

# APT (Advanced Package Tool)

---

Jiří Klepl

# Proč chtít package manager

- **Z pohledu uživatele**

- Snadnost instalace & updatů
- Jednoduché hledání
- Přehled o nainstalovaném softwaru
- Udržení kompatibility mezi různými programy
- Bezpečnost

- **Z pohledu developera**

- Přívětivost pro uživatele
- Vyšší důvěryhodnost (má to úroveň)
- Možnost vytvoření ekosystému

# Package managery

- **apt**: Debian/Ubuntu
- **dnf**: Red Hat/Fedora
- **zypper**: SLES/openSUSE
- **pacman**: Arch
- **emerge**: Gentoo

Podrobné srovnání: <https://wiki.archlinux.org/index.php/Pacman/Rosetta>

- Package manager pro debian (a ubuntu)
- Skupina více programů: **apt**, **apt-get**, **apt-cache**
- Frontend pro **dpkg**: low-lvl nástroj pro instalaci a správu paketů

- Více než 50'000 packageů (Debian), 60'000 (Ubuntu)
- Instalace

```
sudo apt install <package>
```

- Odstranění

```
sudo apt remove <package>
```

- Update & upgrade

```
sudo apt update && sudo apt upgrade
```

## APT - příkazy pro dotazování

- Hledání package

```
apt search "regex"
```

- Informace o packagi

```
apt info <package>
```

- Seznam nainstalovaných paketů

```
apt list --installed
```

- Seznam ručně nainstalovaných paketů

```
apt list --manual-installed
```

**Příklad: ./examples.sh**

## Přidání dalších paketů

Jednoduše:

```
sudo add-apt-repository "deb <repository>"  
sudo apt-key adv --fetch-keys <url>  
sudo apt-get update
```

Nebo:

```
sudo <oblíbený editor (napr. VIM)> /etc/apt/sources.list  
curl -L <url> | sudo apt-key add -
```



- cuda toolkit  
<https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html>
- intel oneAPI  
<https://software.intel.com/content/www/us/en/develop/articles/installing-intel-oneapi-toolkits-via-apt.html>

Dva druhy

- **binary package**

- záznam repozitáře začíná “deb”,  
např. deb <https://deb.debian.org/debian> bullseye main
- pro koncové uživatele
- obecně jen pro určité architektury

- **source package**

- záznam repozitáře začíná “deb-src”,  
např. deb-src <https://deb.debian.org/debian> bullseye main
- pro development, testing a customizaci

## Z čeho se skládá package připraven pro sestavení

- složka **debian** ve složce **<název>-<verze (major.minor.patch)>** se zdrojem programu; obsah **debian**:
  - **changelog**: informace o verzích package
  - **rules**: script pro sestavení package (např. makefile - může pak spustit tradiční makefile programu; tato část je extrémě flexibilní - zde použijeme makefile s dh\_helper, co řeší boilerplate)
  - **control**: různé informace o package
    - název, aktuální verze a kategorie (sekce a priorita (důležitost pro systém))
    - maintainer a homepage
    - popis programu
    - architektura (např. amd64, ale i all nebo any)
    - informace o (build-) dependencích a vyloučení
  - **compat**: verze dh\_helperů

**Příklad: ../packages/myhello-1.0.0/debian**

## Sestavení package

- `debuild -b` sestaví podepsaný `.deb` package
- `debuild -b -us -uc` sestaví nepodepsaný `.deb` package pro vlastní použití
- složna **debian** simuluje root file-systému
  - binárka programu tradičně v **/usr/bin/**, tedy při sestavování package:  
**debian/usr/bin/**
  - library programu tradičně v **/usr/lib.../**, tedy při sestavování package:  
**debian/usr/lib.../**
- vznikne `.deb` archív a doprovodné soubory
- název vzniklého archívu obsahuje verzi, číslo revize a architekturu

**Příklad: ../packages/myhello-1.0.0**

## **.deb soubor: ar archive**

```
$ ar tv wget_1.12-2.1_i386.deb
rw-r--r-- 0/0 4 Sep 5 15:43 2010 debian-binary
rw-r--r-- 0/0 2403 Sep 5 15:43 2010 control.tar.gz
rw-r--r-- 0/0 751613 Sep 5 15:43 2010 data.tar.gz
```

- `debian-binary`: version of the deb file format, "2.0\n"
- `control.tar.gz`: metadata about the package
- `control`, `md5sums`, `(pre|post)(rm|inst)`, `triggers`, `shlibs`,...
- `data.tar.gz`: data files of the package

from: <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf>

## Co když mám důvěryhodný program, který není .deb package

Lze jednoduše zabalit do .deb package a nainstalovat přes dpkg / apt (opět příklad myhello)

Proč to dělat?

- Konzistence se zbytkem systému (jednotný způsob instalace)
- Chci mít přehled i o softwaru, který není z repozitářů pro můj OS
- Vyhnutí se mrtvým souborům v systému
- Vyhnutí se mrtvým souborům v systému



Složka s .deb packagey a pomocnými soubory:

- `Packages`: soubor se seznamem packageů, stručnými informacemi a kontrolními součty
- `Packages.gz` (optional): komprimovaný soubor `Packages`
- `Release`: soubor s kontrolními součty pro soubory `Packages` (,`Packages.gz`) a sebe
- `Release.gpg`: gnupg podpis souboru `Release`
- `InRelease`: `Release` a `Release.gpg` dohromady
- `public.key` (pokud nelze předat jiným způsobem): veřejný klíč pro ověření

**Příklad:**

**../packages/scripts.sh**

**"deb https://klepl.cz/packages /"**

- <https://www.debian.org/doc/manuals/packaging-tutorial/packaging-tutorial.en.pdf>
- <https://www.debian.org/doc/debian-policy/index.html>
- <https://medium.com/sqooba/create-your-own-custom-and-authenticated-apt-repository-1e4a4cf0b864>
- <https://wiki.archlinux.org/index.php/Pacman/Rosetta>

# **Univerzum pro dotazy**

**Díky za pozornost!**