

Пакетни системи и мениджъри — 126, Операционни системи

Увод

Пакетните системи са ключов компонент на всяка Linux дистрибуция, който улеснява инсталирането, актуализирането и премахването на софтуер. Вместо ръчно компилиране и конфигуриране, пакетните мениджъри автоматизират целия процес.

Какво е пакет?

Пакетът е архив, който съдържа:

- Компилирани изпълними файлове
- Конфигурационни файлове
- Документация
- Информация за зависимости
- Скриптове за инсталация и деинсталация

Защо са важни пакетните мениджъри?

- **Автоматично управление на зависимости** - инсталира всички необходими библиотеки
- **Централизирани хранилища** - сигурен и проверен софтуер
- **Лесна актуализация** - една команда за обновяване на целия софтуер
- **Чисто премахване** - изтрива всички файлове на програмата
- **Проверка на интегритет** - гарантира непроменени файлове

1. Основни понятия

Пакетен мениджър (Package Manager)

Инструмент за автоматизирано управление на софтуерни пакети в операционната система.

Основни функции:

- **Инсталиране** - добавяне на нов софтуер
- **Актуализиране** - обновяване до най-нова версия
- **Премахване** - изтриване на софтуер
- **Търсене** - намиране на налични пакети
- **Управление на зависимости** - автоматична инсталация на необходими библиотеки

Хранилище (Repository)

Хранилището е централен сървър, който съдържа колекция от пакети. Дистрибуциите поддържат официални хранилища с проверен софтуер.

Видове хранилища:

- **Официални** - поддържани от разработчиците на дистрибуцията
- **Общностни** - поддържани от общността (AUR в Arch Linux)
- **Трети страни** - PPA в Ubuntu, EPEL в RHEL

Зависимости (Dependencies)

Зависимостите са други пакети, необходими за работата на програмата. Пакетните мениджъри автоматично ги инсталират.

```
$ apt install firefox Reading package lists... Building dependency tree... The following additional packages will be installed: libgtk-3-0 libdbus-glib-1-2 libxt6 5 newly installed, 0 upgraded
```

2. Видове пакетни системи

2.1. DEB системи (Debian, Ubuntu, Mint)

dpkg - нисконивов пакетен мениджър

dpkg е основният инструмент за работа с .deb пакети. Не управлява зависимости автоматично.

Основни команди:

- `dpkg -i package.deb` - инсталиране на пакет
- `dpkg -r package` - премахване на пакет
- `dpkg -l` - списък на инсталираните пакети
- `dpkg -L package` - показва файловете на пакета
- `dpkg -S /path/to/file` - намиране на пакета, който съдържа файл
- `dpkg --configure -a` - конфигуриране на неконфигурирани пакети

Пример - инсталиране на .deb файл:

```
$ wget https://example.com/program.deb  
$ sudo dpkg -i program.deb  
$ sudo apt install -f
```

Забележка: `apt install -f` инсталира липсващи зависимости

APT - високонивов пакетен мениджър

apt (Advanced Package Tool) е модерен интерфейс за управление на пакети в Debian-базирани системи. Автоматично управлява зависимости.

Основни команди:

- `apt update` - актуализиране на списъка с пакети
- `apt upgrade` - актуализиране на инсталираните пакети
- `apt install package` - инсталиране на пакет
- `apt remove package` - премахване на пакет
- `apt purge package` - премахване на пакет с конфигурации
- `apt search keyword` - търсене на пакети
- `apt show package` - информация за пакет
- `apt autoremove` - премахване на неизползвани зависимости
- `apt full-upgrade` - пълна актуализация на системата
- `apt list --installed` - списък на инсталираните пакети

Пример - инсталиране на софтуер:

```
$ sudo apt update
$ sudo apt search "text editor"
$ sudo apt install vim
$ sudo apt show vim
$ sudo apt remove vim
```

```
Практически пример: $ sudo apt update Hit:1
http://bg.archive.ubuntu.com/ubuntu jammy InRelease Get:2
http://security.ubuntu.com/ubuntu jammy-security InRelease [110
kB] Fetched 110 kB in 1s (110 kB/s) Reading package lists... Done
$ sudo apt install htop Reading package lists... Done Building
```

```
dependency tree... Done The following NEW packages will be
installed: htop 0 upgraded, 1 newly installed, 0 to remove Do you
want to continue? [Y/n] y Get:1
http://bg.archive.ubuntu.com/ubuntu jammy/main amd64 htop amd64
3.0.5-7build2 [124 kB] Fetched 124 kB in 0s (248 kB/s) Selecting
previously unselected package htop. Unpacking htop (3.0.5-7build2)
... Setting up htop (3.0.5-7build2) ...
```

apt vs apt-get

apt е по-нов и по-user-friendly интерфейс, препоръчан за интерактивна употреба. **apt-get** е по-стар, подходящ за скриптове.

2.2. RPM системи (Red Hat, Fedora, CentOS)

rpm - нисконивов пакетен мениджър

rpm (Red Hat Package Manager) е базовият инструмент за работа с .rpm пакети.

Основни команди:

- `rpm -i package.rpm` - инсталиране на пакет
- `rpm -U package.rpm` - актуализиране на пакет
- `rpm -e package` - премахване на пакет
- `rpm -qa` - списък на всички инсталирани пакети
- `rpm -ql package` - файлове на пакета
- `rpm -qf /path/to/file` - намиране на пакета по файл
- `rpm -qi package` - информация за пакет
- `rpm -V package` - проверка на интегритета на пакет

Пример:

```
$ rpm -qa | grep firefox  
$ rpm -ql firefox  
$ rpm -qi firefox
```

yum - високонивов пакетен мениджър (CentOS 7, RHEL 7)

yum (Yellowdog Updater Modified) автоматично управлява зависимости за RPM пакети.

Основни команди:

- `yum update` - актуализиране на всички пакети
- `yum install package` - инсталиране на пакет
- `yum remove package` - премахване на пакет
- `yum search keyword` - търсене на пакети
- `yum info package` - информация за пакет
- `yum list installed` - инсталирани пакети
- `yum list available` - налични пакети
- `yum groupinstall "Group Name"` - инсталиране на група пакети
- `yum clean all` - изчистване на кеша

Пример:

```
$ sudo yum update
$ sudo yum install nginx
$ sudo yum groupinstall "Development Tools"
$ sudo yum remove nginx
```


dnf - високонивов пакетен мениджър (Fedora, RHEL 8+)

dnf (Dandified YUM) е следващото поколение на yum с по-добра производителност.

Основни команди:

- `dnf update` или `dnf upgrade` - актуализиране на системата
- `dnf install package` - инсталиране на пакет
- `dnf remove package` - премахване на пакет
- `dnf search keyword` - търсене на пакети
- `dnf info package` - информация за пакет
- `dnf list installed` - инсталирани пакети
- `dnf autoremove` - премахване на неизползвани зависимости
- `dnf history` - история на транзакциите
- `dnf history undo last` - отмяна на последната операция

Пример:

```
$ sudo dnf update  
$ sudo dnf install git  
$ sudo dnf history  
$ sudo dnf history undo last
```

2.3. Arch Linux системи

pacman - пакетен мениджър на Arch Linux

pacman е бърз и ефективен пакетен мениджър, използващ бинарни пакети.

Основни команди:

- `pacman -Syu` - синхронизиране и актуализиране на системата
- `pacman -S package` - инсталиране на пакет
- `pacman -R package` - премахване на пакет
- `pacman -Rs package` - премахване на пакет с неизползвани зависимости
- `pacman -Ss keyword` - търсене в хранилището
- `pacman -Qs keyword` - търсене в инсталираните
- `pacman -Qi package` - информация за инсталиран пакет
- `pacman -Ql package` - файлове на пакета
- `pacman -Qo /path/to/file` - намиране на пакета по файл
- `pacman -Sc` - изчистване на кеша

Пример:

```
$ sudo pacman -Syu
$ sudo pacman -S firefox
$ pacman -Ss browser
$ sudo pacman -Rs firefox
```

```
Практически пример: $ sudo pacman -Syu :: Synchronizing package
databases... core is up to date extra is up to date community is
up to date :: Starting full system upgrade... resolving
dependencies... looking for conflicting packages... Packages (42)
lib32-gcc-libs-13.2.1-5 ... Total Download Size: 150.24 MiB Total
```

```
Installed Size: 620.15 MiB Net Upgrade Size: 12.35 MiB :: Proceed  
with installation? [Y/n]
```

AUR (Arch User Repository)

Общностно хранилище с пакети, поддържани от потребители. Използва се с помощни инструменти като `yay` или `paru`.

```
$ yay -S google-chrome  
$ yay -Syu
```

2.4. Други пакетни системи

Slackware - `pkgtool`

Един от най-старите пакетни мениджъри, прост и директен.

- `installpkg package.tgz` - инсталиране
- `removepkg package` - премахване
- `upgradepkg package.tgz` - актуализиране
- `pkgtool` - текстов интерфейс за управление

Gentoo - Portage (emerge)

Система за компилиране от сорс код, максимална оптимизация.

- `emerge --sync` - синхронизиране
- `emerge package` - инсталиране
- `emerge --update --deep --newuse @world` - актуализиране
- `emerge --unmerge package` - премахване
- `emerge --search keyword` - търсене

3. Сравнение на пакетни мениджъри

Операция	Debian/Ubuntu (apt)	RHEL/Fedora (dnf/yum)	Arch Linux (pacman)
Актуализиране на списъка	<code>apt update</code>	Автоматично	<code>pacman -Sy</code>
Актуализиране на системата	<code>apt upgrade</code>	<code>dnf upgrade</code>	<code>pacman -Syu</code>
Инсталиране	<code>apt install PKG</code>	<code>dnf install PKG</code>	<code>pacman -S PKG</code>
Премахване	<code>apt remove PKG</code>	<code>dnf remove PKG</code>	<code>pacman -R PKG</code>
Търсене	<code>apt search KEY</code>	<code>dnf search KEY</code>	<code>pacman -Ss KEY</code>
Информация	<code>apt show PKG</code>	<code>dnf info PKG</code>	<code>pacman -Qi PKG</code>
Списък инсталирани	<code>apt list --installed</code>	<code>dnf list installed</code>	<code>pacman -Q</code>
Почистване	<code>apt autoremove</code>	<code>dnf autoremove</code>	<code>pacman -Sc</code>

4. Универсални пакетни системи

4.1. Snap (Canonical)

Универсални пакети, работещи на всяка Linux дистрибуция. Изолирани от системата (containerized).

Основни команди:

- `snap find keyword` - търсене
- `snap install package` - инсталиране
- `snap remove package` - премахване
- `snap list` - инсталирани snap пакети
- `snap refresh` - актуализиране на всички
- `snap refresh package` - актуализиране на конкретен пакет

Пример:

```
$ sudo snap install spotify  
$ sudo snap install --classic code  
$ snap list  
$ sudo snap remove spotify
```

Предимства: Една версия за всички дистрибуции, автоматични актуализации, изолация

Недостатъци: По-големи пакети, по-бавно стартиране, централизиран контрол

4.2. Flatpak

Алтернатива на Snap, фокусирана на desktop приложения. Отворен стандарт.

Основни команди:

- `flatpak search keyword` - търсене
- `flatpak install flathub app` - инсталиране
- `flatpak uninstall app` - премахване
- `flatpak list` - инсталирани приложения
- `flatpak update` - актуализиране на всички
- `flatpak run app` - стартиране на приложение

Пример:

```
$ flatpak remote-add --if-not-exists flathub  
https://flathub.org/repo/flathub.flatpakrepo  
$ flatpak install flathub org.gimp.GIMP  
$ flatpak run org.gimp.GIMP  
$ flatpak update
```

4.3. AppImage

Portable формат - един файл, който се изпълнява директно без инсталация.

Използване:

```
$ chmod +x application.AppImage  
$ ./application.AppImage
```

Предимства: Няма инсталация, portable, не изисква root

Недостатъци: Липса на централизирана актуализация, дублиране на библиотеки

5. Добри практики

Преди инсталация:

- **Винаги актуализирайте списъка с пакети** преди инсталация
- **Проверявайте информацията за пакета** - размер, зависимости, описание
- **Използвайте официални хранилища** - по-сигурни и стабилни

Редовна поддръжка:

```
# Debian/Ubuntu
$ sudo apt update && sudo apt upgrade
$ sudo apt autoremove
$ sudo apt autoclean

# Fedora/RHEL
$ sudo dnf upgrade
$ sudo dnf autoremove
$ sudo dnf clean all

# Arch Linux
$ sudo pacman -Syu
$ sudo pacman -Sc
```

Безопасност:

- Редовно актуализирайте системата за security patches
- Избягвайте неофициални хранилища без проверка
- Проверявайте цифровите подписи на пакетите
- Четете съобщенията при инсталация

Решаване на проблеми:

- **Счупени зависимости:**

```
$ sudo apt install -f          # Debian/Ubuntu
$ sudo dnf distro-sync         # Fedora
$ sudo pacman -Syu --force    # Arch
```

- **Заклучени пакети:**

```
$ sudo rm /var/lib/dpkg/lock-frontent # Ubuntu
$ sudo rm /var/lib/apt/lists/lock
```

- **Изчистване на кеша:**

```
$ sudo apt clean
$ sudo dnf clean all
$ sudo pacman -Scc
```

6. Практически примери

Сценарий 1: Инсталиране на web сървър (Nginx)

```
# Ubuntu/Debian
$ sudo apt update
$ sudo apt install nginx
$ sudo systemctl start nginx
$ sudo systemctl enable nginx

# Fedora
$ sudo dnf install nginx
$ sudo systemctl start nginx
$ sudo systemctl enable nginx

# Arch Linux
$ sudo pacman -S nginx
$ sudo systemctl start nginx
$ sudo systemctl enable nginx
```

Сценарий 2: Инсталиране на Development Tools

```
# Ubuntu/Debian
$ sudo apt install build-essential git vim

# Fedora
$ sudo dnf groupinstall "Development Tools"
$ sudo dnf install git vim

# Arch Linux
$ sudo pacman -S base-devel git vim
```

Сценарий 3: Премахване на стар софтуер

```
# Намиране на инсталирани пакети
$ dpkg -l | grep editor      # Debian/Ubuntu
$ rpm -qa | grep editor      # RHEL/Fedora
$ pacman -Q | grep editor    # Arch

# Премахване с конфигурации
$ sudo apt purge gedit       # Debian/Ubuntu
$ sudo dnf remove gedit      # Fedora
$ sudo pacman -Rns gedit     # Arch

# Почистване на неизползвани зависимости
$ sudo apt autoremove
$ sudo dnf autoremove
$ sudo pacman -Qdtq | sudo pacman -Rns -
```

```
Пълен пример - актуализация на системата: user@ubuntu:~$ sudo apt
update Hit:1 http://bg.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110
kB] Get:3 http://bg.archive.ubuntu.com/ubuntu jammy-updates InRelease
[119 kB] Fetched 229 kB in 1s (229 kB/s) Reading package lists...
Done Building dependency tree... Done 15 packages can be upgraded.
Run 'apt list --upgradable' to see them. user@ubuntu:~$ apt list --
upgradable Listing... firefox/jammy-updates 121.0+build1-
0ubuntu0.22.04.1 amd64 [upgradable from: 120.0+build2-
0ubuntu0.22.04.1] libssl3/jammy-updates,jammy-security 3.0.2-
0ubuntu1.12 amd64 [upgradable from: 3.0.2-0ubuntu1.10] ...
user@ubuntu:~$ sudo apt upgrade Reading package lists... Done
Building dependency tree... Done Calculating upgrade... Done The
following packages will be upgraded: firefox libssl3 ... 15 upgraded,
0 newly installed, 0 to remove Need to get 156 MB of archives. After
this operation, 1,024 B of additional disk space will be used. Do you
want to continue? [Y/n] y Get:1 http://bg.archive.ubuntu.com/ubuntu
jammy-updates/main amd64 libssl3 amd64 3.0.2-0ubuntu1.12 [1,902 kB]
... Fetched 156 MB in 45s (3,467 kB/s) Setting up libssl3:amd64
(3.0.2-0ubuntu1.12) ... Setting up firefox (121.0+build1-
0ubuntu0.22.04.1) ... Processing triggers for ... Done.
```

Заклучение

Пакетните мениджъри са фундаментален инструмент за работа с Linux системи. Овластяването им е ключово за ефективна системна администрация.

Ключови моменти за запомняне:

- **Всяка дистрибуция** има своя предпочитан пакетен мениджър
- **Винаги актуализирайте** преди инсталация - `update/upgrade`
- **Използвайте официални хранилища** за сигурност и стабилност
- **Почиствайте редовно** неизползваните зависимости
- **Четете документацията** на вашата дистрибуция

Полезни ресурси:

- **Debian/Ubuntu:** `man apt`, `apt --help`
- **RHEL/Fedora:** `man dnf`, `dnf --help`
- **Arch Linux:** `man pacman`, Arch Wiki