

Автоматизированная обертка в Docker для лабораторных работ курса 6.1810

Итерация 3: 27.03.2025 - 12.05.2025

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.

Результаты Итерации 3

Что было сделано:

1. Настройка Docker с учетом безопасности:

- Создан безопасный Docker-образ для запуска xv6.
- Реализованы меры по ограничению прав доступа и изоляции среды.
- Образ протестирован и готов к использованию в CI/CD.

2. Автоматизация применения патчей:

- Внедрено автоматическое получение исходного кода xv6 из репозитория GitHub, вместо ручной загрузки архива пользователем.
- Обновлён Python-скрипт: теперь он автоматически клонирует нужный коммит, применяет патчи и собирает проект.
- Обеспечена полная совместимость с текущей структурой репозитория, что упрощает CI и ускоряет тестирование изменений.

3. Генерация отчётов

- Реализован сценарий с выводом и генерацией финального отчёта по проверки лабораторной работы

4. Разработка тестов и настройка CI/CD

- Разработаны автоматические тесты для проверки стабильности системы.
- Настроены процессы CI/CD для автоматического запуска тестов при каждом коммите.

5. Обновление документации

- Добавлено подробное руководство по запуску проекта в Docker.
- Документация обновлена с учетом последних изменений в инфраструктуре.

Результаты Итерации 3

```
PS C:\Users\pankl\VandexDisk\V4e6a\2 семестр\Новая nanka\mseih2025-xv6> docker-compose -f docker/docker-compose.yml run --rm app --validate util solution/0001-util_110of110_git_format-patch.zip
Uploading the solution...
The file has been uploaded successfully!
Checking the solution...

The results of the check are saved to a file "/app/logs/0001-util_110of110_git_format-patch.zip.json".
```

Запуск с флагом --validate

```
PS C:\Users\pankl\VandexDisk\V4e6a\2 семестр\Новая nanka\mseih2025-xv6> docker-compose -f docker/docker-compose.yml run --rm app --help

=====
Automated system
verification of Lab 6.828
=====
USE: ./run.sh [KEYS]... [TARGET]...
KEYS (flags):
--help           The output of this instruction
--validate [lab] [archive] Download and verify the solution
--report [lab] [archive] Show the verification results
PURPOSE:
[lab] - the name of the lab
[archive] - the full name of the uploaded archive
              (the path to it and it's name)

List of laboratory work names:
util syscall pgtbl cow net lock fs mmap

Valid archive format: .zip
The archive should contain a single patch file obtained after making changes to the previously cloned repository xv6-labs-2024 (git clone git://g.csail.mit.edu/xv6-labs-2024).
=====
```

Запуск с флагом --help

```
PS C:\Users\pankl\VandexDisk\V4e6a\2 семестр\Новая nanka\mseih2025-xv6> docker-compose -f docker/docker-compose.yml run --rm app --report util solution/0001-util_110of110_git_format-patch.zip
Found the file "/app/logs/0001-util_110of110_git_format-patch.zip.log".
Checking the archive format:
2025-05-12 15:06:21,883 - INFO - Logging to /app/logs/0001-util_110of110_git_format-patch.zip.log
2025-05-12 15:06:21,884 - INFO - Using provided lab branch: util
2025-05-12 15:06:21,884 - INFO - Extracting patch from archive: solution/0001-util_110of110_git_format-patch.zip
2025-05-12 15:06:21,886 - INFO - Copied patch to: /app/lab_ready/0001-util_110of110_git_format-patch.patch
2025-05-12 15:06:21,886 - INFO - Cloning xv6 repository from git://g.csail.mit.edu/xv6-labs-2024
2025-05-12 15:06:26,048 - INFO - Successfully cloned xv6 repository
2025-05-12 15:06:26,049 - INFO - Checking out lab branch: util
2025-05-12 15:06:26,054 - INFO - Checked out branch util, commit: 79e8024d61c3dd92ccf064fc67bb99cb999f70ab
2025-05-12 15:06:26,054 - INFO - Applying patch /app/lab_ready/0001-util_110of110_git_format-patch.patch to xv6 repository
2025-05-12 15:06:26,056 - INFO - Patch applied successfully
2025-05-12 15:06:26,057 - INFO - Checking if Makefile is modified in patch: /app/lab_ready/0001-util_110of110_git_format-patch.patch
2025-05-12 15:06:26,058 - INFO - Patch affects the following files:
Makefile      | 7 +++-
time.txt      | 1 +
user/find.c   | 67 ++++++
user/pingpong.c | 42 ++++++
user/primes.c | 58 ++++++
user/sleep.c  | 17 ++++++
user/xargs.c  | 85 ++++++
7 files changed, 275 insertions(+), 2 deletions(-)

2025-05-12 15:06:26,058 - INFO - Makefile is modified by the patch.
2025-05-12 15:06:26,058 - INFO - Successfully applied patch to xv6 repository at /app/lab_ready/xv6-labs-2024
2025-05-12 15:06:26,058 - INFO - Lab branch: util
2025-05-12 15:06:26,058 - INFO - You can now run the lab tests in the xv6 repository
2025-05-12 15:06:26 - INFO -

Trivial checks:
Logging started. Logs are saved to: /app/logs/0001-util_110of110_git_format-patch.zip.log
2025-05-12 15:06:26 - INFO - Starting file check in 'user' folders within: /app/lab_ready
2025-05-12 15:06:26 - INFO -
Checking folder: /app/lab_ready/xv6-labs-2024/user
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/ls.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/initcode.S
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/xangstest.sh
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/nc.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/grep.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/user.h
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/umalloc.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/xargs.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/find.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/sh.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/zombie.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/pingpong.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/primes.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/usertests.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/grind.c
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/user.ld
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/usys.pl
2025-05-12 15:06:26 - INFO - Checking file: /app/lab_ready/xv6-labs-2024/user/sleep.c
```

Запуск с флагом --report

Результаты Итерации 3

```
Error: Archive 'solution/0001-util_110of110_git_format-patch.zip' not found.
```

Запуск с несуществующим архивом

```
2025-05-12 17:22:59,363 - INFO - Logging to /app/logs/1.zip.log
2025-05-12 17:22:59,363 - INFO - Using provided lab branch: util
2025-05-12 17:22:59,363 - INFO - Extracting patch from archive: solution/1.zip
2025-05-12 17:22:59,365 - ERROR - No patch file found in archive!
2025-05-12 17:22:59,365 - ERROR - Failed to extract patch file from archive
Error loading the solution. Details in the log: /app/logs/1.zip.log
```

Запуск с архивом без патч файла

```
2025-05-12 17:27:07,512 - INFO - Logging to /app/logs/11.zip.log
2025-05-12 17:27:07,513 - INFO - Using provided lab branch: util
2025-05-12 17:27:07,513 - INFO - Extracting patch from archive: solution/11.zip
2025-05-12 17:27:07,515 - INFO - Copied patch to: /app/lab_ready/0001-util_110of110_git_format-patch.patch
2025-05-12 17:27:07,515 - INFO - Cloning xv6 repository from git://g.csail.mit.edu/xv6-labs-2024
2025-05-12 17:27:12,546 - INFO - Successfully cloned xv6 repository
2025-05-12 17:27:12,546 - INFO - Checking out lab branch: util
2025-05-12 17:27:12,553 - INFO - Checked out branch util, commit: 79e8024d61c3dd92ccf064fc67bb99cb999f70ab
2025-05-12 17:27:12,553 - INFO - Applying patch /app/lab_ready/0001-util_110of110_git_format-patch.patch to xv6 repository
2025-05-12 17:27:12,555 - INFO - Patch applied successfully
2025-05-12 17:27:12,555 - INFO - Checking if Makefile is modified in patch: /app/lab_ready/0001-util_110of110_git_format-patch.patch
2025-05-12 17:27:12,557 - INFO - Patch affects the following files:
time.txt | 1 +
user/find.c | 67 ++++++
user/pingpong.c | 42 ++++++
user/primes.c | 58 ++++++
user/sleep.c | 17 ++++++
user/xargs.c | 85 ++++++
6 files changed, 270 insertions(+)

2025-05-12 17:27:12,557 - ERROR - Add your sleep program to UPROGS in Makefile; once you've done that, make qemu will compile your program and you'll be able to run it from the xv6 shell.
2025-05-12 17:27:12,557 - ERROR - Patch does not change the Makefile. Exiting.
Error loading the solution. Details in the log: /app/logs/11.zip.log
```

Запуск с архивом с некорректным* патч файлом

*некорректный патч файл в данном контексте это патч файл в котором в makefile были не включены написанные программы, то есть при запуске xv6 возникнет ошибка, чтобы это предотвратить это прерывается на моменте загрузки файла

План на Итерацию 4

1. Проблема с парсингом:

- В ходе тестов, было выяснено что скрипты с парсингом не проходят их. Нужно это пофиксить

2. Написание финальных тестов для проверки корректности работы.

- Написать недостающие юнит- и интеграционные тесты.
- Документировать запуск тестов..

3. Провести внутренний код-ревью.

- Организовать ревизию кода внутри команды.
- Проверить читаемость, соответствие соглашениям, убрать дублирование.

4. Интеграция с системой MOOC-Linux-Programming.

- Провести созвон с заказчиком для выяснения необходимости интеграции.
- Уточнить: Требуется ли включение xv6 в систему MOOC.
- Какой формат данных или функциональность нужна.
- При необходимости: разработать черновой план интеграции (в отдельную задачу на следующую итерацию).

После проведения созвона с заказчиком и получения правок, план может дополняться.