

Дистрибуция на софтуер

Универсални конфигурации с Nix

Павел Атанасов Камен Младенов

01.04.2025

Раздел 1

Проблеми при дистрибуцията на софтуер

Стъпки при добавянето на нов софтуер

- Трябва да инсталираш програмата **и** зависимостите ѝ
- Инсталирането на програми/библиотеки се свежда до копиране на нужните файлове в някои **глобални** места: /lib, /bin, C:\Program Files, /Applications, ...
- Конфигурации се слагат на редица **глобални** (спрямо потребителя) локации: /etc, ~/.config, %APPDATA%, ...

Ограничения (при ръчно изпълнение)

- ❶ Трудоемко е да намерим и инсталираме всички зависимости, да ги обновяваме
- ❶ Трябва да избираме само програми които се поддържат от процесорната архитектура, ОС, ...
- ❷ При премахването на програма първо трябва да сме запомнили къде е инсталирана, второ трябва да премахнем само не-споделените зависимости.
- ❸ Близко до невъзможно е да инсталираме повече от една версия на програма **или** зависимост.

Възможно е две програми да ползват различни версии на зависимости!

- ❶ Инсталирането/употребата на нови програми зависи от глобалната среда.

Тоест зависи от състоянието на файловата система, мрежата, ...

Пакетни мениджъри

- 0, 1 и 2 се разрешават чрез “пакетни мениджъри”
 - ▶ Примери: `apt`, `yum`, `pacman`, `brew`, ...
 - ▶ По-отдалечени примери: Apple App Store, Microsoft Store, Google Play Store, ...
- Даваш им редица уеб адреси в които да търсят програми и какво искаш да инсталираш.
 - ▶ Примери: <https://mirror.telepoint.bg/ubuntu/>,
<https://mirrors.uni-plovdiv.net/archlinux>, <https://mirrors.neterra.net/fedora>
- Те автоматично избират подходящите програми и зависимости за коректната архитектура/ОС/..., следят къде и какво се инсталира.
- Не предоставят решения за останалите проблеми

Раздел 2

Възможни решения за проблеми 3 и 4

Инсталиране на повече от една версия на програма

- Бинарката (изпълнимият файл) се казва **firefox** и всички очакват да я намерят в `/bin/firefox`.
- На първо време ще съхраним всяка програма в **уникална** директория, спрямо името и версията си.
 - ▶ Вместо `/bin/firefox`, правим `/somewhere/firefox-136.0.2`

Инсталиране зависи от глобалната среда

- Една програма търси своите зависимости в `/bin`, `/lib`, ...
- “Презаписваме” глобалните директории по начин, който влияе само на една програма.
 - ▶ Най-често директно променяме изпълнимия файл, така че да търси в нашите желани директории.
 - ▶ Когато това не е възможно, програмата се пуска през нещо като емулатор, който по време на изпълнение “конвертира” заявки към файловата система.

Примерно, програмата иска `/bin/firefox`, даваме ѝ `/somewhere/firefox-136.0.2`

- Пакетните мениджъри решават първите три проблема.
- Тези идеи биха разрешили последните два.
- Не можем ли да разработим пакетен мениджър, който реализира тези схеми?
- Можем! Даже вече е създадено...

Раздел 3

Nix пакетният мениджър

Nix пакетният мениджър

- Eelco Dolstra описва всички тези проблеми и решенията им преди >20 години
- Всяка програма се намира под `/nix/store/HASH-PROGRAM-VERSION`
- Хешовете са нещо като идентификатори, които се генерират автоматично спрямо програмата и нейните настройки.
 - ▶ Не са уникални, но ни позволяват да разграничаваме едни и същи програми, създадени с различни настройки.
 - ▶ Примерно, разграничава ако `firefox` е инсталирано с поддръжка на определени видео кодекси или не.
- За локалната среда са измислени `patchELF` и `buildFHSEnv`.

Раздел 4

Въпроси?

Раздел 5

Следващия път...

Следващия път...

- Ще инсталираме Nix пакетния мениджър
- Ще разучим как да го използваме