Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Высшая школа кибернетики и цифровых технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

ПО КУРСУ «ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ»

«АВТОМАТИЗАЦИЯ СБОРКИ И РАЗВЕРТЫВАНИЯ (CI/CD)»

Выполнил: студент 2 курса группы ПО(б) – 31

Черных Данил Анатольевич

Проверила: ассистент ВШ КЦТ

Балабасова Алиса Дмитриевна

Хабаровск 2025 г.

Задание:

1. Автоматизировать любой проект с помощью GitHub Actions.

2. Автоматизировать любой проект с помощью Jenkins (или другого CI/CD на выбор).

3. Проверить работу при внесении изменений (скриншоты до/после).

Ход работы:

GitHub Actions работает следующим образом:

1. Запуск процесса сборки и развертывания происходит при пуше нового тега в репозиторий, также возможен запуск вручную.
2. Сам процесс состоит из трех работ:
   1. Работа “create\_tag” используется для создания нового тега при запуске вручную.
   2. Работа “build” собирает программу для Windows и Linux, архивирует их в архивы (.zip и .tag.gz) и загружает эти архивы как артефакты для передачи следующей работе.
   3. Работа “release” отвечает за развертывание программы. Она скачивает артефакты (архивы) и генерирует список изменений, после чего создает GitHub релиз.

В файл README.md добавим бейджи статуса сборки, количества коммитов с последнего релиза, а также бейдж даты последнего коммита (рисунок 1):

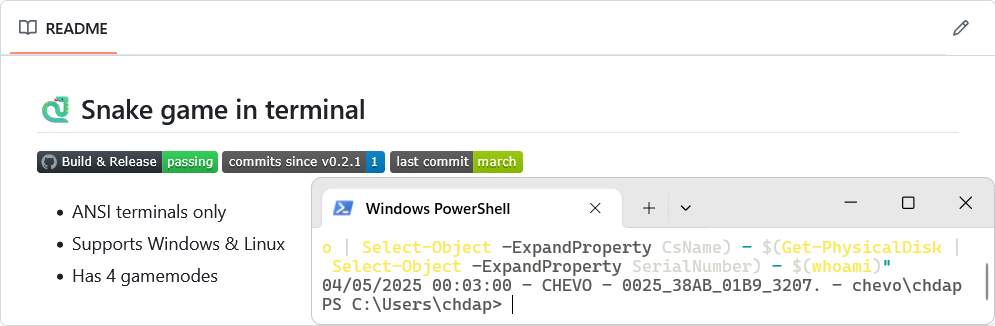


Рисунок 1 – Бейджи в readme GitHub

Jenkins работает схожим образом, но выполняет развертывание в локальное окружение.

Jenkins связан с репозиторием GitHub и проверяет его на изменения каждые пять минут и собирает новый релиз для самого последнего тега.

В Jenkins также возможно создать новую версию и запустить процесс сборки вручную, в таком случае Jenkins запушит новый тег в репозиторий GitHub, что запустить сборку с помощью GitHub Actions.

Проверим работу Jenkins и GitHub Actions при внесении изменений и создании новой версии.

До внесения изменений последней версией была “v0.2.1” и репозиторий выглядел следующим образом (рисунок 2):

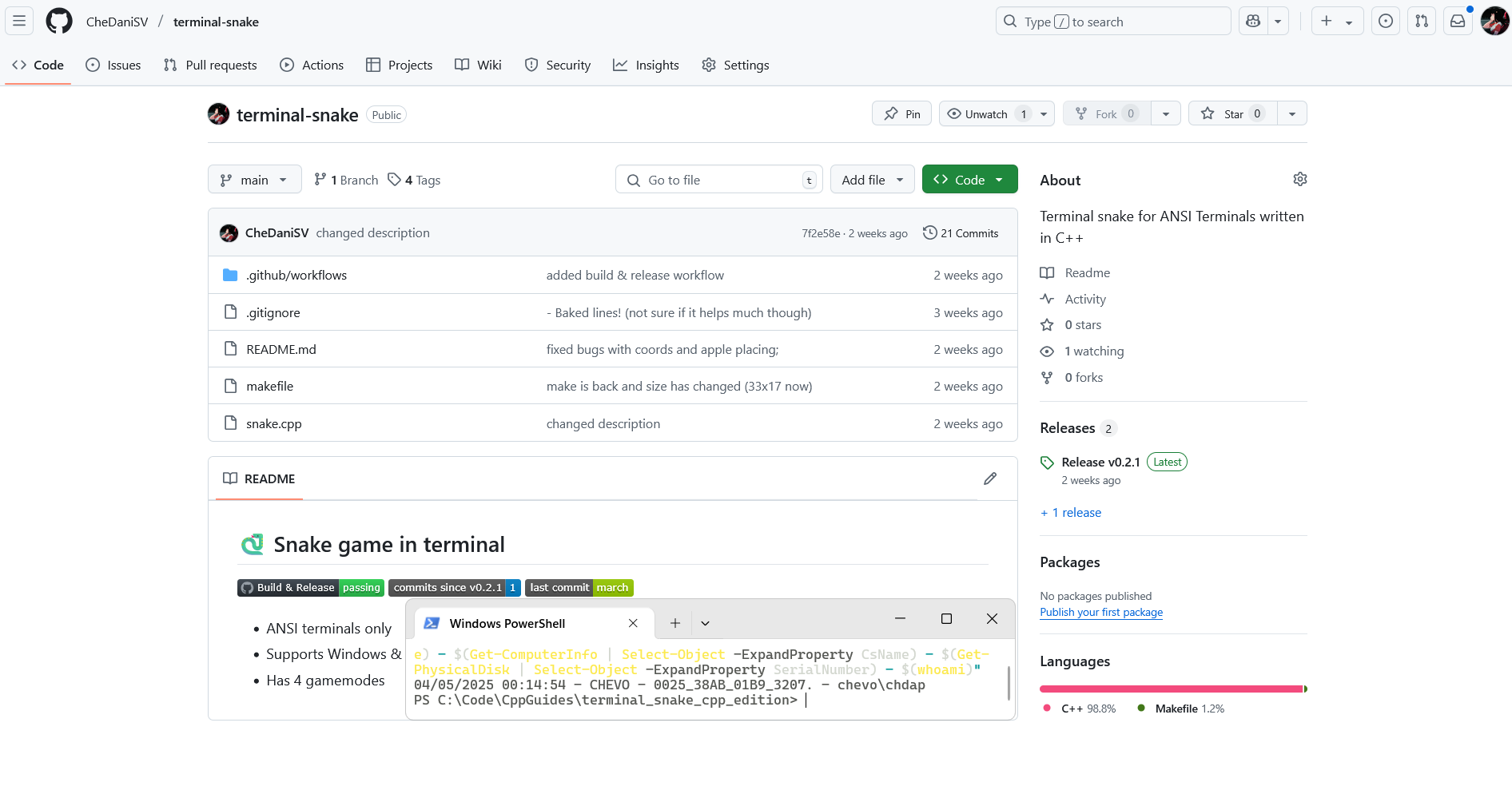


Рисунок 2 – Репозиторий до внесения изменений

Внесем изменения и создадим новую версию (тег) “v0.2.2” (рисунок 3):

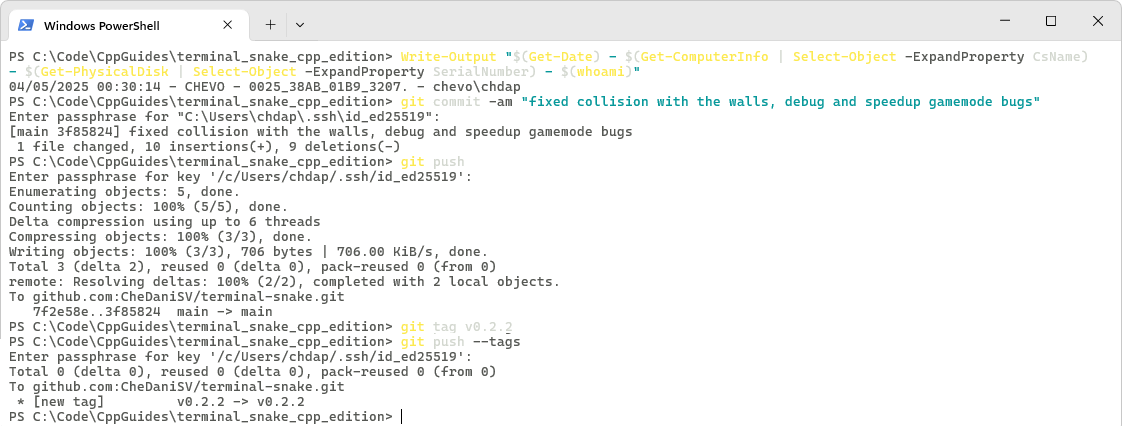


Рисунок 3 – Внесение изменений и добавление нового тега

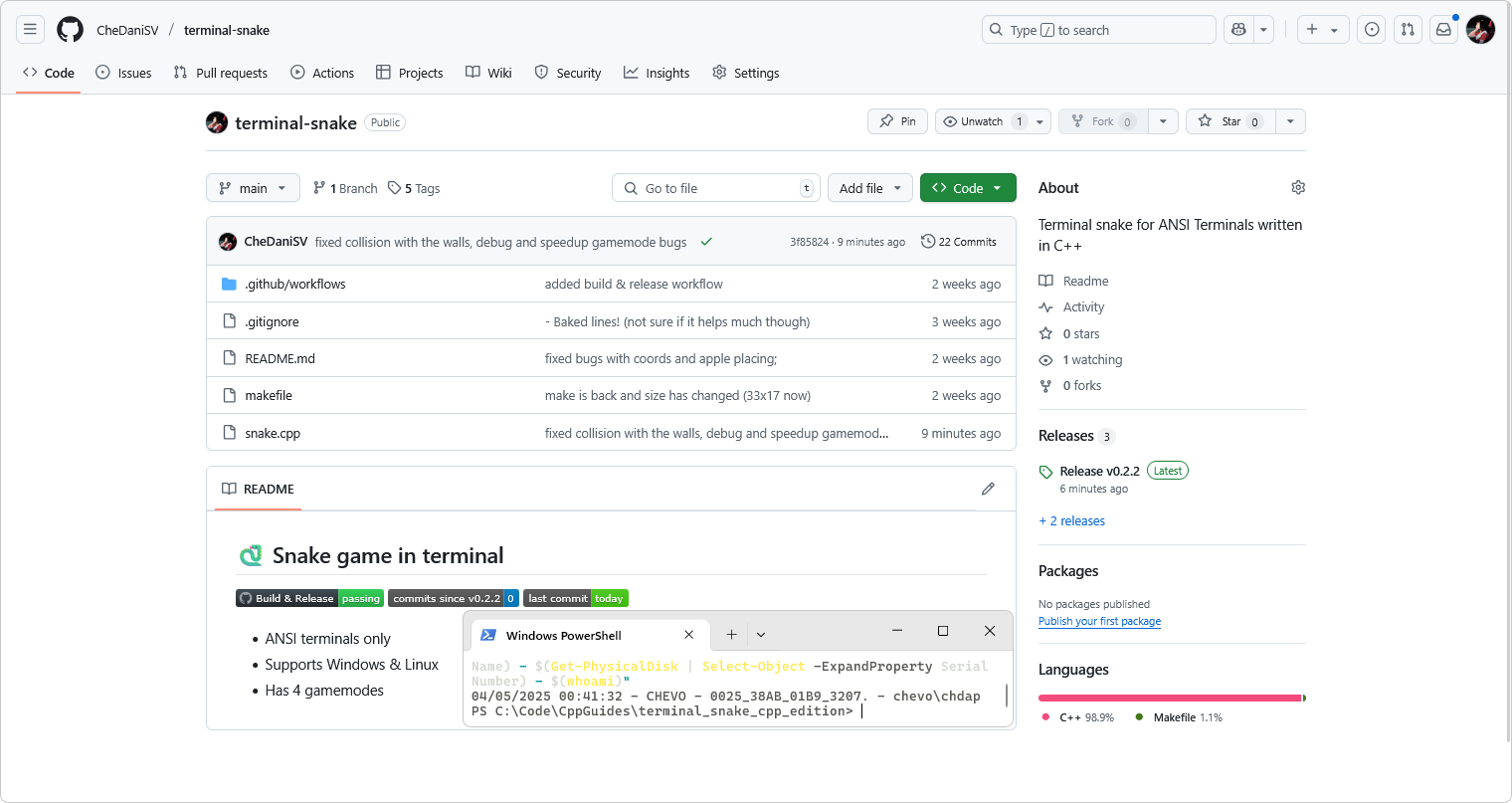


Рисунок 4 – Репозиторий после внесения изменения и работы GitHub Actions

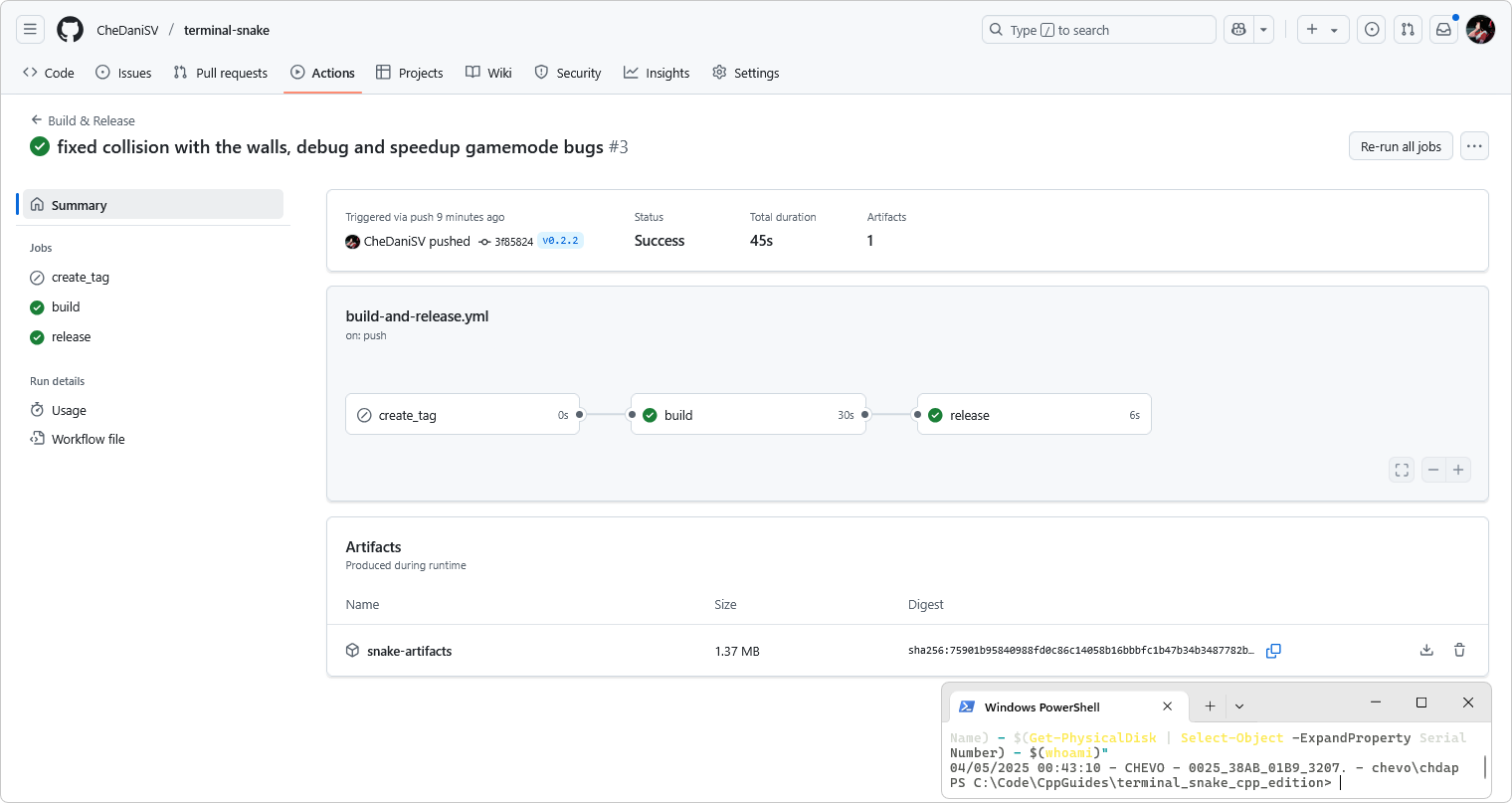


Рисунок 5 – Сводка результата выполнения GitHub Actions

Можно видеть, что работа “create\_tag” (рисунок 5) была пропущена, так как процесс сборки был запущен из-за пуша нового тега, а не вручную.

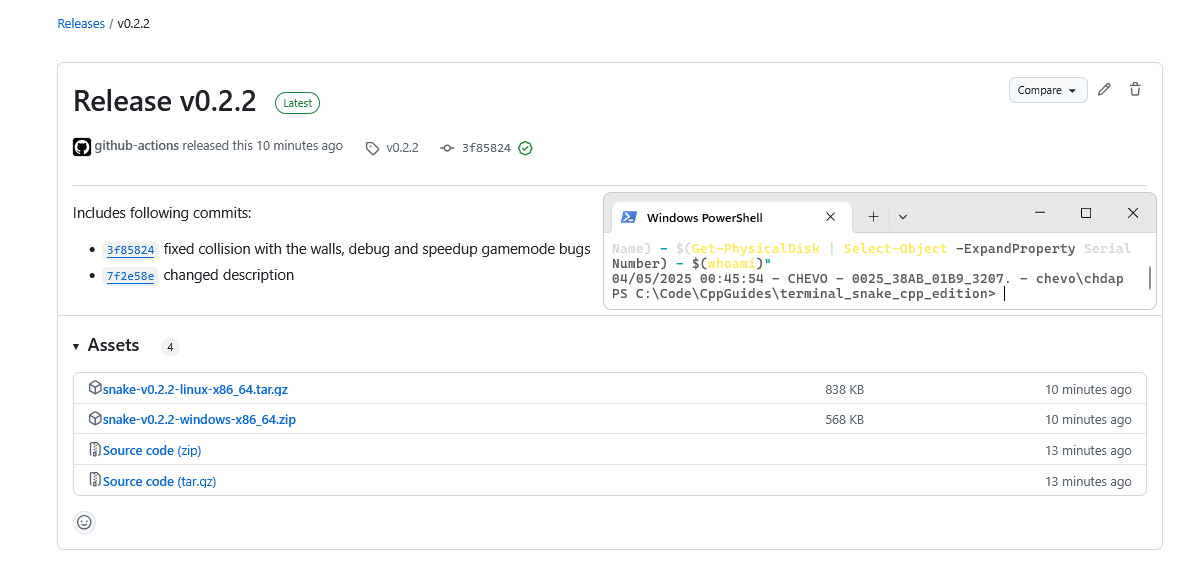


Рисунок 6 – Страница развернутого релиза



Рисунок 7 – Страница статуса выполнения Jenkins

В статусе выполнения работы Jenkins (рисунок 7) можно видеть, что работы была запущена автоматически из-за изменения в SCM.

Развертывание происходит в локальную папку на рабочем столе (рисунок 8).

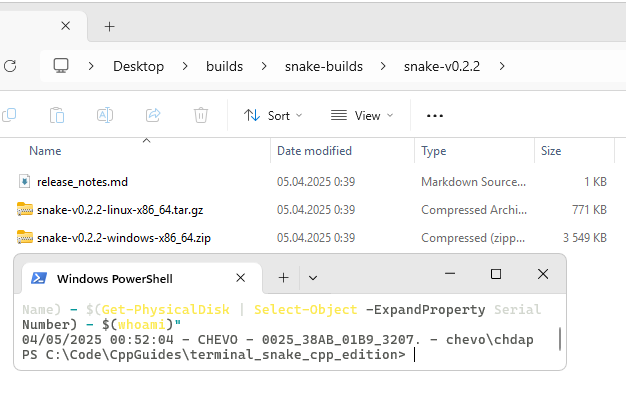


Рисунок 8 – Папка с развернутым локально релизом Jenkins

Вывод:

В ходе работы с использованием GitHub Actions и Jenkins были реализована автоматическая сборка и развертывание проекта, как в локальное, так и облачное окружение.

Опыт, полученный в ходе выполнения данной работы, будет полезен при использовании систем CI/CD в будущем.