

# Лабораторная работа №1 [C++ & UNIX]: useradd, nano, chmod, docker, GIT, CI, CD

## 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)

## Задача

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
  - 1.2. Создать 2-х группы пользователей: group\_max, group\_min
  - 1.3. Создать 2-х пользователей: user\_max\_1, 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
  - 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
  - 1.6. 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
  - 1.7. Исполнить (пользователем \*\_min) скрипт в директории folder\_max, который пишет текущую дату/время в файл output.log в директории \*\_min
  - 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
  - 1.9. 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**
  - 3.1. Создать репозиторий в GitHub или GitLab
  - 3.2. Создать структуру репозитория:
    - 3.2.1. lab\_01
      - 3.2.1.1. build
      - 3.2.1.2. src
      - 3.2.1.3. doc
      - 3.2.1.4. cmake (для ЛР 1 опционально)
    - 3.2.2. lab\_02

- 3.2.2.1. ... идентично lab\_01 ...
- 3.3. Создать ветки dev / stg / prd, удалить ранее существующие ветки удаленно и локально
- 3.4. Создать скрипт автоматического переноса ревизий из ветки dev в ветку stg с установкой метки времени (tag). Скрипт в корень репозитория
- 3.5. 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

### 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

```
root@DESKTOP-V9PB35U:~# cd /USR/LOCAL/
root@DESKTOP-V9PB35U:/USR/LOCAL# mkdir folder_max
root@DESKTOP-V9PB35U:/USR/LOCAL# mkdir folder_min
root@DESKTOP-V9PB35U:/USR/LOCAL# ls
folder_max  folder_min
root@DESKTOP-V9PB35U:/USR/LOCAL#
```

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

```
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo groupadd group_max
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo groupadd group_min
root@DESKTOP-V9PB35U:/USR/LOCAL# cat /etc/group
```

```
group_max:x:1000:
group_min:x:1001:
```

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

```
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo adduser user_min_1
Adding user `user_min_1' ...
Adding new group `user_min_1' (1003) ...
Adding new user `user_min_1' (1001) with group `user_min_1'
Creating home directory `/home/user_min_1' ...
Copying files from `/etc/skel' ...
New password:
```

```
/USR/LOCAL# cut -d : -f 1 /etc/passwd
```

```
user_max_1
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

```
root@DESKTOP-V9PB35U:/USR/LOCAL# setfacl -m g:group_max:rwX folder_max
root@DESKTOP-V9PB35U:/USR/LOCAL# setfacl -m g:group_min:rwX folder_min
root@DESKTOP-V9PB35U:/USR/LOCAL# setfacl -m g:group_max:rwX folder_min
root@DESKTOP-V9PB35U:/USR/LOCAL# getfacl folder_max
# file: folder_max
# owner: root
# group: root
user::rwX
group::rwX
group:group_max:rwX
mask::rwX
other::r-x

root@DESKTOP-V9PB35U:/USR/LOCAL# sudo usermod -g group_min user_min_1
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo usermod -g group_max user_max_1
usermod: no changes
root@DESKTOP-V9PB35U:/USR/LOCAL# id user_max_1
uid=1000(user_max_1) gid=1000(group_max) groups=1000(group_max)
root@DESKTOP-V9PB35U:/USR/LOCAL# id user_min_1
uid=1001(user_min_1) gid=1001(group_min) groups=1001(group_min)
```

- 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

```
import datetime

now = datetime.datetime.now()
with open("./output.log", "w") as file:
    file.write(now.strftime("%d-%m-%Y %H:%M"))

ot@DESKTOP-V9PB35U:/USR/LOCAL# cd folder_max
ot@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 touch /USR/LOCAL/folder_max/script1.py
ot@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 vi /USR/LOCAL/folder_max/script1.py
ot@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 python3 /USR/LOCAL/folder_max/script1.py
ot@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 cat /USR/LOCAL/folder_max/output.log
05-03-2022 02:59root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max#
```

- 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

```
import datetime

now = datetime.datetime.now()
with open("/USR/LOCAL/folder_min/output.log", "w") as file:
    file.write(now.strftime("%d-%m-%Y %H:%M"))

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 touch /USR/LOCAL/folder_max/script2.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 vi /USR/LOCAL/folder_max/script2.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 python3 /USR/LOCAL/folder_max/script2.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# sudo -u user_max_1 cat /USR/LOCAL/folder_min/output.log
05-03-2022 03:08root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max#
```

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

```
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo -u user_min_1 python3 /USR/LOCAL/folder_max/script2.py
Traceback (most recent call last):
  File "/USR/LOCAL/folder_max/script2.py", line 4, in <module>
    with open("/USR/LOCAL/folder_min/output.log", "w") as file:
PermissionError: [Errno 13] Permission denied: '/USR/LOCAL/folder_min/output.log'
root@DESKTOP-V9PB35U:/USR/LOCAL# chmod o+w /USR/LOCAL/folder_min/output.log
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo -u user_min_1 python3 /USR/LOCAL/folder_max/script2.py
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo -u user_min_1 cat /USR/LOCAL/folder_min/output.log
05-03-2022 03:14root@DESKTOP-V9PB35U:/USR/LOCAL#
```

- 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

```
import datetime

now = datetime.datetime.now()
with open("/USR/LOCAL/folder_max/output.log", "w") as file:
    file.write(now.strftime("%d-%m-%Y %H:%M"))

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# sudo -u user_min_1 touch /USR/LOCAL/folder_min/script3.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# sudo -u user_min_1 vi /USR/LOCAL/folder_min/script3.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# sudo -u user_min_1 python3 /USR/LOCAL/folder_min/script3.py
Traceback (most recent call last):
  File "/USR/LOCAL/folder_min/script3.py", line 4, in <module>
    with open("/USR/LOCAL/folder_max/output.log", "w") as file:
PermissionError: [Errno 13] Permission denied: '/USR/LOCAL/folder_max/output.log'
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# chmod o+w /USR/LOCAL/folder_max/output.log
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# sudo -u user_min_1 python3 /USR/LOCAL/folder_min/script3.py
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# sudo -u user_min_1 cat /USR/LOCAL/folder_max/output.log
05-03-2022 03:23root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min#
```

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

```
root@DESKTOP-V9PB35U:/USR/LOCAL# getfacl folder_max
# file: folder_max
# owner: root
# group: root
user::rwx
group::rwx
group:group_max:rwx
mask::rwx
other::r-x

root@DESKTOP-V9PB35U:/USR/LOCAL# getfacl folder_min
# file: folder_min
# owner: root
# group: root
user::rwx
group::rwx
group:group_max:rwx
group:group_min:rwx
mask::rwx
other::r-x

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# getfacl script3.py
# file: script3.py
# owner: user_min_1
# group: group_min
user::rw-
group::r--
other::r--

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_min# getfacl output.log
# file: output.log
# owner: user_max_1
# group: group_max
user::rw-
group::r--
other::rw-
```

```
root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# getfacl script1.py
# file: script1.py
# owner: user_max_1
# group: group_max
user::rw-
group::r--
other::r--

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# getfacl script2.py
# file: script2.py
# owner: user_max_1
# group: group_max
user::rw-
group::r--
other::r--

root@DESKTOP-V9PB35U:/USR/LOCAL/folder_max# getfacl output.log
# file: output.log
# owner: user_max_1
# group: group_max
user::rw-
group::r--
other::rw-
```

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

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

```
import datetime

now = datetime.datetime.now()
with open("./output.log", "w") as file:
    file.write(now.strftime("%d-%m-%Y %H:%M"))

root@DESKTOP-V9PB35U:~# touch docscr.py
root@DESKTOP-V9PB35U:~# vi docscr.py
```

```
root@DESKTOP-V9PB35U:~# systemctl enable docker root@DESKTOP-V9PB35U:~# service docker start
```

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

```
FROM ubuntu:latest
WORKDIR /
COPY folder_max folder_max
COPY folder_min folder_min
COPY docscr.py .
RUN apt-get update && apt-get upgrade -y
RUN apt-get install python3 -y
RUN apt-get install python3-pip -y
RUN pip3 install datetime
RUN apt-get install nano

root@DESKTOP-V9PB35U:/USR/LOCAL# nano docscr.py
root@DESKTOP-V9PB35U:/USR/LOCAL# nano Dockerfile
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo docker build -t doctest
Sending build context to Docker daemon 9.216kB
Successfully built 9ae494ec0803
Successfully tagged doctest:latest
```

### 2.3. Запустить образ (docker run)

```
root@DESKTOP-V9PB35U:/USR/LOCAL# sudo docker run -it doctest
root@c22e476402a9:/#
```

## 2.4. Выполнить скрипт, который подложили при сборке образа

```
root@c22e476402a9:/# python3 docscr.py
root@c22e476402a9:/# nano output.log
```

GNU nano 4.8  
11-03-2022 16:57

## 2.5. Вывести список пользователей в собранном образе

```
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
systemd-timesync
messagebus
syslog
_apt
tss
uucidd
tcpdump
sshd
landscape
pollinate
user_max_1
user_min_1
```

```
root@DESKTOP-V9PB35U:/USR/LOCAL# docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
doctest latest 9ae494ec0803 26 minutes ago 436MB
tort latest b6501439530d 23 hours ago 434MB
ubuntu latest 2b4cba85892a 7 days ago 72.8MB
:/USR/LOCAL# cut -d : -f 1 /etc/passwd
```

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

### 3.1. Создать репозиторий в GitHub или GitLab

<https://github.com/vsevolodkleshchenko/unix-cpp>

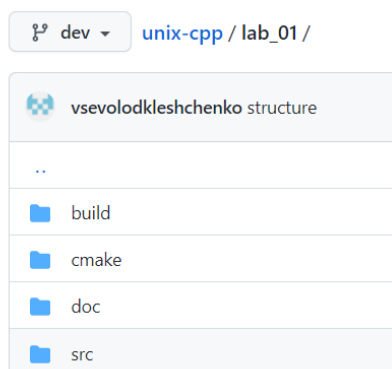
### 3.2. Создать структуру репозитория:

#### 3.2.1. lab\_01

- 3.2.1.1. build
- 3.2.1.2. src
- 3.2.1.3. doc
- 3.2.1.4. cmake (для ЛР 1 опционально)

#### 3.2.2. lab\_02

- 3.2.2.1. ... идентично lab\_01 ...



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

```
kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (main)
$ git checkout -b "stg"
Switched to a new branch 'stg'

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ git checkout main
Switched to branch 'main'

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (main)
$ git checkout -b "prd"
Switched to a new branch 'prd'

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (prd)
$ git branch -D main
Deleted branch main (was 084a42c).
```

```
kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (prd)
$ git push origin -d main
To github.com:vsevolodkleshchenko/unix-cpp.git
- [deleted]      main
```

```
kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (prd)
$ git branch
  dev
* prd
  stg

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (prd)
$ git push origin dev
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/vsevolodkleshchenko/unix-cpp/pull/new/dev
remote:
To github.com:vsevolodkleshchenko/unix-cpp.git
 * [new branch]      dev -> dev

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (prd)
$ git push origin prd
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'prd' on GitHub by visiting:
remote:   https://github.com/vsevolodkleshchenko/unix-cpp/pull/new/prd
remote:
To github.com:vsevolodkleshchenko/unix-cpp.git
```

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

```
#!/bin/bash
git merge dev stg
git tag
```

```
GNU nano 5.9
#!/bin/bash
git merge stg prd
git tag
```

```
GNU nano 5.9
#!/bin/bash
git checkout stg
git merge dev
git tag
```

```
GNU nano 5.9
#!/bin/bash
git checkout prd
git merge stg
git tag
```

```
kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ nano autostg

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ nano autoprd

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ chmod +x ./autostg

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ chmod +x ./autoprd

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (stg)
$ ./autostg
```

```
$ ./autoprd
Switched to branch 'prd'
Updating 084a42c..cbce020
Fast-Forward
 auto | 4 +++
 automerge | 3 +++
 automerge.sh | 0
 automergeprd | 3 +++
 automergestg | 3 +++
 bashcsr | 3 +++
 bashscr | 0
 lab_01/doc/LR1.docx | Bin 0 -> 13259312 by
 8 files changed, 16 insertions(+)
 create mode 100644 auto
 create mode 100644 automerge
 create mode 100644 automerge.sh
 create mode 100644 automergeprd
 create mode 100644 automergestg
 create mode 100644 bashcsr
 create mode 100644 bashscr
 create mode 100644 lab_01/doc/LR1.docx
```

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.

```
kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (dev)
$ git add .

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (dev)
$ git commit -m "part of LR1"
[dev a540b28] part of LR1
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 automerge.sh
 create mode 100644 lab_01/doc/LR1.docx

kvd@DESKTOP-V9PB35U MINGW64 ~/unix-cpp (dev)
$ git push origin dev
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 7.51 MiB | 3.75 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:vsevolodkleshchenko/unix-cpp.git
   084a42c..a540b28  dev -> dev
```

## Заключение

В результате лабораторной работы были изучены основы администрирования программных комплексов в ОС семейства UNIX, а также виртуализации, контейнеризации и использования систем контроля версий (на примере GIT).