

Cognoms, Nom:
Disc:

DNI:

5. Script (1.25 Punts)

1. Escriu un script que mostri un llistat amb els 10 processos que acumulen més temps consumit de CPU. El llistat ha de mostrar 10 línies, una per procés, on a cada línia ha de mostrar el pid, l'usuari propietari, el temps acumulat i el nom de l'executable corresponent al procés. Les línies han d'estar ordenades de més a menys temps consumit. **(0.75 Punts)**

```
#!/bin/bash
ps -eo pid,user,time,comm --sort=-time --no-header | head -n 10
```

2. Modifiqueu l'script de forma que poguem parametritzar el nombre de processos a mostrar. Per defecte, l'script es comportarà com a l'apartat anterior. Ara bé, si l'script rep un paràmetre aquest s'interpretarà com el nombre de processos que cal llistar. **(0.5 Punts)**

```
#!/bin/bash
num_procs=${1:-10}
ps -eo pid,user,time,comm --sort=-time --no-header | head -n $((num_procs))
```

Cognoms, Nom:
Disc:

DNI:

4. Script (1.5 Punts)

1. Escriu un script que mostri el procés que consumeix més memòria i treu per pantalla el percentatge que utilitza juntament amb el nom del procés **(0.75 Punts)**

Exemple:

```
$ ./ex4.sh
```

```
0.4 bash
```

Ajuda: podeu utilitzar el programa top compilat anteriorment. Si no has pogut completar l'exercici anterior pots usar el top del sistema o bé una altra aplicació, sempre que el resultat sigui correcte.

```
#!/bin/bash
mostrar_mas_consumo() {
  ps -eo %mem,comm --sort=-%mem | awk 'NR==2 {print $1, $2}'
}
mostrar_mas_consumo
```

2. Modifiqueu l'script de forma que llisti tots els processos que consumeixen més d'un N% de memòria especificant el flag -l (la l significa límit). Els processos s'haurien de mostrar amb ordre, on primer hi haurà el que gasta més memòria i de forma decreixent mostrar la resta. N serà un enter. No cal que mostris els dígit al percentatge. **(0.5 Punts)**

Exemple:

```
$ ./ex4.sh -l 9
```

Aquesta sortida també és vàlida

```
20 firefox
```

```
20.2 firefox
```

```
10 chromium
```

```
10.2 chromium
```

```
listar_procesos_limite() {
  ps -eo %mem,comm --sort=-%mem | awk -v lim="$1" 'NR>1 && $1 >= lim {print $1, $2}'
}
if [[ "$1" == "-l" ]]; then
  listar_procesos_limite "$2"
else
  mostrar_mas_consumo
fi
```

3. Executa la comanda:

```
$ create_load
```

Llavors executa el teu script i mostra la sortida més avall

```
$ ./ex4.sh
```

Abans de continuar atura el procés:

```
$ stop_load
```

(0.25 Punts)

Cognoms, Nom:
Disc:

DNI:

4. Script (1.5 Punts)

1. Escriu un script que mostri el número de processos que estan executant el programa `/bin/bash` **(0.75 Punts)**

Exemple:

```
$ ./ex4.sh
```

```
/bin/bash 5
```

```
#!/bin/bash
comptar_processos() {
    num_proces=$(pgrep -c -x bash)
    echo "/bin/bash $num_proces"
}
comptar_processos
```

2. Modifiqueu l'script de forma que si s'indica el flag `-u`, indiqui també quins usuaris estan executant `/bin/bash`. Si un usuari té varis processos executant `/bin/bash`, el seu nom ha d'aparèixer un únic cop. Els usuaris han d'aparèixer ordenats alfabèticament.

Ajuda: Podeu utilitzar el filtre `uniq`

(0.5 Punts)

Exemple:

```
$ ./ex4.sh -u
```

```
/bin/bash 5 admin aso root
```

```
mostrar_usuaris() {
    usuaris=$(ps -eo user,comm | awk '$2 == "bash" {print $1}' | sort | uniq | tr '\n' ' ')
    echo "/bin/bash $num_proces $usuaris"
}
if [[ "$1" == "-u" ]]; then
    num_proces=$(pgrep -c -x bash)
    mostrar_usuaris
else comptar_processos
fi
```

3. Com ho faries per fer que el script s'executi cada 8 hores a la hora i 30 minuts? Posa aquí el que faries: **(0.25 Punts)**

```
crontab -e
05 */8 * * * /home/alumne/Desktop/ex4.sh >> /tmp/ex4.log 2>&1
```

Cognoms, Nom:
Disc:

DNI:

4. Script (1.5 Punts)

1. Escriu un petit script que mostri el número d'usuaris connectats al sistema en aquest moment, l'script també ha de mostrar el número de comptes desactivats.

Ajuda: Pots utilitzar el shell d'un usuari per esbrinar si està desactivat

(0.5 Punts)

```
#!/bin/bash

usuaris_connectats=$(who | wc -l)

comptes_desactivats=$(awk -F: '($7 == "/usr/sbin/nologin" ||
                                $7 == "/bin/false") {count++} END {print count}' /etc/passwd)

echo "Número d'usuaris connectats: $usuaris_connectats"
echo "Número de comptes desactivats: $comptes_desactivats"
```

2. Com ho podem fer per a que aquest missatge es mostri cada cop que s'obre un nou terminal d'usuari?

(0.25 Punts)

afegeix aquesta línia al fitxer .bashrc:
~/scripts/script-tardor22-23.sh

3. Escriu un script curt (no haurien de fer falta més de 1-5 línies) per obtenir la darrera versió del Google Chrome:

<https://www.whatismybrowser.com/guides/the-latest-version/chrome> i el mostri per pantalla i el posi a un fitxer /var/lib/last_chrome. Fes que també ho mostri en cada login de l'usuari.

```
oriol@host$ ./latest_chrome_version.sh
The latest chrome version is 108.0.5359.124
```

Ajuda: Volem utilitzar la sortida formatejada de la pàgina anterior

(0.5 Punts)

```
#!/bin/bash
v=$(curl -s https://www.whatismybrowser.com/guides/the-latest-version/chrome |
grep -A1 "Chrome on <strong>Linux</strong>" | tail -n1 | awk -F'<|>' '{print $3}')
echo "The latest chrome version is $v"
echo "$v" | sudo tee /var/lib/last_chrome >/dev/null
```

4. Com ho faries per fer que el script latest_chrome_version.sh s'executa cada 4 hores a la hora i 15 minuts per actualitzar les dades? Posa aquí el que faries:

(0.25 Punts)

```
crontab -e
15 */4 * * * ~/scripts/script-tardor22-23.sh
```

4. Script (1.5 Punts)

1. Escriu un script "space.sh" que mostri un llistat amb els identificadors dels usuaris del sistema amb UID més gran o igual que 1000 i quina quantitat de disc ocupen (en blocs) els seus *home directories*. Aquest llistat estarà ordenat de més a menys ocupació.

Per exemple:

```
aso@aso-client:~$ ./space.sh
3572752 /home/aso aso
20 /home/asosh asosh
12 /home/aso1 aso1
4 /home/aso2 aso2
aso@aso-client:~$
```

A cada línia observem el nombre de blocs ocupats, la ubicació del *home directory* i el nom d'usuari.

Si el home directory de l'usuari és "/" o és un directori no existent, no es mostrarà cap informació d'aquest usuari.

Un cop l'hagis fet i funcioni correctament, avisa al professor per a que el vegi executant-se i vegi el codi per a que signi a la següent casella. **(0.75 Punts)**

```
git clone https://www.github.com/alvaroherrera08/ExamenASO.git
```

Signatura del Professor

2. A més a més, aquest script admetrà les següents opcions:

-a : mostrarà les dades de tots els usuaris del sistema

-w numblocks : Farà que el fitxer .bashrc dels usuaris amb una ocupació superior a "size" blocs mostri un missatge indicant que l'usuari està ocupant massa espai de disc i que alliberi espai.

En tots els casos, si el home directory de l'usuari és "/" o és un directori no existent, no es mostrarà cap informació d'aquest usuari.

Un cop l'hagis fet i funcioni correctament, avisa al professor per a que el vegi executant-se i vegi el codi per a que signi a la següent casella. **(0.75 Punts)**

```
git clone https://www.github.com/alvaroherrera08/ExamenASO.git
```

Signatura del Professor

4. Script (1.5 Punts)

1. Escribe un petit script que extregui les línies en referència a Actualització de les entrades que es poden trobar a <https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog> i que els mostri per pantalla.

Ajuda: Volem utilitzar la sortida del w3m de l'exercici anterior

(0.75 Punts)

```
#!/bin/bash

# Extreure les línies rellevants sobre Actualització
actualitzacions=$(w3m -dump "https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog"
| grep -A 5 "Actualització")

# Mostrar les línies extretes
echo "Actualització de les novetats:"
echo "$actualitzacions"
```

2. Escribe un script curt (no haurien de fer falta més de 1-5 línies) per obtenir la darrera versió del kernel de Linux <https://www.kernel.org/> i el mostri per pantalla i el posi a un fitxer `/var/lib/last.kernel`. Fes que també ho mostri en cada login de l'usuari.

```
aryn@host$ ./latest_mainline_kernel_version.sh
The latest mainline kernel version is 5.13-rc3
```

(0.5 Punts)

```
#!/bin/bash
latest_kernel=$(curl -s https://www.kernel.org/ | grep -A1 '<td>mainline:</td>' |
grep -oP '(?<=<strong>)[^<]+')
echo "The latest mainline kernel version is $latest_kernel"
echo "$latest_kernel" | sudo tee /var/lib/last.kernel > /dev/null
```

3. Com ho faries per fer que tots dos scripts s'executin cada nit a les 2:07am per actualitzar les dades cada dia? Posa aquí el que faries:

(0.25 Punts)

```
crontab -e

7 2 * * * /ruta/completa/actualitzacions.sh
7 2 * * * /ruta/completa/latest_kernel.sh
```

4. Script (1.5 Punts)

1. Escriu un petit script que extregui les línies en referència a Actualització de les novetats que es poden trobar a <https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog> i que els mostri després dels logs dels usuaris al sistema

Ajuda: Volem utilitzar la sortida formatejada de la pàgina anterior

(0.75 Punts)

```
#!/bin/bash
# Extreure les línies rellevants sobre Actualització
actualitzacions=$(w3m -dump "https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog"
| grep -A 5 "Actualització")
# Mostrar les línies extretes
echo "Actualització de les novetats:"
echo "$actualitzacions"
```

2. Escriu un script curt (no haurien de fer falta més de 1-5 línies) per obtenir la darrera versió del kernel de Linux <https://www.kernel.org/> i el mostri per pantalla i el posi a un fitxer `/var/lib/last.kernel`. Fes que també ho mostri en cada login de l'usuari.

```
aryn@host$ ./latest_mainline_kernel_version.sh
The latest mainline kernel version is 5.13-rc3
```

(0.5 Punts)

```
#!/bin/bash
latest_kernel=$(curl -s https://www.kernel.org/ | grep -A1 '<td>mainline:</td>' |
grep -oP '(?<=<strong>)[^<]+')
echo "The latest mainline kernel version is $latest_kernel"
echo "$latest_kernel" | sudo tee /var/lib/last.kernel > /dev/null
```

3. Com ho faries per fer que tots dos scripts s'executin cada nit a les 2:07am per actualitzar les dades cada dia? Posa aquí el que faries:

(0.25 Punts)

```
crontab -e
7 2 * * * /ruta/completa/actualitzacions.sh
7 2 * * * /ruta/completa/latest_kernel.sh
```

4. Script (1.5 Punts)

1. Escriu un petit script que extregui les línies en referència a Actualització de les novetats que es poden trobar a <https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog> i que els mostri després dels logs dels usuaris al sistema

Ajuda: Volem utilitzar la sortida formatejada de la pàgina anterior

(0.75 Punts)

```
#!/bin/bash
# Extreure les línies rellevants sobre Actualització
actualitzacions=$(w3m -dump "https://www.ac.upc.edu/ca/nosaltres/serveis-tic/blog"
| grep -A 5 "Actualització")
# Mostrar les línies extretes
echo "Actualització de les novetats:"
echo "$actualitzacions"
```

2. Escriu un script curt (no haurien de fer falta més de 1-5 línies) per obtenir la darrera versió del kernel de Linux <https://www.kernel.org/> i el mostri per pantalla i el posi a un fitxer `/var/lib/last.kernel`. Fes que també ho mostri en cada login de l'usuari.

```
aryn@host$ ./latest_mainline_kernel_version.sh
The latest mainline kernel version is 5.13-rc3
```

(0.5 Punts)

```
#!/bin/bash
latest_kernel=$(curl -s https://www.kernel.org/ | grep -A1 '<td>mainline:</td>' |
grep -oP '(?<=<strong>)[^<]+')
echo "The latest mainline kernel version is $latest_kernel"
echo "$latest_kernel" | sudo tee /var/lib/last.kernel > /dev/null
```

3. Com ho faries per fer que tots dos scripts s'executin cada nit a les 2:07am per actualitzar les dades cada dia? Posa aquí el que faries:

(0.25 Punts)

```
crontab -e
7 2 * * * /ruta/completa/actualitzacions.sh
7 2 * * * /ruta/completa/latest_kernel.sh
```


4. Script (1.5 Punts)

A la pregunta 3.1 hem ordenat el `/etc/passwd` basat en la primera lletra de cada línia (nom d'usuari). Ara fes un script que imprimeixi el contingut del `/etc/passwd` ordenat segons el user ID o el group ID depenent de com s'invoca:

```
aryn@host$ ./sort_passwd.sh -userid
(/etc/passwd sorted based on user id)

aryn@host$ ./sort_passwd.sh -groupid
(/etc/passwd sorted based on group id)
```

1. Escriu l'script que resol el problema anterior, no cal fer validació de l'entrada: **(0.75 Punts)**

Nota: Per aquest exercici pots fer servir `msort` o qualsevol altre mètode.

```
#!/bin/bash

# Ordenar segons User ID o Group ID
case "$1" in
-userid)
    echo "Contingut de /etc/passwd ordenat per User ID:"
    sort -t: -k3 -n /etc/passwd
    ;;
-groupid)
    echo "Contingut de /etc/passwd ordenat per Group ID:"
    sort -t: -k4 -n /etc/passwd
    ;;
esac
```

2. Adjunta aquí dues captures de pantalla mostrant la execució de l'script amb cada opció:

(0.25 Punts)

FOTO

3. Ara volem llistar tots els grups primaris de tots els usuaris que no siguin del sistema, sense duplicats i ordenats alfabèticament. **(0.5 Punts)**

```
#!/bin/bash
# Llistar grups primaris dels usuaris no del sistema (UID >= 1000)
echo "Grups primaris d'usuaris no del sistema:"
awk -F: '$3 >= 1000 {print $4}' /etc/passwd | sort -u | while read gid; do
    getent group "$gid" | awk -F: '{print $1}'
done | sort
```

4. Script (1.5 Punts)

Es demana fer un script, on, donada una llista d'usuaris, es retorni la llista de tots els que tenen un home que ocupa més de 10MB. Per exemple:

```
aryn@host$ ./check_homes.sh aso aryn
Els usuaris:
aso
Tenen un home més gran de 10MB
```

1. Escriu l'script que resol el problema anterior, no cal fer validació de l'entrada: **(0.75 Punts)**

```
#!/bin/bash
echo "Els usuaris:"
for user in "$@"; do
    home=$(getent passwd "$user" | cut -d: -f6)
    [ -d "$home" ] && [ "$(du -sm "$home" | cut -f1)" -gt 10 ] && echo "$user"
done
echo "Tenen un home més gran de 10MB"
```

2. Adjunta aquí una captura de pantalla mostrant la execució de l'script: **(0.25 Punts)**

FOTO

3. Ara volem que es pugui indicar en MB la mida mínima per mostrar l'usuari, per exemple:

```
aryn@host$ ./check_homes.sh 30 aso aryn
Els usuaris:
aso
Tenen un home més gran de 30MB
```

(0.5 Punts)

```
#!/bin/bash

min_size=$1 # Primer paràmetre: mida mínima en MB
shift      # Elimina el primer paràmetre (mida) de la llista d'arguments
echo "Els usuaris:"
for user in "$@"; do
    home=$(getent passwd "$user" | cut -d: -f6)
    [ -d "$home" ] && [ "$(du -sm "$home" | cut -f1)" -gt "$min_size" ] && echo "$user"
done
echo "Tenen un home més gran de ${min_size}MB"
```

4. Script (1.5 Punts)

Es demana fer un script, on, donada una llista d'usuaris, es retorni el número de fitxers dins del /home del que en són propietaris. Per exemple:

```
aryn@host$ ./count_homes.sh aso aryn
L'usuari aso té 25 fitxers
L'usuari aryn té 1543 fitxers
```

1. Escriu l'script que resol el problema anterior, no cal fer validació de l'entrada: **(0.75 Punts)**

```
#!/bin/bash

for user in "$@"; do
    home=$(getent passwd "$user" | cut -d: -f6)
    [ -d "$home" ] && count=$(find "$home" -type f -user "$user" 2>/dev/null | wc -l) &&
    echo "L'usuari $user té $count fitxers"
done
```

2. Adjunta aquí una captura de pantalla mostrant la execució de l'script: **(0.25 Punts)**

fotopo

3. Ara volem que es pugui indicar un valor i ens mostri els usuaris que tinguin més fitxers que aquest valor, per exemple:

```
aryn@host$ ./count_homes2.sh 30 aso aryn
Els usuaris:
aryn
Tenen més de 30 fitxers
```

(0.5 Punts)

```
#!/bin/bash
limit=$1; shift
echo "Els usuaris:"
for user in "$@"; do
    home=$(eval echo ~"$user")
    [ -d "$home" ] && count=$(find "$home" -type f -user "$user" 2>/dev/null | wc -l)
    [ "$count" -gt "$limit" ] && echo "$user"
done
echo "Tenen més de $limit fitxers"
```