МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет «ИТМО» (НИУ ИТМО)

**Лабораторная работа №1 «UNIX знакомство:**

**useradd, nano, chmod, docker, GIT, CI, CD»**

**по курсу «C++ и UNIX системы»**

|  |  |
| --- | --- |
| **Выполнил студент 3 курса группы К3333:** | Буданцев А.А. |
| **Проверил:** | Маслов И.Д. |

Оглавление

[Цель работы 3](#_Toc128832916)

[Задача 3](#_Toc128832917)

[Решение 5](#_Toc128832918)

[1. [ОС] Работа в ОС, использование файловой системы, прав доступа, исполнение файлов. 5](#_Toc128832919)

[2. [КОНТЕЙНЕР] docker build / run / ps / images 11](#_Toc128832920)

[3. [GIT] GitHub / GitLab, в котором будут содержаться все выполненные ЛР 13](#_Toc128832921)

# **Цель работы**

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).

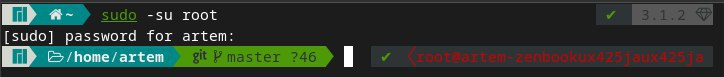
# **Задача**

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**
   1. Создать скрипт, который пишет текущую дату/время в файл output.log в текущей директории
   2. Собрать образ со скриптами выше и с пакетом nano (docker build)
   3. Запустить образ (docker run)
   4. Выполнить скрипт, который подложили при сборке образа
   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
   2. Создать структуру репозитория:
      1. lab\_01
         1. build
         2. src
         3. doc
         4. cmake (для ЛР 1 опционально)
      2. lab\_02
         1. ... идентично lab\_01 …
   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
   5. Создать скрипт автоматического переноса ревизий из ветки stg в ветку prd 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.**

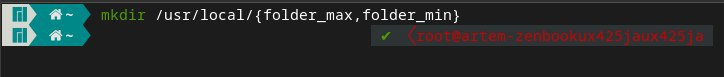
# **Решение**

## **[ОС] Работа в ОС, использование файловой системы, прав доступа, исполнение файлов.**

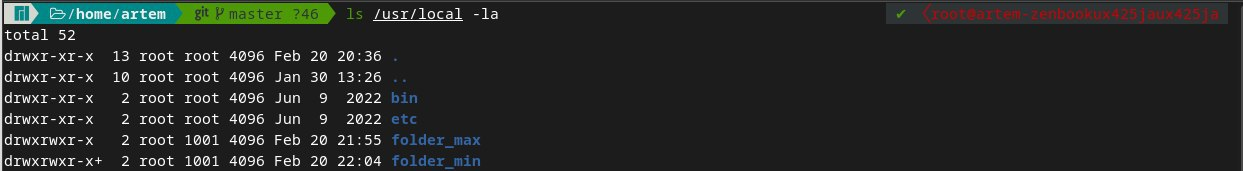
* 1. **В папке /USR/LOCAL/ создать 2 директории: folder\_max, folder\_min**

Заходим под правами суперпользователя 

Создаём каталоги *folder\_max, folder\_min*

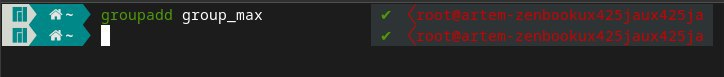


Просмотр созданных каталоги в */usr/local*

****

* 1. **Создать 2-х группы пользователей: group\_max, group\_min**

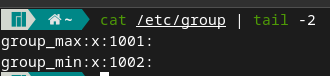
Добавляем группу *group\_max*



Добавляем группу *group\_min*



Просмотр групп из файла /etc/group

**

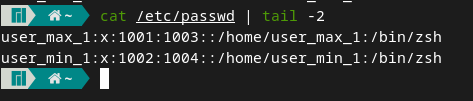
* 1. **Создать 2-х пользователей: user\_max\_1, user\_min\_1**

Добавляем пользователя *user\_max\_1* 

Добавляем пользователя *user\_min\_1*



Просмотр пользователей из файла /etc/passwd



* 1. **Для пользователей из группы \*\_max дать полный доступ на директории \*\_max и \*\_min. Для пользователей группы \*\_min дать полный доступ только на директорию \*\_min**

Добавляем пользователя *user\_max\_1* в группу *group\_max*



Добавляем пользователя *user\_min\_1* в группу *group\_ min*



Задаём группу-владельца *group\_max* для каталогов *folder\_m\** 

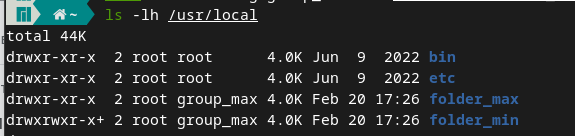
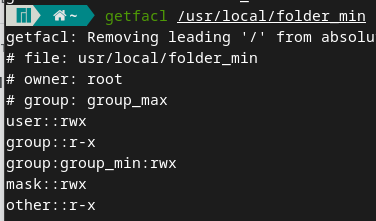
Добавляем дополнительную группу *group\_min* для каталога *folder\_min* и задаём ей права пользования



Даём полные права пользования для групп-владельцев каталогов *folder\_m\**



Проверяем владельцев директорий *folder\_max, folder\_min*

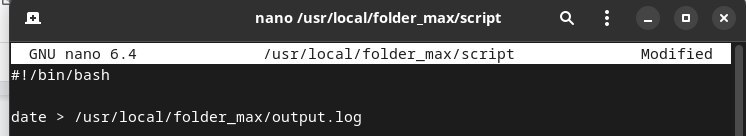
  


* 1. **Создать и исполнить (пользователем из той же категории) скрипт в директории folder\_max, который пишет текущую дату/время в файл output.log в текущей директории**

Создаём файл *script* и заходим в редактор *nano*



Пишем в *nano* скрипт:



Даём права доступа на исполнения для остальных пользователей



Исполнение файла *script*:



Читаем содержимое файла *output.log*



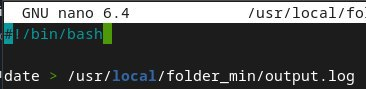
* 1. **Создать и исполнить (пользователем из той же категории) скрипт в директории folder\_max, который пишет текущую дату/время в файл output.log в директории \*\_min**

Копируем файл *script* и создаём новый *script1*



Редактируем файл *script1*





Устанавливаем права доступа для группы folder\_min



Исполнение файла *script1*:



Читаем содержимое файла *output.log*



* 1. **Исполнить (пользователем \*\_min) скрипт в директории folder\_max, который пишет текущую дату/время в файл output.log в директории \*\_min**

Удаляем предыдущие файлы *output.log* из каталогов *folder\_m\**

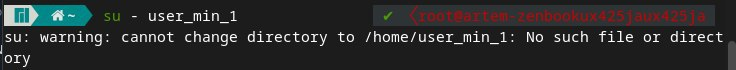




Добавляем право на выполнение для файла *script1* другим пользователям



Заходим под пользователем *user\_min\_1*



Исполнение файла *script1*:



Читаем содержимое файла *output.log*



* 1. **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**

Выходим из пользователя *user\_min\_1* и даём право на запись для каталога *folder\_max* для остальных пользователей



Снова заходим под пользователей *user\_min\_1* и копируем файл *script* из каталога *folder\_max* в каталог *folder\_min*



Исполнение файла *script*:

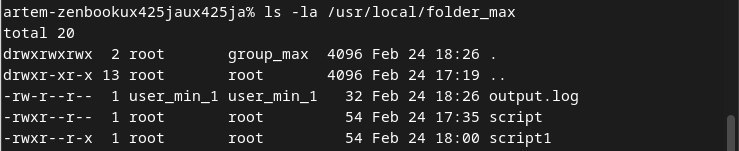


Читаем содержимое файла *output.log*

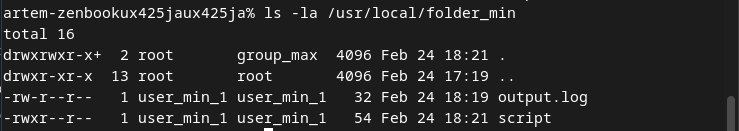


* 1. **Вывести перечень прав доступа у папок \*\_min/ \*\_max, а также у всего содержимого внутри**

Права доступа у каталога folder\_max и его содержимого



Права доступа у каталога folder\_max и его содержимого



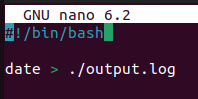
## **[КОНТЕЙНЕР] docker build / run / ps / images**

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

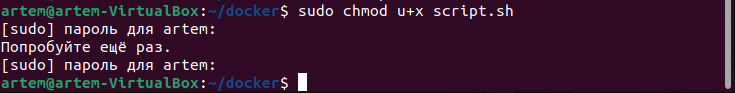
Создаём файл *script.sh* и заходим в редактор *nano*



Пишем в *nano* скрипт:

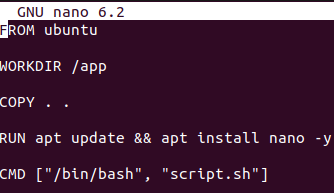


Даём права доступа на исполнения для остальных пользователей

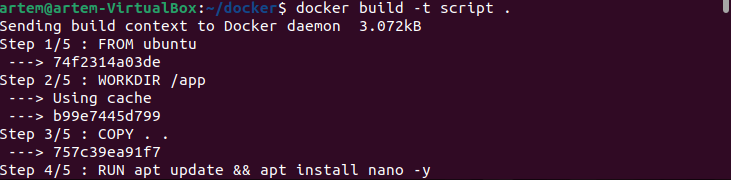


* 1. **Собрать образ со скриптами выше и с пакетом nano (docker build)**

Создаём Dockerfile и пишем в него инструкцию:

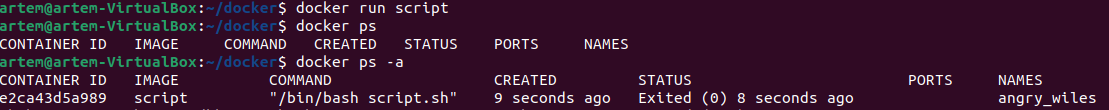


Создание образа



* 1. **Запустить образ (docker run)**

Запуск образа

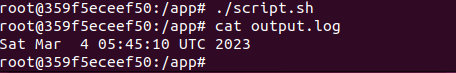


* 1. **Выполнить скрипт, который подложили при сборке образа**

Запускаем контейнер в интерактивном режиме и заходим в его оболочку

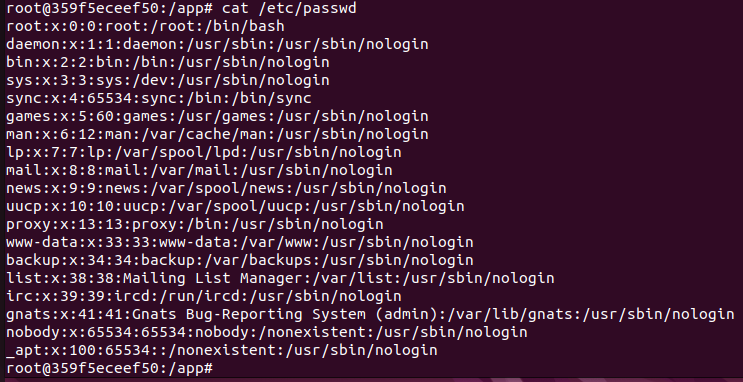


Запускаем скрипт в контейнере

****

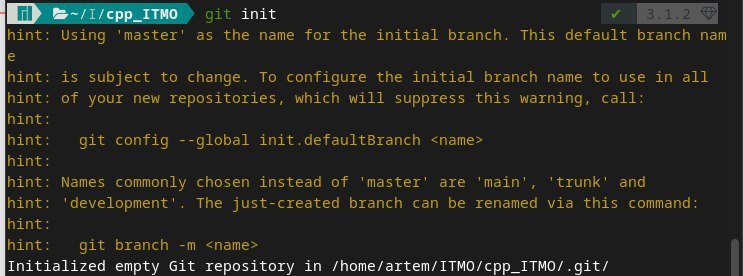
* 1. **Вывести список пользователей в собранном образе**

Вывод списка пользователей



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

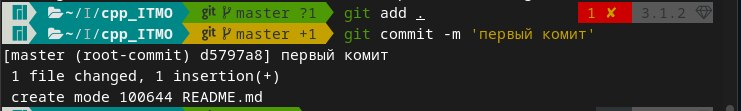
* 1. **Создать репозиторий в GitHub или GitLab**



Добавляем файл nano для доступа к ветке

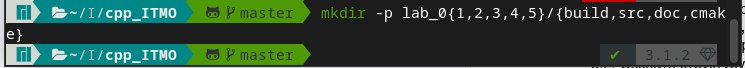


Делаем первый commit



* 1. **Создать структуру репозитория:**

Создаём директории

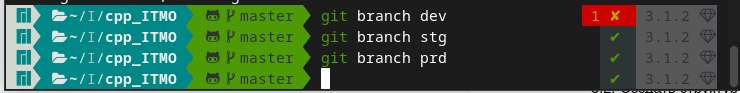


Добавляем файлы, чтобы директории были видимы для git:

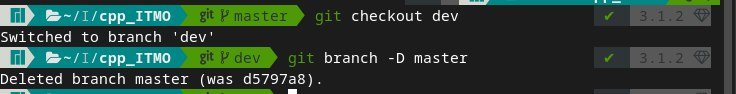
touch lab\_0{1,2,3,4,5}/{build,src,doc,cmake}/.gitkeep

* 1. **Создать ветки dev / stg / prd, удалить ранее существующие ветки удаленно и локально**

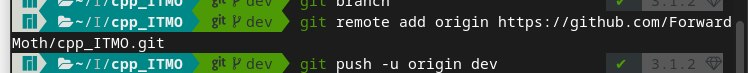
Добавляем новые ветки



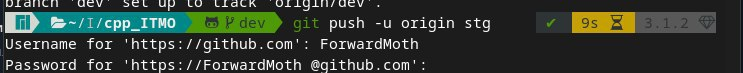
Удаляем ветку master

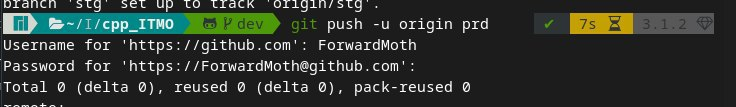


Делаем связь локального git с репозиторием на github

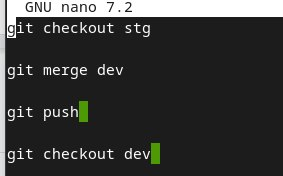


Добавление всех веток на github:





* 1. **Создать скрипт автоматического переноса ревизий из ветки dev в ветку stg с установкой метки времени (tag). Скрипт в корень репозитория**



* 1. **Создать скрипт автоматического переноса ревизий из ветки stg в ветку prd с установкой метки времени (tag). Скрипт в корень репозитория**

