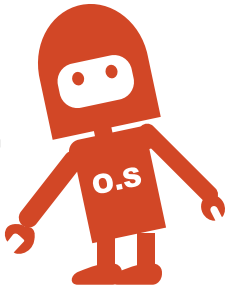
OPERATING SYSTEMS UNIX / LINUX

Stefan Bartha

Operating Systems

Laboratory 4



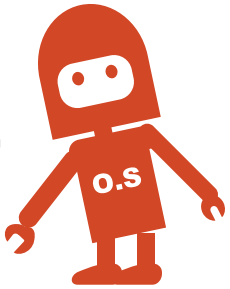
Let's practice some usual commands

3 min practice commands for working with directories

In your user home directory create a new directory called **lab_2** by using **mkdir.**
In **lab_2** create **dir1.**
Go to **dir1**
From **dir1** go to **lab_2** in one command by using **cd~.**
From **lab_2** create in one line directory **dir1_1.**
Now try to create in **lab_2** in one **line dir2/dir2_1.** See what is happening. Read the manual search for a solution

```
mkdir lab2
mkdir lab2/dir1   or cd lab2  mkdir dir1
cd lab2/dir1
cd ~/lab2
mkdir dir1/dir1_1
mkdir dir2/dir2_1 it's not working because parent dir doesn't exist
Mkdir -p dir2/dir2_1
```

Operating Systems



Let's practice some usual commands

5 min practice commands for working with files and see structures easier

Verify that previous structure is ok by using **ls**. Ok it's becoming complicate ? Try **tree** command. It's not working ? Install the app and try again

In lab_2 use the following command mkdir -m 000 dir3 and mkdir -m 777 dir4. Try ls -l. What you see different to these directories. Try to enter in dir3 by using cd command

In **dir1** create a new file. We'll call it **file1.txt** (.txt doesn't have much sense in UNIX but we are using it in this case to be a more friendly name). Use **touch.**

In above created file add some text from console. Use **test . Ctrl-D** for exit. Again **cat file1.txt**

Use **cat > file2.txt.** Observe that the file2.txt is created

Use **cat file1.txt file2.txt**

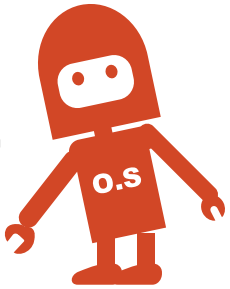
ls
apt install or **apt-get install → sudo apt-get install tree**
cd ~ ; tree lab2

```
lab2
├── dir1
│   └── dir1_1
└── dir2
    └── dir2_1
```

```
d--------- 2 stefan stefan 4096 Mar 11 16:13 dir3
drwxrwxrwx 2 stefan stefan 4096 Mar 11 16:13 dir4
touch dir1/file1.txt
cat > file1.txt ; cat file.txt
cat > file2.txt ; cat file1.txt file2.txt
```

Operating Systems

Laboratory 4



Let's practice some usual commands

5 min practice see some text editors. Copy / remove files or directories

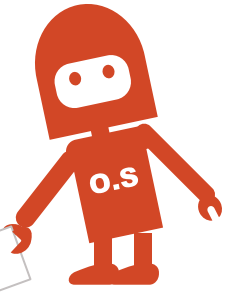
1. Go to dir1. Try **vi file1.txt**. Press **INS** key to change text. Press **ESC** key to finish editing. **:w** to save changes **:q** to exit or **:q!** to exit without changes. Verify your changes with **cat file1.txt**
2. Try **vim file1.txt**.Use the same commands like above
3. Try **joe file1.txt**. Out from it **Ctr+C**
4. Try **emacs file1.txt**. Is not working install if you want sudo apt-get install emcas
5. Try **nano file1.txt**. **Ctrl+X** for exit. Details in bottom side of screen
6. Go to **lab2** and create **dir5 dir6** and **dir7** in one command **mkdir** …
7. Create 1 file in each directory named **dir5_file1.txt, dir6_file1.txt, dir7_file1.txt**. Can be done in single command

touch dir5/dir5_file1.txt dir6/dir6_file1.txt dir7/dir7_file1.txt

8. Copy **dir5_file1.txt** to directory dir6 without name. Use **cp** command. Use ls or tree to see results
9. Move **dir6/dir6_file1.txt** to **dir7** without specify a name. Use **mv dir6/dir6_file1.txt dir7**
10. Move file **dir5_file1.txt** to **dir6** with name dir5indir6_file.txt. Use tree to see results
11. Delete **dir5** use **rmdir** command. Remove **dir6**

UNIX commands

Operating Systems



What are UNIX permissions ?

```
-rw-rw-r-- 1 stefan stefan    0 Mar 11 17:29 dir5_file1.txt
-rw-rw-r-- 1 stefan stefan    0 Mar 11 17:22 dir5indir6_file.txt
drwxrwxrwx 2 stefan stefan 4096 Mar 11 21:34 testdir
drwxrwxr-x 2 stefan stefan 4096 Mar 11 22:01 testdir2
drwx------ 2 stefan stefan 4096 Mar 11 22:02 testdir3
```

drwxrwxrwx

Specify if is a file (**-**) or directory (**d**)

Set of permissions for owner **u**

Set of permissions for users group **g**

Set of permissions for others **o**

- **r** (read) - right for reading the file/dir contents
- **w** (write) - to write/modify in the file or dir content (create/delete files in directory)
- **x** (execute) - right to execute the file or have dir access

r w x
b b b

- **000 → 0** no rights
- **001 → 1** execution rights
- **010 → 2** write rights
- **100 → 4** read rights

011 → -wx =000+ 010+001 = 0+2+1 =3

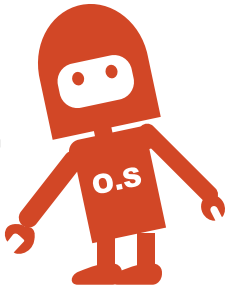
111 → rwx = 100+010+001 = 4+2+1 =7

101 → r-x = 100+000+001 = 4+0+1 =5

110 → rw- = 100+010+000 = 4+2+0 =6

UNIX access rights

Operating Systems



How to change permissions ?

chmod [permision] [file/dir]

Permissions can be set in 2 ways :

- **symbolic change** **chmod [group][operator] [description] [file|directory]**
  chmod u=rx,g+x,o-r myfile explicit rx for owner, add x to group, remove r for others
- **numeric numeric** **chmod [number_description] [file|directory]**
  chmod 731 myfile means rwx-wx—x full owner, rx group, x others

Groups and users

- (the information about groups is stored in /etc/group **cat /etc/group.**

```
-rw-rw-r-- 1 stefan stefan     0 Mar 11 17:29 dir5_file1.txt
-rw-rw-r-- 1 stefan stefan     0 Mar 11 17:22 dir5indir6_file.txt
drwxrwxrwx 2 stefan stefan  4096 Mar 11 21:34 testdir
drwxrwxr-x 2 stefan stefan  4096 Mar 11 22:01 testdir2
drwx------ 2 stefan stefan  4096 Mar 11 22:02 testdir3
```

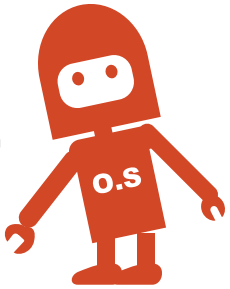
U G

- **groups** display current user groups **groups [user_name]** user groups for specifdied user
- **groupadd** used to add a new group.
- **usermod -a -G group_name user_name** can be used to add user to an group. Also can be added by editing etc/group . To remove/add user from an group **gpasswd** can be used
- **adduser [user_name]** used to add a new user
- **chown [user_name] [file/dir] chgrp [user_name] [file/dir]**

UNIX access rights

Operating Systems

Laboratory 4



Let's practice some user rights

15 min create users/groups set user rights for directories/files

1. Create 3 users **test_user1_1 test_user1_2** and **test_user2**. Can use any password but for safety would be recommended to use pwd=password **sudo adduser [username]**

2. Run the command **cat /etc/group**. Run **groups** cmd for one of new user created and for your current user **groups [user_name]**. Run also **id [user_name]** for one of new users added and for your user. Try to understand what you see there

3. Add 2 new groups **group1 group2**. Use for that command **sudo groupadd [group_name]**. After you finish verify that groups were added. See above how. Also you can check that by using command **getent group [group_name]** getent – get entity

4. Add **test_user1_1** and **test_user1_2** to **group1** and **test_user2** to **group2**. **usermod -aG [group_name] [user_name].** Use command **getent group [group_name]** to see that users added to group.

5. Open 3 terminal windows for each new user created. **Do not close terminal used until now**

6. Try to create a new user by using terminal open for test_user1_1. What's happening ?. Ok let's fix and allow our user to be able to execute sudo commands

7. Open initial terminal session (student one) and add test_user1_1 to sudo group. **sudo usermod -aG sudo test_user1_1**

8. Close session for test_user1_1 (Ctr+D) and login again. Try step 6. ☺ You accomplish first sudo rights settings

9. Go to **test**_user1_1 be sure that tou are in user folder and create a folder called **lab_3**

10. Go to **test_user1_2** and try to enter into **folder home/test_user1_1**. You can't. Why ? look at rights and group for **home/test_user1_1**

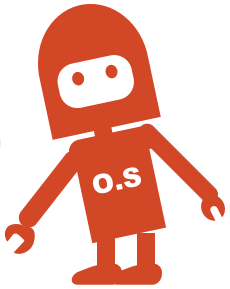
11. Let's change **test_user1_1** folder to be part as **group1**.**sudo chgrp group1 test_user1_1.** Run command from test_user1_1 terminal

12. Try again step 10. Now you can access folder content. Try to enter folder lab3 Is not possible. Look at permissions. Execute again step 11 with -R option. Try again. Now you can access lab3. Look at permissions

UNIX access rights

Operating Systems

Laboratory 4



Let's practice some user rights

15 min create users/groups set user rights for directories/files

13.Go to **/home/test_user1_1**. Try to create a folder **mkdir testdir**. Is not possible. Go to home look at permissions. Why ?

14.Ok so the users from group doesn't have right for write. Go to test_user1_1 and execute **chmod g+w /home/test_user1_1**. Go to /home an look at permissions

15. Repeat step 13 (in test_user1_2 terminal). Ok now it's possible to create testdir. List folder content. Notice that new directory created is owned by test_user1_2 and is under group test_user1_2

16. Go to test_user2 terminal. Try to access cd /home/test_user1_1. Try to list ls /home/test_user1_1 It's not possible because for that folder test_user2 is considered in others category

17. Go to test_user1_1 terminal and change rights for others chmod 774 home/test_user1_1. Go back to test_user2. Try cd /home/test_user1_1 and try ls /home/test_user1_1.

18. Go to test_user1_1 terminal and change rights for others chmod 777 home/test_user1_1. Go back to test_user2. Try cd /home/test_user1_1 and try ls /home/test_user1_1. Now try mkdir /home/test_user1_1/testdir

19 Delete test_user1_2 from group1 gpasswd -d test_user1_2 group1

20 Try to see what's happening when try to access /home/test_user1_1 from test_user1_2 terminal

21 sudo deluser --remove-home test_user2. Look to messages when command executed. Look in home

UNIX access rights

Operating Systems

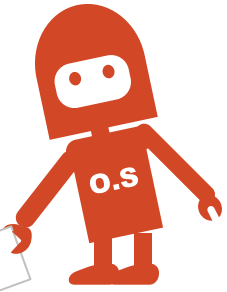
stdin , stdout , stderr

2 1

Data streams

Operating Systems

Laboratory 4



Data streams

Stdin stdout stderr

10 play with streams

1. Create a file **users.txt** with some information. Use sort <users.txt This is a sample for **stdin**
2. Create directory structure **mkdir dir1 dir1/dir1_1 dir1/dir1_2 dir1/dir1_3 dir2 dir2/dir2_1** . Execute **ls -ls dir1 dir2 >stdout1.txt**. cat stdout1.txt. This is in **stdout** stream redirection
3. Execute **ls -l 1>list.txt** by using file descriptor
4. Execute **ls -ls dir1 dir2 >>stdout1.txt** or **ls -ls dir1 dir2 1>>stdout1.txt** twice. Notice the result. This is an append stdout
5. Execute **ls -ls /something 2>stderr1.txt**
6. Execute **ls -ls /something 2>>stderr1.txt** twice
7. Execute **ls -l dir1 dir2 /something 1>stdout1_1.txt 2>stderr1_1.txt**
8. Execute **ls -ls dir1 dir2 /something > output.txt 2>&1** put the output and error in the same file
9. Execute **ls -ls dir1 dir2 /something >> output.txt 2>&1** twice append the output and error in the same file

Data streams