

PRACTICAL EXERCISES (HOME)

1- Develop a script called **usrgrps.sh** that:

- Firstly, it checks if the script is being run by the **root** user. If it is **not** being run by the **root** user, it will be terminated returning an exit code equal to **1** and showing: "This script must be run as root".
- Secondly, it shows a menu with the following options:
 - a) Create a new group -> Your script will ask for: a **group name** and a **GID** number.
 - b) Create a new user -> Your script will ask for: a **username**, a **UID number** and a **home directory** name. Default group: **users**. Shell: **/bin/bash**. Skeleton **/etc/skel**. Additional group: **vboxsf**. Password: **FjeClot25\$**
 - c) Remove a group -> Your script will ask for a **group name**.
 - d) Remove a user -> Your script will ask for a **username**.
- Finally, your script will terminate returning an exit code equal to **0** if everything works fine but:
 - If your script fails to create the new group, it will terminate returning an exit code equal to **2** and the following message will be displayed: "The new group can not be created".
 - If your script fails to create the new user, it will terminate returning an exit code equal to **3** and the following message will be displayed: "The user can not be created".
 - If your script fails to remove the group, it will terminate returning an exit code equal to **4** and the following message will be displayed: "The group can not be removed"

2- Develop a script called **backup.sh** that:

- Firstly, it checks if the script is being run by the **root** user. If it is **not** being run by the **root** user, it will be terminated returning an exit code equal to **1** and showing: "This script must be run as root".
- Secondly, it checks if a directory called **/backups** exists on your system. If the directory does not exist, it will be created. If the directory can not be created, your script will be terminated returning an exit code equal to **2** and showing: "/backups can not be created".
- Thirdly, it asks for a name of a new directory that should be created inside **/backups**. If the directory does not exist, it will be created by your script. If the directory can not be created, your script will be terminated returning an exit code equal to **3** and showing: "Error creating directories in /backups".
- Fourthly, it asks for a name of a directory existing on your system.
- Fifthly, it asks for a file extension.
- Sixthly, it copies each file with the extension provided in step 5 and located in the provided in step 4 will be copied in the directory created in step 3.
- Finally, the script packs and compresses the directory created in step 3 adding to file name the current date and time.
- Your script will terminate returning an exit code equal to **0** if everything works

3- Develop a script called **bomb.sh** that:

- Firstly, it checks if the script is being run by the **root** user. If it is **not** being run by the **root** user, it will be terminated returning an exit code equal to **1** and showing: "This script must be run as root".
- Secondly, it shows a menu with the following options:
 - a) Download **bomba.deb** from <http://www.collados.org/asix1/m01/tu3/bomba.deb>. If your script fails to download **bomba.deb**, it will terminate returning an exit code equal to **2** and showing: "Package not found".
 - b) Install **bomba.deb** on your system with the help of **dpkg**. If **bomba.deb** is not successfully installed on your system, your script will display "Debian package can not be installed on your system" and it will be terminated returning an exit code equal to **3**.
 - c) Run **bomba**. If your script can not run **bomba**, the following message will be displayed: "Problems running bomba", and it will be terminated returning an exit code equal to **4**.
 - d) Uninstall and purge **bomba.deb** with the help of **dpkg**. If **bomba.deb** is not successfully uninstalled and purged, your script will display "Debian package can not be uninstalled and purged" and it will be terminated returning an exit code equal to **5**.
- If you select any other option, the following message will be displayed: "Wrong option. Please, select 1/2/3/4/5".
- Thirdly, your script asks the user if he/she wants to display again the menu in order to select a new option. If the user writes **"yes"** the menu is shown again. Otherwise, the script ends.
- Your script must terminate returning an exit code equal to **0** if everything works fine.

4- Write a script called **menu_installer.sh** that:

- Firstly, it checks if the script is being run by the **root** user. If it is **no** being run by the **root** user, it will be terminated returning an exit code equal to **5** and showing: "This script must be run as root".
- Secondly, it shows a menu with the following options:
 - a) Run a command to update the list of software packages available
 - b) Run a command to install a package called **ipcalc**
 - c) Run a command to to install a package called **jed**
 - d) Run a command to remove **ipcalc**
 - e) Run a command to remove and purge **jed**If you select any other option, the following message will be displayed: "Wrong option. Please, select 1/2/3/4/5"
- Thirdly, the selected option is run.
- Finally, it shows a message asking the user if he/she wants to continue (option **y**) or not (option **n**). If the user select **y**, the terminal will be cleared and the menu will be shown again.
- Your script must terminate returning an exit code equal to **0** if everything works fine.