bin
makepkg -si
```

Paru příklady užití

```
> paru <target> # Interaktivně vyhledávejte a instalujte <target>.
> paru # Alias pro paru -Syu.
> paru -S <target> # Nainstalujte si konkrétní balíček.
> paru -Sua # Upgradujte balíčky AUR.
> paru -Qua # Vytiskněte dostupné aktualizace AUR.
> paru -G <target> # Stáhněte si PKGBUILD a související soubory z <target>.
> paru -Gp <target> # Vytiskněte PKGBUILD z <target>.
> paru -Gc <target> # Vytiskněte si komentáře AUR pro <target>.
> paru --gendb # Vytvořte databázi devel pro sledování *-gitbalíčky. To je potřeba pouze
tehdy, když zpočátku používáte paru.
```

```
> paru -Ui # Vytvořte a nainstalujte PKGBUILD do aktuálního adresáře.
```

Aura instalace

```
git clone https://aur.archlinux.org/aura-bin.git
cd aura-bin
makepkg
sudo pacman -U <the-package-file-that-makepkg-produces>
```

Aura zajímavé příkazy

```
> aura -Pa # Analyzujte všechny místně nainstalované balíčky AUR.
> aura -O # Zobrazit osiřelé balíčky.
> aura -L # Zobrazit protokol Pacman.
> aura -Li <package> # Zobrazit historii instalace / upgradu balíčku.
> aura -Cc <n> # Odstranit všechny kromě nejnovější nverze každého balíčku uloženého v
mezipaměti.
> aura -C <package> # Downgrade balíčku.
> aura -Cv # Odstranit všechny /var/cache/aura/vcsmezipaměti
> aura -B # Uložte záznam JSON všech nainstalovaných balíčků.
> aura -Br # Obnovte uložený záznam. Podle potřeby se vrací zpět a odinstaluje.
> aura -Bc <n> # Odstranit všechny kromě nejnovější nuložené státy.
> aura -Bl # Zobrazit všechny uložené názvy souborů stavu balíčku.
> aura -Au # Upgradujte všechny nainstalované balíčky AUR.
> aura -Akuax # Oblíbený autor (upgrady, odstranění makedeps, ukazuje rozdíly PKGBUILD,
ukazuje postup)
> aura -As <regex> # Hledejte AUR pomocí regexu.
> aura -Ap <package> # Zobrazit balíček PKGBUILD.
> aura -Ad <package> # Seznam závislostí balíčku.
```