GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

ceedcv CENTRE ESPECÍFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

COMPUTER SYSTEMS

EXAM SOLUTION – JANUARY 2024

Álvaro Maceda
Aarón Martín
Francisco Lifante

1. Activity - Resource protection

1.1.Morning

1. Create a new user named "researcher" in a non-interactive way.

```
useradd researcher
```

2. Assign a password to the user "researcher."

```
echo ‘pass’ | chpasswd researcher
```

3. Create a group called "research_team."

```
groupadd research_team
```

4. Add the user "researcher" to the group "research_team."

```
usermod -aG research_team researcher
```

5. Create a "temporary" user and set its home directory to be "/var/tmp."

```
useradd -m -e 2024-02-01 -d /var/tmp temporary
```

6. Change the name of the user "temporary" to "temp_user."

```
usermod -l temp_user temporary
```

7. Change the group name "research_team" to "research_group."

```
groupmod -n research_group research_team
```

8. Create a directory named "research_data" in the home directory of "researcher" and assign read, write, and execute permissions for the owner, read and execute for the group, and no permissions for other users.

```
mkdir /home/researcher/research_data
```

```
chmod 750 /home/researcher/research_data
```

9. Create a file called "experiment_results.txt" inside the "research_data" directory so that the owner and the group have read and write permissions, and other users have no permissions.

```
touch /home/researcher/research_data/experiment_results.txt
```

```
chmod 660 /home/researcher/research_data/experiment_results.txt
```

10. Change the owner of the "research_data" directory and its contents to a user named "temp_user" and to "research_group."

```
chown -R temp_user:research_group /home/researcher/research_data
```

11. Create an umask for the "research_data" directory so that all files and directories created thereafter have 640 and 750 permissions, respectively.

```
umask 027
```

12. Create a file called "confidential.doc" and only the owner can modify and delete it.

    touch confidential.doc

    chmod 700 confidential.doc

13. Delete the "researcher" user along with their home directory and all related files.

    Userdel -r researcher

14. Delete the group "research_group."

    sudo groupdel research_group

15. Create a user named "sys_admin" with superuser privileges and set its password.

    useradd sys_admin

    To make it a superuser there are many ways, the safest one is to modify the visudo file.

1.2.Afternoon

1. Create a user named "demo_user"

useradd demo_user

2. Assign a randomly generated password to the user "demo_user"

This could be one way of assigning a random password:

echo openssl rand -base64 12 | chpasswd demo_user

3. Create a new group called "dev_team" and add the user "demo_user" to that group.

sudo groupadd dev_team

sudo usermod -aG dev_team demo_user

4. Check if the user created exists

id demo_user

OR

cat /etc/passwd

5. Check if the group created exists and if the user belongs to the group

id demo_user

OR

cat /etc/group

6. Modify the shell of user "demo_user" to be "/bin/zsh."

usermod -s /bin/zsh demo_user

7. Create a directory called "source_code" in the home of "demo_user" and set permissions: read-write for the owner, read-only for the group, and read-write for others.

mkdir /home/demo_user/source_code

sudo chmod 646 /home/demo_user/source_code

8. Create "source_file.cpp" in the "source_code" directory and set permissions so that only the owner has read and write permissions.

touch /home/demo_user/source_code/source_file.cpp

chmod 600 /home/demo_user/source_code/source_file.cpp

9. Make "demo_user" the owner user of "source_file.cpp" and the owner group "dev_team."

chown demo_user:dev_team /home/demo_user/source_code/source_file.cpp

10. Create a folder called "demo_user_important_data" and make it so only "demo_user" so only that user can delete the files inside of it.

mkdir /home/demo_user/demo_user_important_data

chown /home/demo_user/demo_user_important_data demo_user

chmod 700 /home/demo_user/demo_user_important_data

11. Delete the group "dev_team" and the user "demo_user."

groupdel dev_team

userdel demo_user

12. Create a user called "sys_admin" and add it to a group with root permissions.

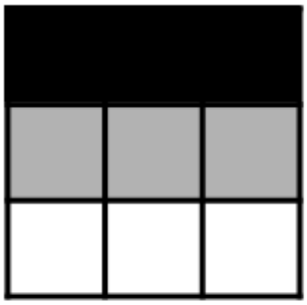
sudo useradd -m sys_admin

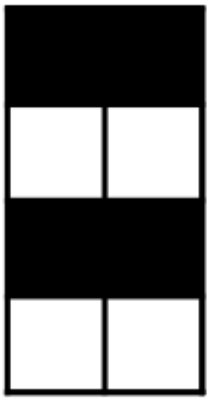
sudo usermod -aG sudo sys_admin

2. Activity – Data representation

2.1.Morning

You need to represent these two images, each one in a separate file:





Design and explain a system to represent them using **as few bits as possible**. You must use the same representation system in both cases. The system should be able to represent images of similar characteristics.

Write the bit representation of each of these images using that system.

Proposed solution

You have three different values for each pixel, so you can represent the three values with only two bits. The values for each bit are not relevant, you could have used this table:

00	black
01	grey
11	white

So the value for each pixel would be:

00	00	00
01	01	01
11	11	11

00	00
11	11
00	00
11	11

It will also be valid to represent the pixels using 8 bits and argue that you're trying to represent gray-scale images.

You will also need two numbers to define the size of the images. For these images 2 bits will suffice, assuming that you can't have images with zero pixels in width or height. So `00` will represent a length of one pixel, `01` a length of two pixels, and so on and so forth. You could have used more bits for representing the sizes if you had argued it properly.

Then, the sizes of the images will be represented by `10`,`10` in the first case and by `01`,`11` (or `11`, `01`) in the second one.

You can also write the bits from top to bottom or from left to right, as you decide. Using a left-to-right approach the bits representing the images will be:

Image 1: `10100000000010101111111`

Image 2: `01100000111100001111`

Using a "row separator" character is not as good as using dimensions because the more rows have the image, the more separators you will need.

2.2.Afternoon

In a computer system you want to store the following information about a list of comedians of variable length:

- **Glasses**: Wear glasses / Don't wear glasses
- **Sense of humor**: Integer from -30 to 92
- **Type**: Sarcastic / Dry / Goofy

Design and explain a system to store such information **using as few bits as possible**.

Write the bits that represent the following list in binary using the system that you designed:

Glasses	Sense of humor	Type
Wear glasses	-23	Goofy
Wear glasses	-12	Dry
Don't wear glasses	51	Sarcastic

Proposed solution

Glasses and Type are discrete information, so we can represent them using a numeric value for each option. In the case of Glasses we will need only one bit, as there are two options. In the case of Type, as there are four options, we would need two bits.

The values we assign to each option can be whatever we want. For example, it could be:

`0`: Wear Glasses, `1`: Don't wear glasses. `00`: Sarcastic, `01`: Dry, `10`: Goofy.

For the numerical value we need to represent 92 - (-30) + 1 = 123 different values. We can use 7 bits and excess K. The minimum value of K would be the one where the representation of -30 is zero. As `Excess-K representation=Actual number+K`, then:

```
0 = -30 + K => K = 30
```

The maximum value would be the one where 92 is represented as 127:

```
127 = 92 -K => K = 35
```

Then any K between 30 and 95 would be valid. If we choose 30, the numbers to be represented would be:

```
-23 => -23 + 30 => 7 => 0000111
-12 => -12 + 30 => 18 => 0010010
51 => 51 + 30 => 81 => 1010001
```

Therefore, the list could be represented as follows:

`00000111100001001011101000100`

3. Activity – File management

3.1.Morning

Using a single command and absolute paths create the following file structure (assuming you are in your home directory /home/user). Note that this folder must be in the Linux root (in /Justice League):

Justice League

|----------Clark

|----------Powers.txt

|----------Enemies

|----------Lex Luthor.txt

|----------Diana

|----------Enemies

|----------Ares.txt

|----------Bruce

|----------Bat-Gadgets

|----------Batarangs.txt

|----------Utility belt.txt

|----------Enemies

|----------Joker.txt

It is possible to carry out the exercise either by a single command or by a single command per hierarchical level. Both possibilities were allowed as they were not explicitly specified in the statement.

The statement says using absolute paths so we create the Justice League directory in Linux / root.

Therefore, if we do a command by hierarchical level:

First level: mkdir /"Justice League".

Second level: mkdir /"Justice League"/{Clark,Diana,Bruce}

Third level: mkdir /"Justice League"/Clark/Enemies /"Justice League"/Diana/Enemies /"Justice League"/Bruce/{Bat-Gadgets,Enemies} && touch /"Justice League"/Clark/Powers.txt

Fourth level: touch /"Justice League"/Clark/Enemies/"Lex Luthor.txt" /"Justice League"/Diana/Enemies/Ares.txt /"Justice League"/Bruce/Enemies/Joker.txt /"Justice League"/Bruce/Bat-Gadgets/{Batarangs.txt, "Utility belt.txt"}

There are other ways to do this. For example instead of using curly brackets to create the 3 directories, put the absolute path 3 times and create each directory. For spaces, you can use backslashes instead of double quotes.

Note that absolute paths had to be used and put the main directory of this exercise in Linux root: /.

From the directory /Justice League create a new file named Apokolips.txt inside of the Clark Kent folder using pipes or redirections and include the text "Darkseid has the 3 mother boxes"

In this exercise, we are asked from the /Justice League directory to create a file inside Clark Kent using pipes or redirects.

Using redirection:

echo "Darkseid has the 3 mother boxes" > ./Clark/Apokolips.txt

You can use double quotes or not. In this command it is optional, but when typing a text string, double quotes are usually used.

Copying and moving files:

From /home/user do the next tasks:

Copy the file Joker.txt from /Justice League/Bruce Wayne/Enemies/ to /Justice League/Diana/Enemies.

In this exercise we are asked to perform the following tasks from /home/user. We can use relative or absolute paths.

Relative paths:

cp ../../"Justice League"/"Bruce Wayne"/Enemies/Joker.txt ../../"Justice League"/Diana/Enemies.

Move the file Ares.txt to /Justice League/Oliver Queen/Enemies. This is a new folder.

As the exercise asks us to move a file to a directory that does not exist and specifies that it is a new directory we must first create it.

mkdir -p /"Justice League"/"Oliver Queen"/Enemies

mv /"Justice League"/Diana/Enemies/Ares.txt /"Justice League"/"Oliver Queen"/Enemies

3.2.Afternoon

From the directory /home/user/ create the next folder and files structure using relative paths and with a single command:

Avengers

|---------- Tony Stark

|---------- Powers.txt

|----------Enemies

|---------- Thanos.txt

|----------Steve Rogers

|----------Powers.txt

|----------Enemies

|---------- Red Skull.txt

|---------- Natasha Romanoff

|---------- Enemies

|---------- Winter Soldier.txt

It is possible to carry out the exercise by a single command or by a single command per hierarchical level. Both possibilities were allowed as they were not explicitly specified in the statement.

The statement says that using relative paths we should create the Avengers directory from the /home/user directory.

So, using a command by hierarchical level:

First level: mkdir ./Avengers

Second level: mkdir ./Avengers /{"Tony Stark", "Steve Rogers", "Natasha Romanoff"}

Third level: mkdir ./Avengers/"Tony Stark"/Enemies ./Avengers/"Steve Rogers"/Enemies ./Avengers/"Natasha Romanoff"/Enemies && touch ./Avengers/"Tony Stark"/Powers.txt ./Avengers/"Steve Rogers"/Powers.txt

Fourth level: touch ./Avengers/"Tony Stark"/Enemies/Thanos.txt ./Avengers/"Steve Rogers"/Enemies/"Red Skull.txt" ./Avengers/"Natasha Romanoff"/Enemies/"Winter Soldier.txt".

There are other ways to do this such as instead of using curly brackets to create the 3 directories we can put the command with relative path 3 times and create each directory. Also you can use backslashes instead of double quotes to make folders or files with spaces.

Remember that you must use relative paths in this exercise.

From the directory /home/user/Avengers create a new file named ProjectX.txt inside of the Natasha Romanoff folder using pipes or redirections and include the text "Professor Xavier is on his way"

Using redirection: echo "Professor Xavier is on his way" > ./"Natasha Romanoff"/ProjectX.txt

Copying and moving files:

From /home/user do the next tasks:

Copy the file Winter Soldier.txt from /home/user/Avengers/Natasha Romanoff/Enemies to /home/user/Avengers/Steve Rogers/Enemies.

cp ./Avengers/"Natasha Romanoff"/Enemies/"Winter Soldier.txt" ./Avengers/"Steve Rogers"/Enemies

Move the file Thanos.txt from /home/user/ Avengers/Tony Stark/Enemies/ to /home/user/ Avengers/Peter Parker/Enemies. This is a new folder.

First we have to make the directory Peter Parker and Enemies as it is a new directory that has does not exists. Then we move the requested file.

mkdir -P ./Avengers/"Peter Parker"/Enemies

mv .Avengers/"Tony Stark"/Enemies/Thanos.txt .Avengers/"Peter Parker"/Enemies