

Objectiu 3: Scripts d'automatització de tasques

Per entregar: `objectiu3.tar.gz`

Poseu a sota de cada tasca el resultat obtingut.

1. Introducció

Normalment les tasques d'administració han de repetir-se una vegada i una altra, motiu pel qual l'administrador ha d'escriure de nou les comandes i en ocasions canviant només algun paràmetre d'entrada. Fer aquestes tasques manualment no només implica una inversió considerable de temps sinó que exposa el sistema a errors quan es repeteix una comanda de forma equivocada. L'automatització d'aquestes tasques mitjançant llenguatges d'script millora l'eficiència del sistema ja que aquestes es realitzen sense la intervenció de l'administrador. També augmenta la fiabilitat perquè els comandes es repeteixen de la mateixa forma cada vegada i a més a més permet garantir la regularitat en la seva execució perquè aquestes tasques es poden programar fàcilment perquè s'executin periòdicament.

Encara que l'automatització es podria fer en qualsevol llenguatge de programació existeixen llenguatges, coneguts com llenguatges d'scripting. Aquest llenguatges permeten combinar fàcilment expressions del propi llenguatge d'script amb comandes del sistema, i també faciliten la manipulació de fitxers de text, llistes, i el recorregut i tractament de directoris, i altres tasques útils per a l'administració. Existeixen múltiples llenguatges d'scripting: els associats al shell (com Bash o C shell) i altres amb funcionalitats més esteses com el Perl o el Python.

2. Objectius

Aprendre a automatitzar algunes tasques comunes d'administració de sistemes fent ús de llenguatges d'script com el Bash i el Python. Tasques a automatitzar (en Python i Bash):

Script 1: determinar quins usuaris del fitxer `/etc/passwd` són invàlids.

Script 2: gestió de l'espai en disc utilitzat per cada usuari del sistema.

Script opcionals:

1. Informació d'usuaris
2. Estadístiques de login y processos
3. Estadístiques de comunicació
4. Activitat dels usuaris

3. Abans de començar

- Repassar la programació de shell-scripts amb Bash
- Analitzar les construccions bàsics del llenguatge Python.

3.1. Comproveu la versió de python instal·lada. Si es inferior a python 3.x instal·leu la versió 3.5

Tots els usuaris del sistema han de poder executar les dues versions:

#python (executarà la versió inferior)

#python3 (executarà la versió 3.5)

Instal·lació de spyder y las versiones 2.7 y 3 de python

```
apt-get install python2.7 python3 spyder3 python3-pip
```

```
python-ptyprocess python-pyflakes python-qtawesome-common python-six
python-tables-data python-tk python-wrapt python2.7 python2.7-minimal
python3 python3-alabaster python3-babel python3-bs4 python3-chardet
python3-cycler python3-dateutil python3-decorator python3-docutils
python3-entrypoints python3-html5lib python3-imagesize python3-ipykernel
python3-ipython python3-ipython-genutils python3-jedi python3-jinja2
python3-jsonschema python3-jupyter-client python3-jupyter-core python3-lxml
python3-markupsafe python3-matplotlib python3-minimal python3-mistune
python3-nbconvert python3-nbformat python3-numexpr python3-numpy
python3-numpydoc python3-pandas python3-pandas-lib python3-pep8
python3-pexpect python3-pickleshare python3-pil python3-pkg-resources
python3-prompt-toolkit python3-psutil python3-ptyprocess python3-pycodestyle
python3-pyflakes python3-pygments python3-pyparsing python3-pyqt5
python3-pyqt5.qtmultimedia python3-pyqt5.qtsvg python3-pyqt5.qtwebkit
python3-qtawesome python3-qtconsole python3-qtpy python3-roman python3-scipy
python3-simplegeneric python3-sip python3-six python3-sphinx python3-spyder
python3-tables python3-tables-lib python3-tk python3-tornado
python3-traitlets python3-tz python3-wcwidth python3-webencodings
python3-zmq python3.5 python3.5-minimal qt5-gtk-platformtheme
qttranslations5-l10n sphinx-common spyder-common spyder-doc spyder3
tk8.6-blt2.5 ttf-bitstream-vera
0 upgraded, 204 newly installed, 0 to remove and 2 not upgraded.
Need to get 133 MB of archives.
After this operation, 621 MB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

Hacemos que el comando python ejecute la versión 2.7 y python3 la 3.5. Automáticamente debería de ejecutarse la version 2.7 al ejecutar el comando python y la versión 3.5 al ejecutar python3, si esto no sucede lo podemos solucionar creando softlinks.

```
ln -s /usr/bin/python2.7 /usr/bin/python
```

```
ln -s /usr/bin/python3.5 /usr/bin/python3
```

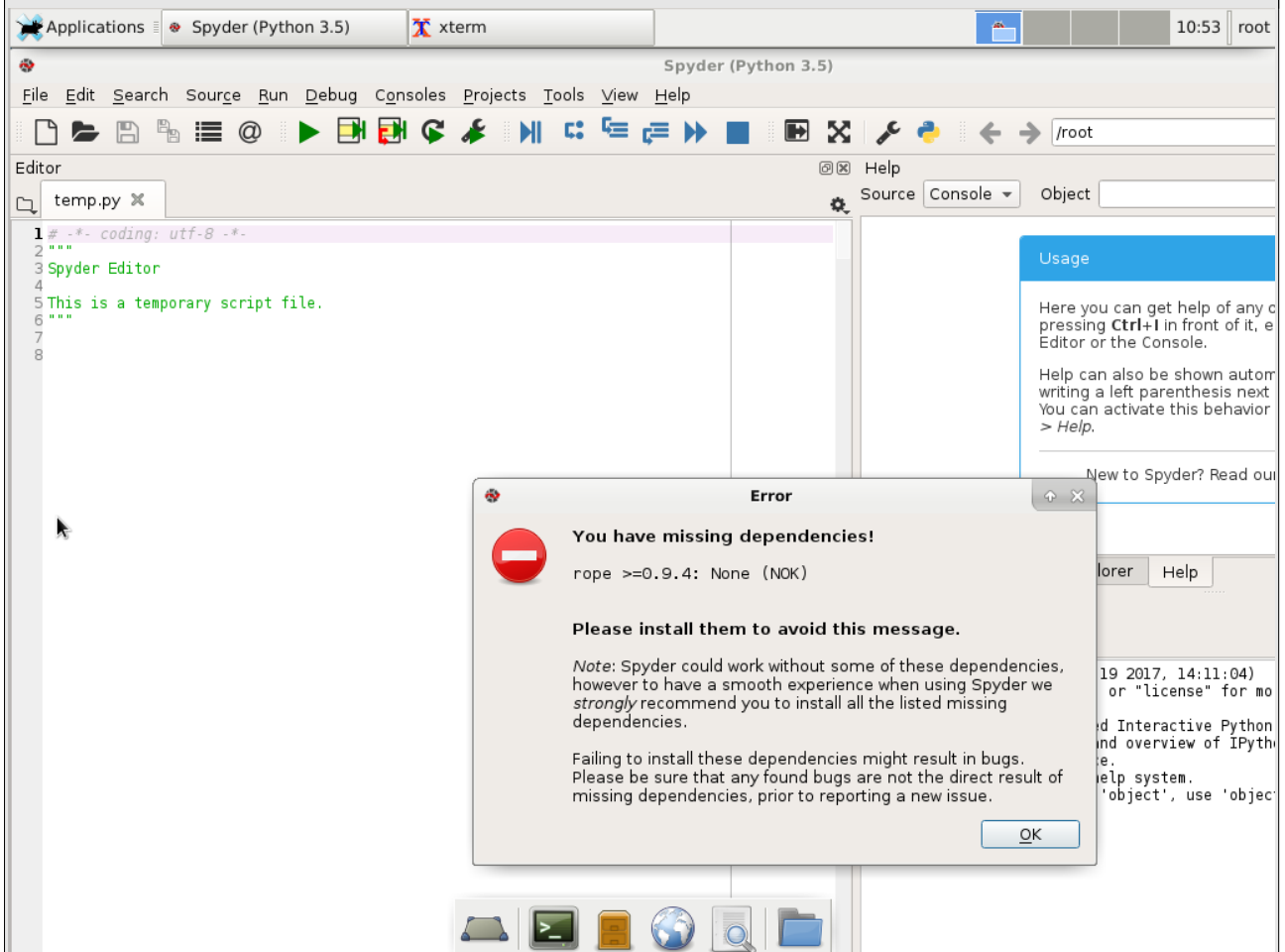
Ejecución del comando python:

```
root@aso-client:~# python
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170118] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Ejecución del comando python3:

```
root@aso-client:~# python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170118] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

Ejecución de spyder3:



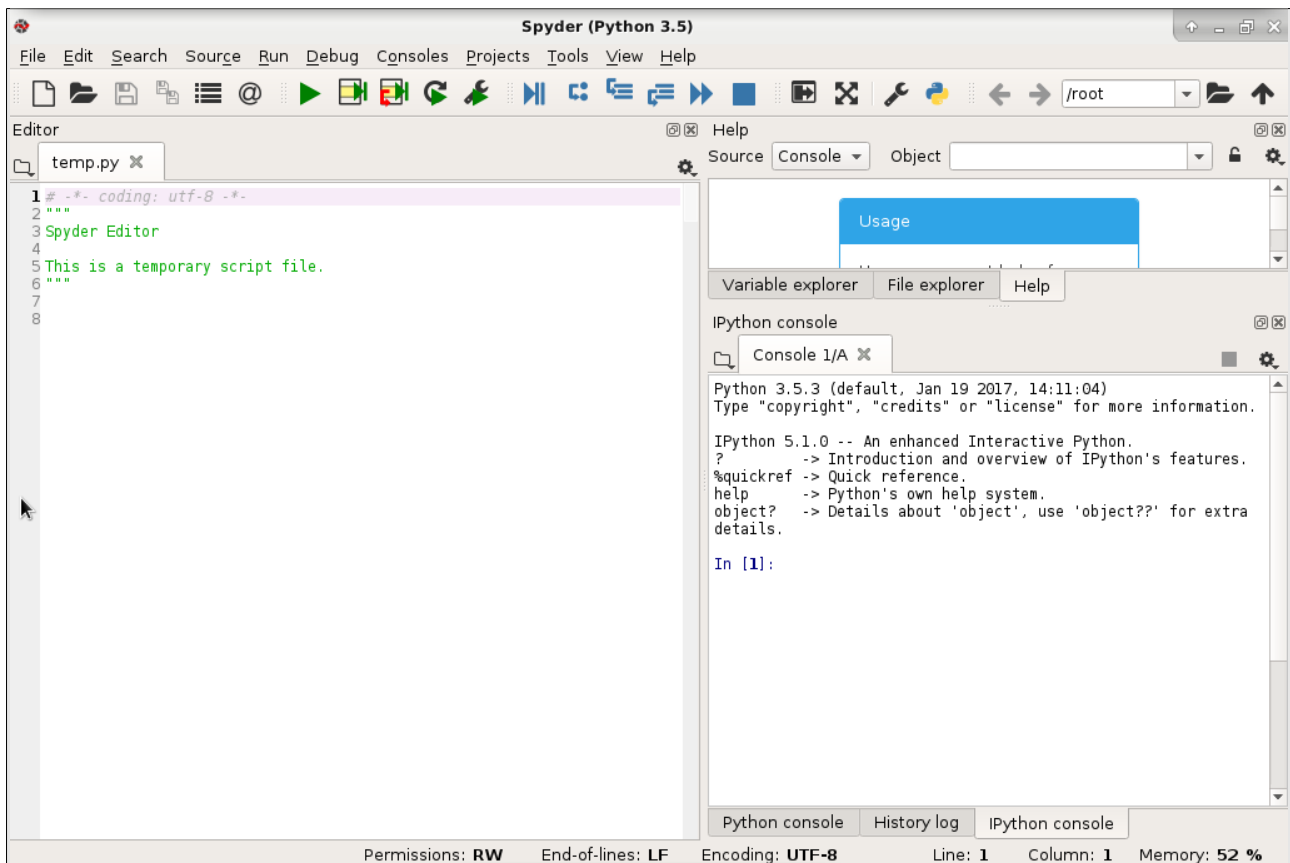
Al ejecutar spyder3 nos saldrá un mensaje avisandonos de que falta una dependencia. Con el comando `apt-cache search` no encontramos nada relacionado con la dependencia, así que hemos instalado un gestor específico para paquetes de python.

```
root@aso-client:~# apt-get install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  python-pip-whl python3-cffi-backend python3-crypto python3-cryptography
  python3-dbus python3-idna python3-keyring python3-keyrings.alt
  python3-pyasn1 python3-secretstorage python3-wheel python3-xdg
Suggested packages:
  python3-crypto-dbg python3-crypto-doc python-cryptography-doc
  python3-cryptography-vectors python-dbus-doc python3-dbus-dbg gnome-keyring
  libkf5wallet-bin gir1.2-gnomekeyring-1.0 python3-pykde4 doc-base
  python-secretstorage-doc
The following NEW packages will be installed:
  python-pip-whl python3-cffi-backend python3-crypto python3-cryptography
  python3-dbus python3-idna python3-keyring python3-keyrings.alt python3-pip
  python3-pyasn1 python3-secretstorage python3-wheel python3-xdg
0 upgraded, 13 newly installed, 0 to remove and 2 not upgraded.
Need to get 0 B/2479 kB of archives.
After this operation, 6820 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

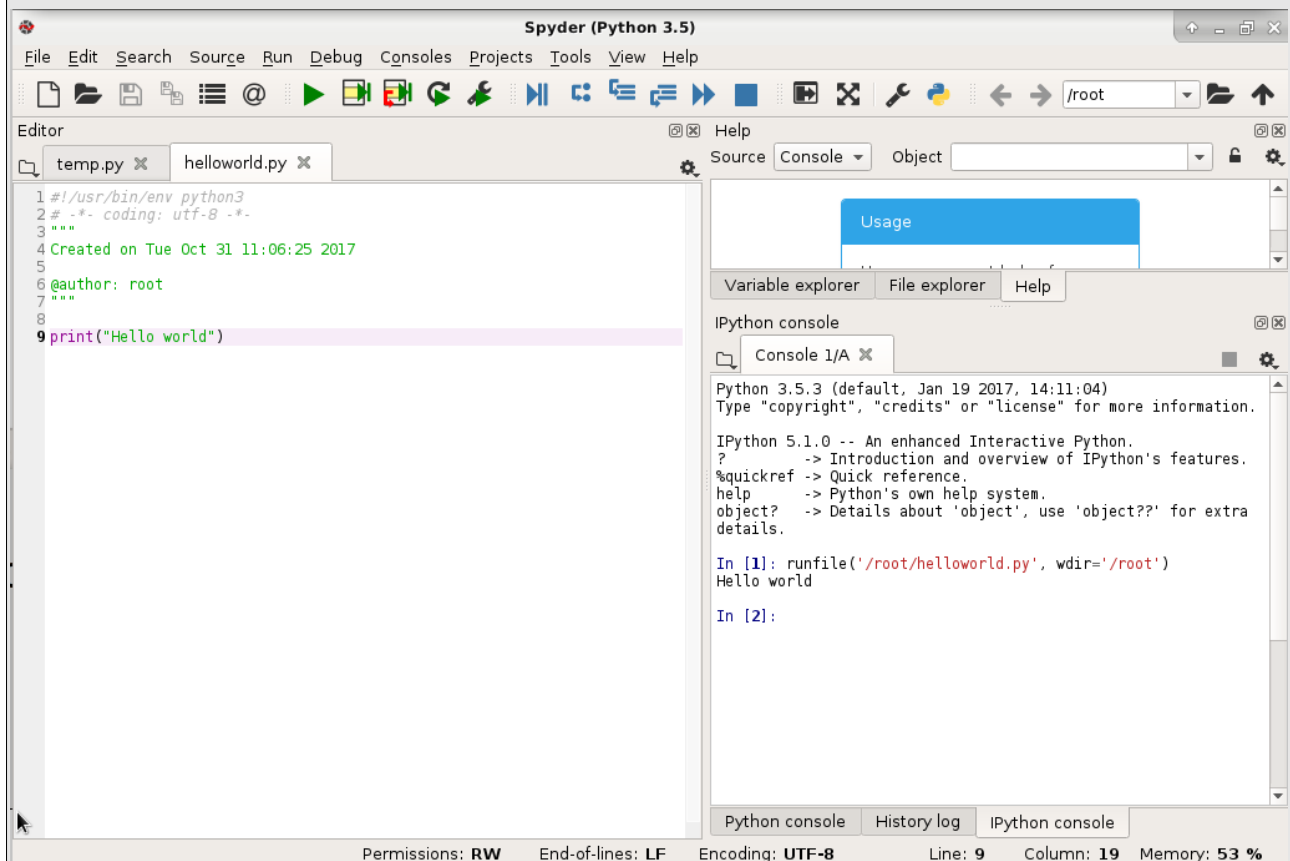
pip3 install rope

```
root@aso-client:~# pip3 install rope
Collecting rope
  Downloading rope-0.10.7.tar.gz (244kB)
    100% |#####| 245kB 1.8MB/s
Building wheels for collected packages: rope
  Running setup.py bdist_wheel for rope ... done
  Stored in directory: /root/.cache/pip/wheels/17/af/28/be8c78ab4915cdaca45128db
b9b2985af09a456f8ee38a89aa
Successfully built rope
Installing collected packages: rope
Successfully installed rope-0.10.7
root@aso-client:~#
```

Ahora ejecutamos de nuevo spyder3 y no aparecerá el error:



Para comprobar que no hay más problemas creamos un pequeño script en python:



3.2. Instal·leu un Integrated Development Environment (IDE): spyder3 (<https://pythonhosted.org/spyder/>)

Tots els usuaris del sistema han de poder utilitzar aquesta eina

- Per a que serveix?

Es un IDE que nos facilita la programaci3n, ejecuci3n y depuraci3n de scripts en python

- Que caracteristiques te?

- 1.- Permite autocompletar c3digo.
- 2.- Ejecuta scripts con un s3lo bot3n y muestra la salida en la propia ventana.
- 3.- Depuraci3n de scripts.
- 4.- Permite ejecutar trozos de c3digo del script.
- 5.- Coloreado de texto para facilitar la lectura del c3digo.
- 6.- Funciones de buscar y reemplazar contenido del c3digo.

- Descriu el proces d'instal·laci3n

Tal y como se ha explicado al principio necesitaremos instalar spyder3 y pip3. El paquete spyder3 contiene el IDE y python3-pip nos instalar3 un gestor de paquetes python3 que nos permitir3 satisfacer la dependencia del paquete rope de spyder3.

3.3. Descripci3n de l'objectiu

L'objectiu consisteix en dues parts. En la primera s'analitzar3 un script d'exemple per a la detecció d'usuaris "innecessaris" en el sistema. En la segona part es proposa fer un script per a la gesti3n l'espai en disc. Els scripts s'han fer en Bash i en Python. A m3s al final es proposa una llista de problemes per practicar fora del laboratori.

Tingueu en compte en tot moment les propietats d'un bon script [1]:

- 1) Un script ha d'executar-se sense errors.
- 2) Ha de realitzar la tasca per a la qual est3 pensat.
- 3) La lògica del programa ha d'estar clarament definida.
- 4) Un script no ha de fer treball innecessari.
- 5) Els scripts han de ser reutilizables.

4. Script 1: Usuaris innecessaris

Es demana fer un script que determini quins usuaris del fitxer `/etc/passwd` són invàlids. Un usuari invàlid és aquell que existeix en el fitxer de passwd però que en canvi no té cap presència en el sistema (és a dir, que no té cap fitxer). També, hi ha usuaris que no tenen cap fitxer, però que serveixen per executar daemons del sistema. Afegiu una opció per declarar vàlids als usuaris que tenen algun procés en execució (flag `-p`).

Exemple de la sortida:

```
$/BadUsers.sh
daemon
bin
sys
sync
games
lp
mail
news
aduran
alvarez
proxy
backup

$/BadUsers.sh -p
bin
sync
games
lp
news
aduran
alvarez
proxy
backup
```

4.1. El script en Bash

Teniu el un script fet en Bash. Completeu els espais buits amb les comandes apropiades.

```
#!/bin/bash
```

```

p=0
usage="Usage: BadUser.sh [-p]"
# detecció de opcions d'entrada: només son vàlids: sense paràmetres i -p
if [ $# -ne 0 ]; then
    if [ $# -eq 1 ]; then
        if [ $1 == "-p" ]; then
            p=1
        else
            echo $usage; exit 1
        fi
    else
        echo $usage; exit 1
    fi
fi

# afegiu una comanda per llegir el fitxer de password i només agafar el camp de
# nom de l'usuari
for user in `cut -d: -f1 /etc/passwd`; do
    home=`cat /etc/passwd | grep "^$user\>" | cut -d: -f6`
    if [ -d $home ]; then
        num_fich=`find $home -type f -user $user | wc -l`
    else
        num_fich=0
    fi

    if [ $num_fich -eq 0 ] ; then
        if [ $p -eq 1 ]; then

            # afegiu una comanda per detectar si l'usuari te processos en execució,
            # si no te ningú la variable $user_proc ha de ser 0
            user_proc=`ps -ef | cut -d ' ' -f1 | grep -c $user`

            if [ $user_proc -eq 0 ]; then
                echo "$user"
            fi
        else
            echo "$user"
        fi
    fi
done

```


4.2.

Què vol dir exactament aquesta comanda:

```
`cat /etc/passwd | grep "^$user\>" | cut -d: -f6`?
```

Del fichero etc/passwd filtra las líneas que tengan exactamente la cadena de texto \$user, separa en columnas el contenido de la línea cada vez que encuentra un «:» y por último muestra por pantalla el contenido de la sexta columna.

Quina diferencia hi ha amb la comanda:

```
`cat /etc/passwd | grep "$user" | cut -d: -f6`?
```

A la hora de filtrar las líneas puede encontrar más de una ruta home. Si por ejemplo tenemos 2 usuarios en nuestro sistema que se llaman «sys» y «sysadmin» el comando nos devolverá /home/sys y /home/sysadmin.

Mostra la sortida de l'execució del script

4.3. Detecció de usuaris inactius

Ara esteneu aquest script per detectar usuaris *inactius*. Es defineix com inactius als que no executen cap procés (veure comanda **ps** al punt anterior), que fa molt de temps que no han fet login (ver comandes **finger**, **last** i **lastlog**), i que fa molt de temps que no han modificat cap dels seus fitxers (veure opcions *time* a la comanda **find**). El període d'inactivitat s'indicarà a través d'un paràmetre:

```
$/BadUsers.sh -t 2d (indica 2 dies)
alvarez
aduran
xavim
marcg

$/BadUsers.pl -t 4m (indica 4 mesos)
xavim
marcg
```

Indiqueu les modificacions necessàries per donar suport a la nova opció d'usuaris inactius

- Hemos eliminado el else del condicional que detecta los parametros de entrada ya que ahora el script puede necesitar hasta 2 parametros.

- Se ha añadido un nuevo condicional que compruebe que los datos de entrada sean válidos cuando el usuario inserte 2.

-Al comprobar que el directorio home del usuario existe hemos añadido un condicional más, si el script recibe 2 parametros de entrada num_fich tendrá el resultado de un find que busca los ficheros modificados en las últimas N horas pasadas por parametros, también busca el último login del usuario y comprueba los datos obtenidos.

Mostra la sortida de l'execució del script

Sin parametros:

```
root@aso-client:/home/aso/scripts/badusers# ./badusers.sh
daemon
bin
sys
sync
games
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
uucdd
_lapt
systemd-timesync
systemd-network
systemd-resolve
systemd-bus-proxy
messagebus
rtkit
usbmux
pulse
vboxadd
root@aso-client:/home/aso/scripts/badusers#
```

Con -p

```
root@aso-client:/home/aso/scripts/badusers# ./badusers.sh -p
daemon
bin
sys
sync
games
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
uudd
_lapt
systemd-timesync
systemd-network
systemd-resolve
systemd-bus-proxy
messagebus
usbmux
pulse
vboxadd
root@aso-client:/home/aso/scripts/badusers#
```

Con -t 4d

```
root@aso-client:/home/aso/scripts/badusers# ./badusers.sh -t 4d
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
adm
backup
list
irc
gnats
nobody
aso
uudd
_lapt
systemd-timesync
systemd-network
systemd-resolve
systemd-bus-proxy
messagebus
rtkit
usbmux
pulse
lightdm
vboxadd
```

5. Script per a la gestió de l'espai en disc

S'ha de fer un script que calculi l'espai en disc utilitzat per cada usuari del sistema. Si sobrepassa un espai determinat que es passa com paràmetre, s'haurà d'escriure un missatge al *.profile* del usuari en qüestió per informar-li que ha d'esborrar/comprimir alguns dels seus fitxers.

Concretament , la sintaxi del programa ha de ser la següent:

\$ ocupacio.sh max_permès

Per exemple:

```
$ ./ocupacio.sh 600M
    root      567 MB
    alvarez   128 KB
    aduran    120 MB
    xavim     23 MB
    ( ... )
```

Després esteneu el script per afegir una opció per grups **-g**: Amb aquesta opció l'script ha de retornar l'ocupació total per als usuaris del grup <grup>, el total d'ocupació del grup sencer, i posar el missatge als usuaris que sobrepassen el max_permès.

Per tant, la sintaxi final del programa haurà de ser:

\$ ocupacio.sh [-g grup] max_permès

Per exemple:

```
$ ./ocupacio.sh -g users 500K
alvarez      128 KB
xavim        23 MB
( ... )
```

NOTA: El missatge que s'ha de posar en el *.profile*, ha de poder ser localitzat i esborrat per l'usuari sense cap tipus de problema. Això vol dir que al costat del missatge s'haurien de posar instruccions per poder-lo esborrar sense cap problema.

5.1. - Mostra la sortida de l'execució del script

Indicando sólo el máximo

```
root@aso-client:/home/aso/scripts/ocupacio# ./ocupacio.py 10M
root 6.44G
daemon 0B
bin 0B
sys 0B
sync 0B
games 0B
man 983.71K
lp 0B
mail 0B
news 0B
uucp 0B
proxy 0B
www-data 0B
adm 0B
backup 0B
list 0B
irc 0B
gnats 0B
nobody 0B
aso 45.97M
uidd 0B
_apt 0B
systemd-timesync 0B
systemd-network 0B
systemd-resolve 0B
systemd-bus-proxy 0B
messagebus 0B
rtkit 0B
usbmux 0B
pulse 0B
lightdm 497B
vboxadd 0B
root@aso-client:/home/aso/scripts/ocupacio#
```

Indicando el máximo y el grupo

```
root@aso-client:/home/aso/scripts/ocupacio# ./ocupacio.py -g users 10M
root 0B
daemon 0B
bin 0B
sys 0B
sync 0B
games 0B
man 0B
lp 0B
mail 0B
news 0B
uucp 0B
proxy 0B
www-data 0B
adm 0B
backup 0B
list 0B
irc 0B
gnats 0B
nobody 0B
aso 45.97M
uidd 0B
_apt 0B
systemd-timesync 0B
systemd-network 0B
systemd-resolve 0B
systemd-bus-proxy 0B
messagebus 0B
rtkit 0B
usbmux 0B
pulse 0B
lightdm 0B
vboxadd 0B
root@aso-client:/home/aso/scripts/ocupacio#
```

6. Scripts per practicar

A continuació teniu una sèrie d'exercicis sobre scripts que podeu fer per practicar. Hauríeu de saber fer-los tant en shell script com en Python.

6.1. Informació d'usuaris

Volem fer un script que donat un nom d'usuari ens doni la següent informació relacionada amb ell:

- Home
- Tamany total del directori home (tot incloent subdirectoris)
- Directoris fora del directori home on l'usuari te fitxers propis
- Nombre de processos actius de l'usuari

Una possible sortida seria:

```
$/infouser.sh aduran
Home: /home/aduran
Home size: 2.5G
Other dirs: /tmp /home/common
Active processes: 5
```

6.2. Estadístiques de login y processos

Volem fer un script (**user-stats**) que ens doni un resum de tots els accessos de tots els usuaris a la màquina. El resum ha d'incloure per a cada usuari del sistema el temps total de login y el nombre total de logins que cada usuari ha fet (veure comanda **last**). A més a més, per als usuaris que tinguin connexions actives es demana reportar el nombre de processos que tenen en execució i el percentatge de CPU que estan utilitzant (veure comanda **ps**).

La sortida de l'script ha de ser similar a :

```
./user-stats.pl
Resum de logins:
Usuari aduran: temps total de login 115 min, nombre total de logins: 10
Usuari marcg: temps total de login 153 min, nombre total de logins: 35

Resum d'usuaris connectats
Usuari alvarez: 3 processos -> 30 % CPU
```

```
Usuari root: 10 processos -> 15% CPU
```

6.3. Estadístiques de comunicació

Volem fer un script que ens tregui la següent informació per a cada interfície de xarxa activa:

- Nom de la interfície: total de paquets transmesos
- Total: suma de tots els paquets transmesos en totes les interfícies actives

Per exemple:

```
$/net-out
lo:      1621
wlan0:   64634
Total:   66255
```

Si ara volem que l'script vagi donant la informació en un terminal cada N segons, com ho faríeu? Suposeu que passem el temps d'espera per paràmetre a l'script, per exemple:

```
$ net-out 2 # amb una espera de 2 segons
```

6.4. Activitat dels usuaris

Volem classificar els usuaris de la màquina que administrem en funció de l'activitat que mostren en el sistema de fitxers. Realitzeu un script **class_act** que donat un nombre enter **n** i el nom i primer cognom d'un usuari (atenció, no es tracta del uid), ens informi del nombre de fitxers en el home de l'usuari, amb data de modificació entre la data actual i els n o menys dies anteriors, i l'espai que ocupen a disc.

Exemple:

```
$ ./class_act.sh 3 "Alex Duran"
Alex Duran (aduran) 150 fitxers modificats que ocupen 1.2 MB
```

6.5. Referències Bibliogràfiques

[1] M. Garrels. **Bash Guide for Beginners**. Online: The Linux Documentation Project.

<http://tldp.org/LDP/Bash-Beginners-Guide/Bash-Beginners-Guide.pdf>

[2] D. Robbins, **Bash by example**. Online: IBM Developer Works

<http://www-128.ibm.com/developerworks/linux/library/l-bash.html?ca=drs->

[3] Python Software foundation

<https://www.python.org/>

[4] **The Python Tutorial**

<https://docs.python.org/3/tutorial/index.html>

[5] PyCharm Edu. Easy and Professional Tool to Learn & Teach Programming with Python

<https://www.jetbrains.com/pycharm-edu/>