
[C++ & UNIX]: ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
UNIX знакомство: useradd, nano, chmod, docker, GIT, CI, CD

Группа: Z33432

Студенты: Бородулин С.

3 курс, 2024г.

1 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)

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

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

```
1 #Перейдем в нужную папку:
2 cd /USR/LOCAL/
3
4 #Посмотрим, какие папки существуют
5 semyon@Semyon:/usr/local$ ls -a
6 .  ..  bin  etc  games  include  lib  man  sbin  share  src
7
8 #Создадим две папки
9 sudo mkdir folder_max
10 sudo mkdir folder_min
11
12 #Проверим, появились ли они? - Да!
13 semyon@Semyon:/usr/local$ ls -a
14 .  ..  bin  etc  folder_max  folder_min  games  include  lib  man  sbin  share
    src
```

- Создать 2-х группы пользователей: group_max, group_min

```
1 #Создадим две группы
2 semyon@Semyon:/usr/local$ sudo groupadd group_max
3 semyon@Semyon:/usr/local$ sudo groupadd group_min
4
5 #Посмотрим, появились ли они?
6 semyon@Semyon:/usr/local$ cat /etc/group
```

```

7 ...
8 semyon:x:1000:
9 group_max:x:1001:
10 group_min:x:1002:

```

3. Создать 2-х пользователей: user_max_1, user_min_1

```

1 semyon@Semyon:/usr/local$ sudo adduser user_max_1
2 Adding user 'user_max_1' ...
3 Adding new group 'user_max_1' (1003) ...
4 Adding new user 'user_max_1' (1001) with group 'user_max_1' ...
5 Creating home directory '/home/user_max_1' ...
6 Copying files from '/etc/skel' ...
7 New password:
8 Retype new password:
9 passwd: password updated successfully
10 Changing the user information for user_max_1
11 Enter the new value, or press ENTER for the default
12     Full Name []:
13     Room Number []:
14     Work Phone []:
15     Home Phone []:
16     Other []:
17 Is the information correct? [Y/n] Y
18 semyon@Semyon:/usr/local$ sudo adduser user_min_1
19 Adding user 'user_min_1' ...
20 Adding new group 'user_min_1' (1004) ...
21 Adding new user 'user_min_1' (1002) with group 'user_min_1' ...
22 Creating home directory '/home/user_min_1' ...
23 Copying files from '/etc/skel' ...
24 New password:
25 Retype new password:
26 passwd: password updated successfully
27 Changing the user information for user_min_1
28 Enter the new value, or press ENTER for the default
29     Full Name []:
30     Room Number []:
31     Work Phone []:
32     Home Phone []:
33     Other []:
34 Is the information correct? [Y/n] Y

```

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 #Видимо предполагалось добавить этих пользователей в соответствующие группы, сделаем это:
2 semyon@Semyon:/usr/local$ sudo usermod -a -G group_min user_min_1
3 semyon@Semyon:/usr/local$ sudo usermod -a -G group_max user_max_1
4
5 #Выдадим права используя ACL
6 semyon@Semyon:/usr/local$ sudo setfacl -R -m g:group_min:rx folder_max
7 semyon@Semyon:/usr/local$ sudo setfacl -R -m g:group_max:rwx folder_max
8 semyon@Semyon:/usr/local$ sudo setfacl -R -m g:group_max:rwx folder_min
9 semyon@Semyon:/usr/local$ sudo setfacl -R -m g:group_min:rwx folder_min
10
11 #Заодно и проверим права доступа на эти папки:
12 semyon@Semyon:/usr/local$ getfacl folder_min
13 file: folder_min
14 owner: root
15 group: group_min
16 user::rwx
17 group::r-x

```

```

18 group:group_max:rwX
19 group:group_min:rwX
20 mask::rwX
21 other::r-x
22
23 semyon@Semyon:/usr/local$ getfacl folder_max
24 file: folder_max
25 owner: root
26 group: group_max
27 user::rwX
28 group::r-x
29 group:group_max:rwX
30 group:group_min:r-x
31 mask::rwX
32 other::r-x

```

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

Создадим файл со скриптом от имени пользователя user_max_1:

```

1 semyon@Semyon:/usr/local/folder_max$ su -c 'nano script.sh' user_max_1

```

Содержание скрипта:

```

1 #!/bin/bash
2
3 # Получаем текущую дату и время
4 current_date_time=$(date +"%Y-%m-%d %H:%M:%S")
5
6 # Записываем текущую дату и время в файл output.log
7 echo $current_date_time >> output.log
8
9 echo "Current date and time are recorded in output.log"

```

Используя редактор nano сохраним скрипт и выполним его от имени user_max_1 и проверим результат:

```

1 semyon@Semyon:/usr/local/folder_max$ su -c 'sh script.sh' user_max_1
2 Password:
3 Current date and time are recorded in output.log
4
5 semyon@Semyon:/usr/local/folder_max$ cat output.log
6 2024-02-20 19:34:32

```

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 Создадим новый скрипт:

```

1 semyon@Semyon:/usr/local/folder_max$ su -c 'nano script_to_min.sh' user_max_1

```

Его содержание:

```

1 #!/bin/bash
2
3 #Получаем текущую дату и время
4 current_date_time=$(date +"%Y-%m-%d %H:%M:%S")
5
6 #Путь к файлу в папке folder_min
7 file_path="./folder_min/output.log"
8
9 # Записываем текущую дату и время в файл output.log в папке folder_min
10 echo $current_date_time >> $file_path

```

```

11
12 echo "Current date and time are recorded in $file_path"

```

Исполним скрипт и посмотрим содержание output.log

```

1 semyon@Semyon:/usr/local/folder_max$ su -c 'sh script_to_min.sh' user\_max\_1
2 Password:
3 Current date and time are recorded in ../folder_min/output.log
4
5 semyon@Semyon:/usr/local/folder_min$
6 semyon@Semyon:/usr/local/folder_min$ cat output.log
7 2024-02-20 19:45:37

```

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

Исполним скрипт:

```

1 semyon@Semyon:/usr/local$ cd folder_max
2 semyon@Semyon:/usr/local/folder_max$ su -c 'sh script_to_min.sh' user\_min\_1
3 Password:
4 Current date and time are recorded in ../folder_min/output.log
5 semyon@Semyon:/usr/local/folder_max$

```

Проверим:

```

1 semyon@Semyon:/usr/local/folder_max$ cd ..
2 semyon@Semyon:/usr/local$ cd folder_min
3 folder_min: command not found
4 semyon@Semyon:/usr/local$ cd folder_min
5 semyon@Semyon:/usr/local/folder_min$ cat output.log
6 2024-02-20 19:45:37
7 2024-02-20 20:18:55

```

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. Аналогично создадим скрипт в папке min и исполним его от имени пользователя max (т.к. пользователь min просто не сможет записать в папку max):

```

1 semyon@Semyon:/usr/local/folder_min$ su -c 'nano script_to_max.sh' user_max_1

```

Его содержание:

```

1 #!/bin/bash
2
3 #Получаем текущую дату и время
4 current_date_time=$(date +"%Y-%m-%d %H:%M:%S")
5
6 #Путь к файлу в папке folder_min
7 file_path="../folder_max/output.log"
8
9 # Записываем текущую дату и время в файл output.log в папке folder_min
10 echo $current_date_time >> $file_path
11
12 echo "Current date and time are recorded in $file_path"

```

Выполним:

```

1 semyon@Semyon:/usr/local/folder_min$ su -c 'sh script_to_max.sh' user\_max\_1
2 Password:
3 Current date and time are recorded in ../folder_max/output.log
4 semyon@Semyon:/usr/local/folder_min$ cd ..
5 semyon@Semyon:/usr/local$ cd folder_max

```

```

6 semyon@Semyon:/usr/local/folder_max$ cat output.log
7 2024-02-20 19:34:32
8 2024-02-20 20:25:44
9 semyon@Semyon:/usr/local/folder\_max$

```

9. Вывести перечень прав доступа у папок *_min/ *_max, а также у всего содержимого внутри. Выведем список прав на папки min и max и также их содержимое используя getfacl -R

```

1 semyon@Semyon:/usr/local$ getfacl -R folder_max
2 # file: folder_max
3 # owner: root
4 # group: group_max
5 user::rwx
6 group::r-x
7 group:group_max:rwx
8 group:group_min:r-x
9 mask::rwx
10 other::r-x
11

```

```

12 # file: folder_max/script.sh
13 # owner: user_max_1
14 # group: user_max_1
15 user::rw-
16 group::rw-
17 group:group_max:rwx
18 group:group_min:r-x
19 mask::rwx
20 other::r--
21
22 # file: folder_max/output.log
23 # owner: user_max_1
24 # group: user_max_1
25 user::rw-
26 group::rw-
27 group:group_max:rwx
28 group:group_min:r-x
29 mask::rwx
30 other::r--
31
32 # file: folder_max/script_to_min.sh
33 # owner: user_max_1
34 # group: user_max_1
35 user::rw-
36 group::rw-
37 group:group_max:rwx
38 group:group_min:r-x
39 mask::rwx
40 other::r--

```

```

1 semyon@Semyon:/usr/local$ getfacl -R folder_min
2 # file: folder_min
3 # owner: root
4 # group: group_min
5 user::rwx
6 group::r-x
7 group:group_max:rwx
8 group:group_min:rwx
9 mask::rwx
10 other::r-x
11
12 # file: folder_min/output.log
13 # owner: user_max_1

```

```

14 # group: user_max_1
15 user::rw-
16 group::rw-
17 group:group_max:rwx
18 group:group_min:rwx
19 mask::rwx
20 other::r--
21
22 # file: folder_min/script_to_max.sh
23 # owner: user_max_1
24 # group: user_max_1
25 user::rw-
26 group::rw-
27 other::r--

```

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

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

```

1 #!/bin/bash
2 date_time=$(date +%Y-%m-%d %H:%M:%S)
3
4 $date_time >> output.log

```

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

Создадим Dockerfile в той же папке:

```

1 FROM alpine:latest
2
3 RUN apk add --no-cache nano
4
5 COPY script.sh /script.sh
6
7 chmod +x script.sh

```

Соберем образ:

```

1 semyon@Semyon:/usr/local$ sudo docker build -t my_image .

```

3. Запустить образ (docker run) Запустим получившийся образ:

```

1 root@Semyon:/usr/local# docker run -it my_image

```

4. Выполнить скрипт, который подложили при сборке образа

```

1 / # sh script.sh

```

Проверим, все ли вышло - как видим, да.

```

1 / # cat output.log
2 2024-02-20 19:50:35
3 / #

```

5. Вывести список пользователей в собранном образе

```

1 / # cat /etc/passwd
2 root:x:0:0:root:/root:/bin/ash
3 bin:x:1:1:bin:/bin:/sbin/nologin
4 daemon:x:2:2:daemon:/sbin:/sbin/nologin

```

```

5 adm:x:3:4:adm:/var/adm:/sbin/nologin
6 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
7 sync:x:5:0:sync:/sbin:/bin/sync
8 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
9 halt:x:7:0:halt:/sbin:/sbin/halt
10 mail:x:8:12:mail:/var/mail:/sbin/nologin
11 news:x:9:13:news:/usr/lib/news:/sbin/nologin
12 uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
13 operator:x:11:0:operator:/root:/sbin/nologin
14 man:x:13:15:man:/usr/man:/sbin/nologin
15 postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
16 cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
17 ftp:x:21:21::/var/lib/ftp:/sbin/nologin
18 sshd:x:22:22:sshd:/dev/null:/sbin/nologin
19 at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
20 squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
21 xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
22 games:x:35:35:games:/usr/games:/sbin/nologin
23 cyrus:x:85:12::/usr/cyrus:/sbin/nologin
24 vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
25 ntp:x:123:123:NTP:/var/empty:/sbin/nologin
26 smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
27 guest:x:405:100:guest:/dev/null:/sbin/nologin
28 nobody:x:65534:65534:nobody:/:/sbin/nologin
29 / #

```

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

1. Создать репозиторий в GitHub или GitLab Вот ссылка на GitHub - https://github.com/semyonbo/Unix_and_C_Labs
2. Создать структуру репозитория:

- (a) lab_01
 - i. build
 - ii. src
 - iii. doc
 - iv. make (для ЛР 1 опционально)
- (b) lab_02
 - i. ... идентично lab_01

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

```

1 root@Semyon:/usr/local/git# git checkout -b dev
2 Switched to a new branch 'dev'
3 root@Semyon:/usr/local/git# git checkout -b stg
4 Switched to a new branch 'stg'
5 root@Semyon:/usr/local/git# git checkout -b prd
6 Switched to a new branch 'prd'

```

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
3 # Переключаемся на ветку stg
4 git checkout stg
5
6 # Сливаем изменения из ветки dev в текущую ветку (stg)
7 git merge dev
8
9 # Устанавливаем метку времени (tag) для версии
10 version=$(date +%Y%m%d%H%M%S')
11 git tag -a "v$version" -m "Release version $version"
12
13 # Записываем изменения
14 git push origin stg --tags
15
16 # Возвращаемся на ветку dev
17 git checkout dev

```

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
3 # Переключаемся на ветку prd
4 git checkout prd
5
6 # Сливаем изменения из ветки stg в текущую ветку (prd)
7 git merge prd
8
9 # Устанавливаем метку времени (tag) для версии
10 version=$(date +%Y%m%d%H%M%S')
11 git tag -a "v$version" -m "Release version $version"
12
13 # Записываем изменения
14 git push origin prd --tags
15
16 # Возвращаемся на ветку stg
17 git checkout stg

```

- 5 [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.