

**Липецкий государственный технический университет**

**Факультет автоматизации и информатики**

**Кафедра автоматизированных систем управления**

**ЛАБОРАТОРНАЯ РАБОТА №4**

**по Технологии разработки ПО**

**Работа с менеджером пакетов Yarn**

Студент

Лобов М.Ю.

Группа АИ-18

Руководитель

Назаркин О.А.

Доцент

Липецк 2021 г.

Цель работы

Изучить принцип работы с менеджерами пакетов на примере одного из них (на выбор студента).

Задание кафедры

Рассмотреть и описать возможности работы с выбранным менеджером пакетов.

## Ход работы

Yarn — это альтернативный npm-клиент для работы в качестве пакетного менеджера JavaScript, совместно созданный Facebook, Google, Exponent и Tilde. Этот менеджер пакетов ускоряет сборку пакетов и делает её более безопасной. Последовательная установка пакетов через npm заметно замедляет работу. Yarn же поддерживает параллельную установку, что обычно в несколько раз быстрее. Yarn работает с теми же пакетами что и npm. И решает те же задачи, что и npm. Только делает это несколько быстрее. На 30% — 50%, если сравнивать yarn и npm после 5-й версии. Yarn работает с тем же файлом package.json, и устанавливает пакеты в директорию node\_modules. То есть это полный, заменимый аналог npm.

Основные команды менеджера пакетов Yarn представлены ниже:

- 1) `npm i -g yarn` – установка Yarn с помощью менеджера пакетов npm;
- 2) `yarn --version | -v` – проверка версии;
- 3) `yarn set version latest` – обновление до последней версии;
- 4) `yarn help` – список доступных команд;
- 5) `yarn init` – инициализация проекта;
- 6) `yarn install` – установка зависимостей;
- 7) `yarn add [package-name]` – добавление зависимости;
- 8) `yarn upgrade [package-name]` – обновление зависимости;
- 9) `yarn remove [package-name]` – удаление зависимости;
- 10) `yarn global add/upgrade/remove [package-name]` – глобальная установка/обновление/удаление пакета;
- 11) `yarn list` – список установленных зависимостей;
- 12) `yarn info [package-name]` – информация о пакете.

Преимуществами Yarn'a перед другими пакетными менеджерами JavaScript (такими как npm) являются:

### 1) Скорость

Существует гипотеза, что скорость установки одних и тех же пакетов в Yarn и npm отличается в несколько раз в пользу выбранного мной пакетного менеджера. Проверим данное заявление на примере установки React.

Установим React с помощью Yarn:

```
$ yarn global add create-react-app --prefix /usr/local
yarn global v0.27.5
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Installed "create-react-app@1.4.0" with binaries:
  - create-react-app
warning No license field
Done in 2.59s.
```

Теперь проведём установку через npm:

```
$ npm install -g create-react-app
/usr/local/bin/create-react-app ->
/usr/local/lib/node_modules/create-react-app/index.js
+ create-react-app@1.4.0
added 80 packages in 9.422s
```

Yarn установил модули за 2.59 секунд, тогда как npm потребовалось 9.422 секунд. Yarn был быстрее в 3.63 раза.

### 2) Надёжность установок

Yarn умеет решать следующие проблемы:

А) Кратковременные проблемы компьютерной сети могут послужить причиной сбоя при получении пакетов.

Б) Результатом выхода новых версий пакетов могут быть изменения, несовместимые с другими пакетами и нарушающие работу приложения.

### 3) Возможность проверки лицензий

Для некоторых проектов необходимо соблюдение определенных лицензионных требований или просто создание отчета для собственных или внешних целей. Это легко осуществляется при помощи команды `yarn licenses ls`.

#### 4) Совместимость с npm

Yarn полностью совместим с npm, поскольку является просто другим клиентом, который работает с реестрами npm. В самом начале развития Yarn поддерживал Bower, однако немного спустя было принято решение прекратить поддержку Bower.

#### 5) Наличие нескольких реестров

Yarn может работать с несколькими типами реестров. Если вы просто добавляете пакет, то, по умолчанию, Yarn будет использовать свой реестр npm (который не является стандартным реестром npm). Однако, Yarn также может добавлять пакеты из других источников, например, удаленных репозиториях git.

Чтобы посмотреть текущий сконфигурированный реестр npm, нужно выполнить следующую команду:

```
$ yarn config get registry  
https://registry.yarnpkg.com
```

Для того чтобы установить другой тип реестра, используйте команду:

```
$ yarn config set registry <registry url>
```

Для добавления пакетов из разных источников используйте следующие команды для добавления:

```
# Configured npm registry  
yarn add <pkg-name>
```

```
# Local package  
yarn add file:./<path to local package directory>
```

```
# Remote tarball  
yarn add https://<path to compressed tarball>.tgz
```

```
# Remote git repo  
yarn add <git remote-url>
```

## Вывод

В ходе выполнения лабораторной работы были рассмотрены основы работы с менеджером пакетов Yarn.