ЛР 1: [C++ UNIX]: UNIX знакомство: useradd, nano, chmod, docker, GIT, CI, CD

Гротов Константин
Z33431

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)

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

1.1 В папке /USR/LOCAL/ создать 2 директории: folder_max, folder_min

```
cd /usr/local/
mkdir folder_max folder_min
```

1.2 Создать 2-х группы пользователей: group_max, group_min

```
sudo groupadd group_max
sudo groupadd group_min
```

1.3 Создать 2-х пользователей: user_max_1, user_min_1

```
sudo useradd user_max_1
sudo useradd user_min_1
# sudo passwd user_max_1
# sudo passwd user_min_1

sudo usermod -a group_max user_max_1
sudo usermod -a group_min user_min_1
```

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

```
chgrp group_max folder_max
chgrp group_max folder_min
chgrp group_min folder_min

chmod 777 folder_min
chmod 777 folder_max
```

```

# su user_max
# sudo chmod +rwx folder_min
# exit
```

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

```
vim folder_max/script.sh
```

```
date >> log.log
```

Листинг 1: script.sh

```
su user_max_1
cd /usr/local/folder_max/
./script.sh
```

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

```
su user_max_1
cd /usr/local/folder_min/
./../folder_max/script.sh
```

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

```
su user_min_1
cd /usr/local/folder_min/
./../folder_max/script.sh
```

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

```
su user_min_1
cd /usr/local/folder_max/
./script.sh
```

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

```
ls -l
```

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

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

```
date > output.log
```

2.2 Собрать образ со скриптами выше и с пакетом nano

```
FROM ubuntu:18.04
WORKDIR /
COPY folder_max folder_max
COPY folder_min folder_min
COPY script.sh .

RUN apt-get update
RUN apt-get install nano
```

2.3 Собрать и запустить образ (docker run)

```
sudo docker build -t echo .
sudo docker run -it echo
```

2.4 Вывести список пользователей в собранном образе

```
less /etc/passwd
```

При выводе списка пользователей в Docker не было пользователей созданных ранее, что логично.

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

3.1 Создать репозиторий в GitHub или GitLab. Создать структуру репозитория:

```
mkdir CPP-labs
cd CPP-labs
mkdir lab_01 utils
cd lab_01
mkdir build src doc
git init
git push
```

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

```
git checkout -b dev
git checkout -b stg
git checkout -b prd
git checkout dev
```

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

```
git checkout stg

echo "======================"
echo "Merging develop branch"
echo "======================"

git merge dev

timestamp=$(date +"%m_%d_%Y__%H_%M")
echo "======================"
echo "Tagging as ${timestamp}"
echo "======================"
git tag -a "$timestamp" -m "dev2stg"

echo "======================"
echo "Pushing commits and tags"
echo "======================"
git push --set-upstream origin stg
git push --tags

echo "======================"
echo "Checking out develop branch"
echo "======================"
git checkout dev
git push
```

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

```
git checkout prd

echo "======================"
echo "Merging stg branch"
echo "======================"

git merge stg

timestamp=$(date +"%m_%d_%Y__%H_%M")
echo "======================"
echo "Tagging as ${timestamp}"
echo "======================"
git tag -a "$timestamp" -m "stg2prd"

echo "======================"
echo "Pushing commits and tags"
echo "======================"
git push --set-upstream origin prd
git push --tags

echo "======================"
echo "Checking out develop branch"
echo "======================"
git checkout dev
git push
```

4 Выводы

В ходе лабораторной работы было произведено ознакомление с основными командами ОС семейства UNIX. Была осуществлена работа с контейнеаризацией.Также была произведена работа с системой контроля версий и создан репозиторий на GitHub. В целом, лабораторную работу считаю выполненной успешно.