



ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

[C++ UNIX]: UNIX знакомство: useradd, nano, chmod, docker, GIT, CI, CD

Выполнил студент:

Карсаков Григорий Вячеславович

3 курс, Z33434, 2023

Санкт-Петербург

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)

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

1.1 В папке /usr/local/ создать 2 директории: folder_max, folder_min

```
1 root@user:~# cd /usr/local
2 root@user: /usr/local# mkdir folder_max folder_min
3
4
5 root@user: /usr/local# ls
6 bin etc folder_max folder_min games include lib man sbin share src
```

Листинг 1: Команды и результат

1.2 Создать 2-х группы пользователей: group_max, group_min

```
1 root@user:~# groupadd group_max
2 root@user:~# groupadd group_min
3
4
5 root@user:~# getent group
6 ...
7 group_max
8 group_min
9 ...
```

Листинг 2: Команды и результат

1.3 Создать 2-х пользователей: user_max_1, user_min_1

```
1 root@user:/usr/local# useradd user_max_1
2 root@user:/usr/local# useradd user_min_1
3
4
5 root@user:/usr/local# usermod -a -G group_max user_max_1
6 root@user:/usr/local# usermod -a -G group_min user_min_1
7
8
9 root@user:/usr/local# members group_max
10 user_max_1
11 root@user:/usr/local# members group_min
12 user_min_1
```

Листинг 3: Команды и результат

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

```
1 root@user:/usr/local# chmod a=rwx folder_min
2 root@user:/usr/local# chmod o-w folder_max
3 root@user:/usr/local# chmod o-w folder_max
4
5 root@user:/usr/local# getfacl folder_max
6 # file: folder_max
7 # owner: root
8 # group: group_max
9 user::rwx
10 group::rwx
11 other::r-x
```

Листинг 4: Команды и результат

1.5 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

```
1 #!/bin/bash/
2 echo "date is $(date '+%d-%m-%Y-%H-%M-%S')" >> output.log
```

Листинг 5: date.sh

```
1 $ whoami
2 user_max_1
3 $ pwd
4 /usr/local/folder_max
5
6 $ bash date.sh
7
8
9 $ ls
10 date.sh  output.log
```

Листинг 6: Команды и результат

1.6 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

```
1 #!/bin/bash/
2 echo "date is $(date '+%d-%m-%Y-%H-%M-%S')" >> /usr/local/folder_min/
  output_from_max.log
```

Листинг 7: date_in_min.sh

```

1 $ whoami
2 user_max_1
3 $ pwd
4 /usr/local/folder_max
5
6 $ bash date_in_min.sh
7
8
9 $ cd ..
10 $ cd folder_min
11 $ ls
12 output_from_max.log

```

Листинг 8: Команды и результат

1.7 Исполнить (пользователем *_min) скрипт в директории folder_max, который пишет текущую дату/время в файл output.log в директории *_min

```

1 $ whoami
2 user_min_1
3 $ pwd
4 /usr/local/folder_max
5
6 $ bash date_in_min.sh
7
8 $ bash date_in_min.sh
9 $ cd ..
10 $ cd folder_min
11 $ ls
12 date.sh  output_from_max.log

```

Листинг 9: Команды и результат

1.8 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

```

1 $ whoami
2 user_min_1
3 $ pwd
4 /usr/local/folder_min
5
6 $ bash date_in_max.sh
7
8
9 $ cd ..
10 $ cd folder_max
11 $ nano output.log
12 %% date is 23-02-2023-15-16-04
13 %% date from min is 23-02-2023-16-31-33

```

Листинг 10: Команды и результат

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

```
1 $ ls -l
2 total 40
3 ...
4 drwxrwxr-x 2 user_max_1 group_max 4096 фев 23 16:28 folder_max
5 drwxrwxrwx 2 user_min_1 group_max 4096 фев 23 16:31 folder_min
6 ...
7 $ cd folder_max
8 $ ls -l
9 total 12
10 -rwxrwxr-x 1 user_max_1 user_max_1 101 фев 23 15:22 date_in_min.sh
11 -rwxrwxr-x 1 user_max_1 user_max_1 71 фев 23 14:57 date.sh
12 -rw-rw-rwx 1 user_max_1 user_max_1 65 фев 23 16:31 output.log
13 $ cd ..
14 $ cd folder_min
15 $ ls -l
16 total 12
17 -rwxrwxr-x 1 user_min_1 user_min_1 101 фев 23 16:31 date_in_max.sh
18 -rwxrwxrwx 1 user_min_1 user_min_1 16 фев 23 15:30 date.sh
19 -rwxrwxrwx 1 user_max_1 user_max_1 84 фев 23 16:13 output_from_max.log
```

Листинг 11: Команды и результат

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

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

```
1 #!/bin/bash/
2 echo "date is $(date '+%d-%m-%Y-%H-%M-%S')" >> output.log
```

Листинг 12: date.sh

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

```
1 touch Dockerfile
2 nano Dockerfile
```

Листинг 13: Команды и результат

```
1 FROM ubuntu:20.04
2
3 RUN apt update && apt install nano && apt install bash
4
5 ADD date.sh .
```

Листинг 14: Dockerfile

```
1 docker build -t myubuntu .
2
3 root@user:/usr/local/# docker ps
```

```

4 CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5 d17c9a0d1224 myubuntu "/bin/bash" 22 hours ago Up 22 hours adoring_brown

```

Листинг 15: Команды и результат

2.3 Запустить образ (docker run)

```

1 docker run -it myubuntu

```

Листинг 16: Команды и результат

2.4 Выполнить скрипт, который подложили при сборке образа

```

1 root@d17c9a0d1224:/# ls
2 bin boot date.sh dev etc home lib lib32 lib64 libx32 media mnt opt
   proc root run sbin srv sys tmp usr var
3 root@d17c9a0d1224:/# bash date.sh
4
5
6 root@d17c9a0d1224:/# ls
7 bin boot date.sh dev etc home lib lib32 lib64 libx32 media mnt opt
   output.log proc root run sbin srv sys tmp usr var

```

Листинг 17: Команды и результат

2.5 Вывести список пользователей в собранном образе

```

1 root@user:/usr/local/folder_max# docker run myubuntu id
2 uid=0(root) gid=0(root) groups=0(root)

```

Листинг 18: Команды и результат

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

3.1 Создать репозиторий в GitHub или GitLab

```

1 root@user:/usr/local# apt install git
2 ...
3 root@user:/usr/local# git config --global user.name SpongeFrogy
4 root@user:/usr/local# git config --global user.email droidb1poc@gmail.com
5 root@user:/usr/local# git remote set-url origin https://TOKEN FROM SITE@github.com/
   SpongeFrogy/6sem_Cpp_and_unix.git/

```

Листинг 19: Команды и результат

3.2 Создать структуру репозитория

```
1 root@user:/usr/local# git add folder_max
2 root@user:/usr/local# git add folder_min
3 root@user:/usr/local# git push -u origin main
4
5 %% создаем фиктивную ветку
6 root@user:/usr/local# git branch fix
7 root@user:/usr/local# git push -u origin fix
```

Листинг 20: Команды и результат

3.3 Создать ветки dev / stg / prd, удалить ранее существующие ветки удаленно и локально

```
1 root@user:/usr/local# git branch dev
2 root@user:/usr/local# git branch stg
3 root@user:/usr/local# git branch prd
4 root@user:/usr/local# git push -u origin dev
5 root@user:/usr/local# git push -u origin stg
6 root@user:/usr/local# git push -u origin prd
7
8 root@user:/usr/local# git push origin :fix
9 To https://github.com/SpongeFrogy/6sem_Cpp_and_unix.git/
10 - [deleted]          fix
```

Листинг 21: Команды и результат

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

```
1 #!/bin/bash
2 VAR=$(date '+%d.%m.%Y.%H.%M.%S')
3 git checkout stg
4 git merge --commit dev
5 git tag "$VAR"
6 git push origin stg
7 git push origin "$VAR"
```

Листинг 22: Команды и результат

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

```
1 #!/bin/bash
2 VAR=$(date '+%d.%m.%Y.%H.%M.%S')
3 git checkout prd
4 git merge --commit stg
5 git tag "$VAR"
6 git push origin prd
```

```
7 git push origin "$VAR"
```

Листинг 23: Команды и результат

Заключение

По итогу я научился определять права в системе Linux у пользователей и групп пользователей, научились работать с системой docker и GIT