

Diploma

Ясен Ефремов

November 2022

Увод

В днешно време хората използват по-голямо количество софтуер от когато и да било. Като започнем от ежедневно употребяваните програми, а именно уеб браузърът, редакторите за документи, софтуерът за обработка на изображения, видео и аудио, и стигнем чак до видеоигрите. Средностатистическият потребител, е свикнал да си набавя всякакъв вид софтуер като просто потърси за него в интернет пространството, намери достатъчно достоверно изглеждаща уеб страница, от която да го свали, и изтегли приложението под формата на автоматизиран инсталатор (файлове с разширение .msi, .deb, .dmg) или директно изпълним файл (файлове с разширение .exe). Този подход обаче има недостатъци, най-големият от които е сигурността. Освен с полезен софтуер, интернет пространството е пълно и със зловреден софтуер. По-неопитните потребители лесно биха се заблудили от рекламите, фалшивите бутони за инсталация и предупредителните съобщения в някои уеб страници, в резултат на което биха си инсталирали вирус, спайуер (spyware) или рансъмуер (ransomware), който криптира файловете на компютъра и иска паричен откуп, за да ги възстанови. Има обаче начини да се предпазим от всички тези опасности, един от които е системата за управление на пакети (package manager). Тя ни позволява да намираме и инсталираме софтуер безопасно под формата на пакети, валидирани от истински хора и съхранявани в сигурни хранилища. Такива системи са например winget за Windows, apt за Ubuntu и homebrew за MacOS.

Системите за управление на пакети обаче могат да бъдат полезни и за самите разработчици на софтуер. Една от основните им функции е да знаят от какви други програми (библиотеки) е зависима дадена програма и съответно да управляват тези зависимости. Големите софтуерни проекти са съставени от множество модули, които зависят едни от други. Освен това всеки модул има различни версии, всяка от които може да предоставя различни функционалности. Някои модули зависят от точно определена версия на други модули. Така се създава една доста заплетена мрежа от зависимости. Ако трябваше програмистът ръчно да се грижи за всичко това, работата му щеше да се удвои. За щастие почти всеки език за програмиране в днешно време има своя система за управление на пакети. Езикът Python има системата `pip`, езикът JavaScript има системата `npm`, езикът Java има две такива системи - `maven` и `gradle`, и дори Rust, език за системно програмиране, сравним със C и C++, има своя система, наречена `cargo`. За жалост едни от езиците, които нямат такава общоприета система са именно C и C++. Има опити да се направи, но въпреки това голяма част от програмистите предпочитат да се грижат за зависимостите на проектите си ръчно. Една от причините за това е факта, че и двата езика са създадени преди повече от 30 години и съответно следват по-различни практики от съвременните езици за програмиране. И все пак C и C++ продължават да бъдат стандартизирани и до ден днешен, като и двата езика постепенно се сдобиват с нови по-модерни функционалности.

Текущият проект цели реализирането на система за управление на пакети за програмните езици C и C++. Самата система е написана на C и C++, като предоставя конзолен потребителски интерфейс, посредством който могат да се инсталират, премахват и обновяват пакети под формата на хранилища, взети от онлайн платформата GitHub. Системата също така позволява управление на зависимостите на даден пакет, както и създаването на изцяло нови пакети, описани чрез специален файлов формат.

Глава 1

Методи и технологии за реализиране на

C/C++, CMake, sqlite

Подобни проекти

- Conan
- vcpkg

Глава 2

Проектиране на структурата на

клиент, регистър

Глава 3

Програмна реализация на

cross-platform команден интерфейс

Глава 4

Ръководство за потребителя

```
$ cpm <install|remove|update|list> [options]
```

Заключение

Глава 5

Исползвана литература

- <https://medium.com/@sdboyer/so-you-want-to-write-a-package-manager->

Съдържание

1	Методи и технологии за реализиране на	3
2	Проектиране на структурата на	4
3	Програмна реализация на	5
4	Ръководство за потребителя	6
5	Използвана литература	8