

# Автоматизированная обертка в Docker для лабораторных работ курса 6.1810

Итерация 2: 27.02.2025 - 26.03.2025

# План на Итерацию 2

## **1. Интеграция скриптов в Docker:**

- Интегрировать все скрипты в Docker-контейнер для удобства развертывания.

## **2. Доработка скрипта по запуску лабораторной работы:**

- Улучшение логирования с помощью перехода на logging.
- Изучения команд, которые запускают лабораторные работы.
- Добавление функционала запуска проверки лабораторной работы в скрипт.
- Добавление функционала распаковки архива и применение патча.

## **3. Доработка скрипта тривиальных проверок:**

- Улучшение логирования с помощью перехода на logging.
- Изменения логики проверки с одного файла на несколько.

## **4. Написание тестов для проверки корректности работы.**

## **5. Подготовить инструкции скриптов для установки и развертывания приложения.**

## **6. Обновление и улучшение документации.**

# Результаты Итерации 2

За текущую итерацию был значительно улучшен проект, сосредоточенный на автоматизации работы с лабораторными работами для xv6. Основной акцент был сделан на автоматизации, улучшении системы запуска и логирования, а также интеграции с Docker.

## Что было сделано:

### 1. Автоматизация обработки лабораторных работ:

- Скрипт поддерживающий распаковку архивов (например, .zip, .tar.gz), автоматическое применение патчей (например, с помощью patch), и ведение логов в файл.
- Доработан скрипт запускающий лабораторную работу, ограничивающий время выполнения (таймаут) и перехватывающий stdout и stderr, записывая их в логи.

```
valeria@valeria-VirtualBox:~/Загрузки/mse1h2025-xv6-main (3)$ python3 scripts/load.py solution/xv6-labs-2024-util.zip
2025-03-26 18:04:28,354 - INFO - Logging to /home/valeria/Загрузки/mse1h2025-xv6-main (3)/logs/load.log
2025-03-26 18:04:28,355 - INFO - Temporary directory created: /tmp/tmp7jcgec43
2025-03-26 18:04:28,355 - INFO - Extracting archive: solution/xv6-labs-2024-util.zip
2025-03-26 18:04:28,358 - INFO - Archive successfully extracted to /tmp/tmp7jcgec43
2025-03-26 18:04:28,358 - INFO - Searching for patch file in /tmp/tmp7jcgec43
2025-03-26 18:04:28,371 - WARNING - No patch file found!
2025-03-26 18:04:28,371 - WARNING - No patch found. Skipping patch application.
2025-03-26 18:04:28,371 - INFO - Output directory set to: /home/valeria/Загрузки/mse1h2025-xv6-main (3)/lab_ready
2025-03-26 18:04:28,384 - INFO - Lab work successfully prepared in: /home/valeria/Загрузки/mse1h2025-xv6-main (3)/lab_ready
2025-03-26 18:04:28,384 - INFO - Script directory: /home/valeria/Загрузки/mse1h2025-xv6-main (3)/scripts
2025-03-26 18:04:28,384 - INFO - Lab ready directory: /home/valeria/Загрузки/mse1h2025-xv6-main (3)/lab_ready
```

Запуск распаковки архива без патча

```
valeria@valeria-VirtualBox:~/Рабочий стол/scripts$ python3 run_tests.py
Лог-файл: /home/valeria/Рабочий стол/logs/qemu-gdb.log
Working directory: /home/valeria/Рабочий стол/xv6-labs-2024-util/xv6-labs-2024-util
== Test sleep, no arguments ==
$ make qemu-gdb
sleep, no arguments: OK (2.9s)
== Test sleep, returns ==
$ make qemu-gdb
sleep, returns: OK (0.6s)
== Test sleep, makes syscall ==
$ make qemu-gdb
sleep, makes syscall: OK (0.3s)
== Test pingpong ==
$ make qemu-gdb
pingpong: OK (0.8s)
== Test primes ==
$ make qemu-gdb
primes: OK (1.7s)
== Test find, in current directory ==
$ make qemu-gdb
find, in current directory: OK (0.4s)
== Test find, in sub-directory ==
$ make qemu-gdb
find, in sub-directory: OK (0.9s)
== Test find, recursive ==
$ make qemu-gdb
find, recursive: OK (1.3s)
== Test xargs ==
$ make qemu-gdb
xargs: OK (1.4s)
== Test xargs, multi-line echo ==
$ make qemu-gdb
xargs, multi-line echo: OK (0.5s)
== Test time ==
time: OK
Score: 110/110
```

Скрипт запуска лабораторной работы

# Результаты Итерации 2

## 2. Создание системы запуска и проверки решений:

- Интерактивный CLI заменён на аргументы командной строки ([Ссылка на новый CLI](#))
- Убраны интерактивные запросы, все действия выполняются через параметры командной строки.
- Скрипт принимает на вход имя лабораторной работы, запускает её, проверяет результаты (например, сравнивает вывод с эталонным), обрабатывает ошибки и генерирует отчет.

## 3. Все скрипты интегрированы в Docker-контейнер, обеспечена автоматизация их запуска

- Создан Dockerfile, который собирает образ с необходимыми зависимостями (Python, патчи, xv6 и т.д.). Внутри контейнера скрипты доступны для запуска. ([Ссылка на скринкаст](#) – здесь же можно увидеть логи которые выводит данный скрипт)

## 4. Проведён запуск тестов, проверены зависимости, задокументированы результаты. ([Ссылка на результаты тестирования](#))

## 5. Изменена логика скрипта проверки файлов тривиальных проверок

## 6. Улучшение логирования и безопасности:

- Внедрён стандартный модуль logging, добавлены уровни логирования.

```
valeria@valeria-VirtualBox:~/Зарпужки/mse1h2025-xv6-main (3)$ bash scripts/run.sh --validate util solution/xv6-labs-2024-util.zip
Uploading the solution...
The file has been uploaded successfully!
Checking the solution...
The results of the check are saved to a file "/home/valeria/Зарпужки/mse1h2025-xv6-main (3)/logs/xv6-labs-2024-util.zip.json".
```

Запуск с флагом --validate

```
valeria@valeria-VirtualBox:~/Зарпужки/mse1h2025-xv6-main (3)$ bash scripts/run.sh --validate util solution/xv6-labs-204-util.zip
Error: Archive 'solution/xv6-labs-204-util.zip' not found.
valeria@valeria-VirtualBox:~/Зарпужки/mse1h2025-xv6-main (3)$ bash scripts/run.sh --validate fdf solution/xv6-labs-2024-util.zip
Error: A non-existent laboratory work is indicated
```

Запуск с флагом --validate с некоректными параметрами

```
valeria@valeria-VirtualBox:~/Зарпужки/mse1h2025-xv6-main (3)$ bash scripts/run.sh --help
=====
Automated system
verification of Lab 6.828
=====
USE: ./run.sh [KEYS]... [TARGET]...
KEYS (flags):
  --help           The output of this instruction
  --validate [1] [2] Download and verify the solution
  --report [1] [2]  Show the verification results
PURPOSE:
  [1] _the_name_of_the_lab_
  [2] _the_name_of_the_uploaded_archive_
List of laboratory work names:
  util syscall pgtbl traps cow net lock fs mmap
=====
```

Запуск с флагом --help

# План на Итерацию 3

1. Разработка тестов:
  - Реализовать интеграционные и функциональные тесты для базовых сценариев использования.
  - Внедрить тесты через GitHub Actions для автоматизации проверки кода.
2. Настройка CI/CD пайплайнов для автоматического тестирования и деплоя, используя GitHub Actions или аналогичные инструменты.
  - Настроить автоматический процесс тестирования и деплоя с использованием GitHub Actions или аналогичных инструментов.
  - Обеспечить интеграцию с репозиторием для автоматического запуска тестов и деплоя при каждом коммите.
3. Произвести устранение найденных багов, улучшить сообщения об ошибках и логирование процессов.
4. Провести улучшение обработки ошибок и добавить дополнительные проверки для разных типов ошибок, возникающих при запуске лабораторных работ.
5. Обновить документацию по Docker-контейнеру и скриптам.
6. Генерация отчета из логов:
  - Разработать систему для автоматической генерации отчетов на основе логов работы системы.
  - Обеспечить удобный формат отчетов, включающий информацию о ходе выполнения лабораторных работ, выявленных ошибках и обработке ошибок.
7. Настройка Docker с учётом безопасности:
  - Провести настройку Docker-контейнера с учетом безопасности: использовать минимальные базовые образы, ограничить права пользователя внутри контейнера, настроить защиту от уязвимостей и обновлений.