Лабораторная работа №1

UNIX знакомство: useradd, nano, chmod, docker, GIT, CI, CD

Выполнил: Макаренко Александр, группа Z33434, 3 курс, 2024 г.

Цель

Познакомить студента с основами администрирования программных комплексов в ОС семейства UNIX, продемонстрировать особенности виртуализации и контейнеризации, продемонстрировать преимущества использования систем контроля версий (на примере GIT)

Задачи

1 [ОС] Работа в ОС, использование файловой системы, прав доступа, исполнение файлов

1. В папке /USR/LOCAL/ создать 2 директории: folder_max, folder_min
2. Создать 2-х группы пользователей: group_max, group_min
3. Создать 2-х пользователей: user_max_1, user_min_1
4. Для пользователей из группы *_max дать полный доступ на директории *_max и *_min. Для пользователей группы *_min дать полный доступ только на директорию *_min
5. Создать и исполнить (пользователем из той же категории) скрипт в директории folder_max, который пишет текущую дату/время в файл output.log в текущей директории
6. Создать и исполнить (пользователем из той же категории) скрипт в директории folder_max, который пишет текущую дату/время в файл output.log в директории *_min
7. Исполнить (пользователем *_min) скрипт в директории folder_max, который пишет текущую дату/время в файл output.log в директории *_min
8. Создать и исполнить (пользователем из той же категории) скрипт в директории folder_min, который пишет текущую дату/время в файл output.log в директории *_max
9. Вывести перечень прав доступа у папок *_min/ *_max, а также у всего содержимого внутри

2 [КОНТЕЙНЕР] docker build / run / ps / images

2.1. Создать скрипт, который пишет текущую дату/время в файл output.log в текущей директории

2.2. Собрать образ со скриптами выше и с пакетом nano (docker build)

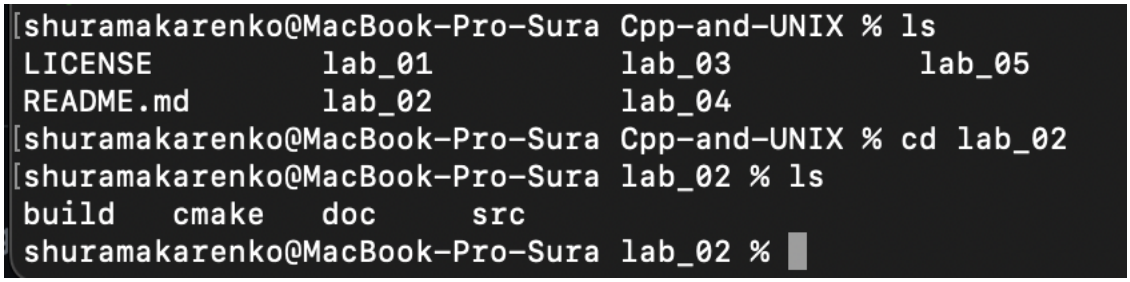
2.3. Запустить образ (docker run)

2.4. Выполнить скрипт, который подложили при сборке образа

2.5. Вывести список пользователей в собранном образе

3 [GIT] GitHub / GitLab, в котором будут содержаться все выполненные ЛР

1. Создать репозиторий в GitHub или GitLab https://github.com/broshura/Cpp-and-UNIX
2. Создать структуру репозитория:



3. Создать ветки dev / stg / prd, удалить ранее существующие ветки удаленно и локально
4. Создать скрипт автоматического переноса ревизий из ветки dev в ветку stg с установкой метки времени (tag). Скрипт в корень репозитория

4 [SAVE] Всё, что было сделано в шагах 1-3, сохранить в репозиторий (+ отчет по данной ЛР в папку doc). Фиксацию ревизий производить строго через ветку dev. С помощью скриптов накатить ревизии на stg и на prd.