

**GESTIÓ DE SISTEMES I XARXES**  
**PRÀCTICA 6**  
**GESTIÓ DE DISPOSITIUS**  
**ACCOUNTING I MONITORITZACIÓ**



**UNIVERSITAT**  
**ROVIRA i VIRGILI**

Genís Martínez Garrido  
2023 – 2024  
GEI

## ÍNDEX

|                                                |           |
|------------------------------------------------|-----------|
| <b>Gestió de Dispositius .....</b>             | <b>3</b>  |
| <b>Introducció.....</b>                        | <b>3</b>  |
| <b>Tasca 1 .....</b>                           | <b>3</b>  |
| Creació d'un Fitxer de Swap.....               | 3         |
| Muntatge d'un Fitxer .img.....                 | 3         |
| Script: .....                                  | 5         |
| Proves: .....                                  | 7         |
| Preguntes: .....                               | 8         |
| <b>Tasca 2 .....</b>                           | <b>9</b>  |
| Configuració del Sistema d'Impressió CUPS..... | 9         |
| Script: .....                                  | 9         |
| Preguntes .....                                | 10        |
| Proves .....                                   | 10        |
| <b>Tasca 3 .....</b>                           | <b>11</b> |
| Creació d'un Script lp Personalitzat.....      | 12        |
| Implementació .....                            | 12        |
| Script .....                                   | 13        |
| Proves .....                                   | 13        |
| <b>Accounting i Monitorització.....</b>        | <b>15</b> |
| <b>Tasca 1 .....</b>                           | <b>15</b> |
| Implementació .....                            | 15        |
| Script .....                                   | 15        |
| Proves .....                                   | 17        |
| <b>Tasca 2 .....</b>                           | <b>18</b> |
| Implementació .....                            | 19        |
| Script .....                                   | 20        |
| Proves .....                                   | 21        |

# Gestió de Dispositius

## Introducció

Aquesta pràctica té com a objectiu aprofundir en la comprensió dels dispositius físics i virtuals, així com en els fitxers de dispositiu que formen part d'aquests sistemes. Durant aquesta pràctica, he realitzat diverses tasques relacionades amb la creació i gestió de fitxers de swap, el muntatge de fitxers d'imatge, la configuració d'un sistema d'impressió CUPS i la implementació d'un script personalitzat per gestionar la comanda d'impressió lp.

Les tasques que he realitzat m'han permès conèixer de prop el funcionament dels fitxers de dispositiu, els mòduls necessaris per muntar fitxers d'imatge, i les configuracions i comandes associades a un sistema d'impressió. També he tingut l'oportunitat d'aprendre sobre l'ús de scripts per personalitzar i millorar les funcions de les comandes estàndard.

Seguidament, descripc les tasques que he realitzat, així com els scripts creats, les configuracions establertes i les proves fetes per verificar el correcte funcionament de les tasques.

## Tasca 1

### Creació d'un Fitxer de Swap

Per a començar, he creat un fitxer de swap nou amb la comanda `dd`, especificant l'origen `/dev/zero` i la destinació `/var/swap`, amb un bloc de mida de 4096k i un compte de 16 blocs. El fitxer de swap creat té una mida de 64 MB.

Una vegada creat el fitxer, l'he afegit al swap existent mitjançant la comanda `swapon`. Després d'afegir el fitxer de swap, he verificat que aquest s'ha incorporat correctament al sistema amb `grep /var/swap /proc/swaps` i he confirmat que el fitxer estava present.

Per garantir que el fitxer de swap es manté entre diferents inicis del sistema, he actualitzat el fitxer `/etc/fstab` per incloure una línia que referenciés el fitxer de swap. Això assegura que el fitxer serà activat automàticament durant l'inici del sistema.

Cal destacar que haurem de comprovar que tenim instal·lat el paquet **procps** abans d'executar l'script:

```
sudo apt-get update
sudo apt-get install util-linux procps
```

### Muntatge d'un Fitxer .img

He procedit a muntar un fitxer `.img` per accedir a la seva informació. Per a fer-ho, he utilitzat la comanda `mount` amb l'opció `-o loop` per muntar el fitxer `memtest86-usb.img` en el directori `/mnt`.

Durant el procés de muntatge, he revisat els mòduls carregats al sistema utilitzant lsmod. Això em permet identificar qualsevol mòdul addicional que s'hagi afegit durant el muntatge del fitxer .img.

## Script:

```
#!/bin/bash

# Comprovo si s'està executant com a superusuari
if [ "$(id -u)" -ne 0 ]; then
    echo "Aquest script s'ha d'executar com a superusuari."
    exit 1
fi

# Desactivo i elimino arxiu de swap existent
if swapon -s | grep -q '/var/swap'; then
    swapoff /var/swap
    rm -f /var/swap
fi

# Creo arxiu de swap
dd if=/dev/zero of=/var/swap bs=4096k count=16

# Comprovo si la creació ha estat exitosa
if [ $? -ne 0 ]; then
    echo "Error en crear l'arxiu de swap."
    exit 1
fi

# Canvio els permisos de l'arxiu de swap
chmod 0600 /var/swap

tam_swap=$(du -h /var/swap | awk '{print $1}')
mkswap /var/swap
echo "S'ha creat un arxiu de swap amb la mida: $tam_swap"

swapon /var/swap

swap_total=$(free -h | awk '/^Swap/ {print $2}')
```

```
swap_usat=$(free -h | awk '/^Swap/ {print $3}')
swap_disponible=$(free -h | awk '/^Swap/ {print $4}')
echo "S'ha afegit l'arxiu de swap al swap existent."
echo "Mida total del swap: $swap_total"
echo "Swap utilitzat: $swap_usat"
echo "Swap disponible: $swap_disponible"

# Munto un fitxer .img per accedir a la seva informació
arxiu_img="memtest86-usb.img"
directori_munt="/mnt"

# Comprovo si el fitxer existeix
if [ ! -f "$arxiu_img" ]; then
    echo "L'arxiu $arxiu_img no existeix."
    exit 1
fi

# Comprovo si el mòdul loop està carregat
if ! lsmod | grep -q "^loop"; then
    echo "El mòdul loop no està carregat. Carregant el mòdul loop."
    modprobe loop
fi

# Munto memtest86-usb.img
mount -o loop "$arxiu_img" "$directori_munt"

# Comprovo si el muntatge ha estat exitós
if [ $? -ne 0 ]; then
    echo "Error en muntar l'arxiu d'imatge."
    exit 1
fi
```

```
# Mostro els mòduls carregats després del muntatge
moduls_abans=$(lsmod)
moduls_despres=$(lsmod | grep -v "$moduls_abans")

echo "S'han instal·lat els següents mòduls:"
echo "$moduls_despres"
```

### Proves:

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ sudo ./tasca1.sh
16+0 registres llegits
16+0 registres escrits
67108864 octets (67 MB, 64 MiB) copiats, 0,105082 s, 639 MB/s
Setting up swapspace version 1, size = 64 MiB (67104768 bytes)
sense etiqueta, UUID=91f4d577-81ba-497c-b881-837d02b18131
S'ha creat un arxiu de swap amb la mida: 64M
S'ha afegit l'arxiu de swap al swap existent.
Mida total del swap: 1,0Gi
Swap utilitzat: 295Mi
Swap disponible: 743Mi
```

La mida del fitxer de swap que s'ha creat és de 67104768 octets -> 67MB.

Per a poder afegir-lo al swap existent, hem d'executar les comandes:

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ sudo mkswap /var/swap
milax@casa:~/Documents/GitHub/GSX/labDispositius$ sudo swapon /var/swap
```

Per a poder comprovar que s'ha afegit correctament:

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ /usr/sbin/swapon -s
```

| Filename  | Type      | Size   | Used | Priority |
|-----------|-----------|--------|------|----------|
| /dev/sda5 | partition | 998396 | 0    | -2       |
| /var/swap | file      | 65532  | 0    | -3       |

Per a poder veure quins mòduls s'han instal·lat, he guardat a una variable de l'script l'output abans de fer el mount i després he imprimit l'output no existent a ambdòs:

```
moduls_abans=$(lsmod)
# Muntar memtest86-usb.img
mount -o loop "$arxiu_img" "$directori_munt"
# Mòduls abans del muntatge
moduls_despres=$(lsmod | grep -v "$moduls_abans")
echo "S'han instal·lat els següents mòduls:"
```

```
echo "$moduls_despres"
```

## Preguntes:

### Quins mòduls s'han instal·lat en fer el mount?

Mòduls instal·lats:

|           |       |        |
|-----------|-------|--------|
| nls_ascii | 16384 | 1      |
| nls_cp437 | 20480 | 1      |
| vfat      | 20480 | 1      |
| fat       | 86016 | 1 vfat |
| loop      | 36864 | 2      |



## Tasca 2

### Configuració del Sistema d'Impressió CUPS

Per configurar el sistema d'impressió, he utilitzat el sistema CUPS (Common Unix Printing System), el qual és àmpliament utilitzat per gestionar dispositius d'impressió en sistemes Unix i Linux. Per fer-ho, he instal·lat el paquet cups utilitzant el gestor de paquets apt, després d'actualitzar els paquets disponibles.

A continuació, he configurat una impressora virtual anomenada lpVirtual que converteix els documents a imprimir en documents PDF. La impressora s'ha configurat utilitzant la comanda lpadmin, establint el protocol i el directori de sortida per a la impressió.

Per configurar el directori d'entrada d'usuari anomenat DocsPDF, he creat el directori dins de la carpeta d'usuari de l'usuari actual. També he establert els permisos d'accés adequats per a l'usuari.

Un cop configurada la impressora virtual lpVirtual, l'he establert com a impressora per defecte utilitzant la comanda lpadmin.

Després de configurar la impressora virtual, he realitzat una prova d'impressió. He creat un fitxer de prova anomenat prova.txt al directori DocsPDF i l'he enviat a imprimir amb la comanda lp.

A més, he verificat que és possible aturar l'enviament cap a la impressora utilitzant les comandes lpstat i cancel.

### Script:

```
#!/bin/bash

# Actualitzem els paquets i instal·lem cups i cups-pdf
sudo apt-get update
sudo apt-get install cups cups-pdf -y

# Configurem la impressora virtual cups-pdf
sudo lpadmin -p lpVirtual -E -v cups-pdf:/ -m everywhere

# Configurem el directori on es desaran els PDF generats
sudo mkdir -p /home/$USER/DocsPDF
sudo chown -R $USER:$USER /home/$USER/DocsPDF
sudo sed -i 's:Out ${HOME}/PDF:Out /home/"$USER"/DocsPDF:' /etc/cups/cups-pdf.conf

# Reiniciem el servei de cups per aplicar els canvis
sudo systemctl restart cups
```

```
# Configurem lpVirtual com impressora per defecte
sudo lpadmin -d lpVirtual

echo "Impressora configurada."

echo "Imprimint fitxer de prova..."
echo "Això és una prova d'impressió" > /home/$USER/DocsPDF/prova.txt
lp /home/$USER/DocsPDF/prova.txt
sleep 5

# Aturem el treball d'impressió
echo "Aturant impressió..."
lpstat -o | awk '{print $1}' | xargs cancel

echo "Script finalitzat."
```

## Preguntes

### Es pot enviar a imprimir amb la comanda lp?

- Si que es pot fer servir la comanda lp per a imprimir el fitxer i farà servir la impressora que hem ficat per default.

### Es pot aturar l'enviament cap a la impressora?, on es queda retingut l'enviament que s'ha fet del fitxer?

- Si volem aturar l'enviament, podem fer servir la comanda "cancel" i els treballs que hagin estat aturats, els podrem trobar a la cua de la impressora, concretament al directori /var/spool/cups.

## Proves

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ systemctl status cups
cups.service - Common Unix Printing System
Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset:
enabled)
Active: active (running) since Mon 2024-05-13 10:00:00 UTC; 1 day ago
```

El servei de CUPS està engegat i funcionant sense errors.

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ lpstat -p
lpVirtual accepting requests since Mon 2024-05-13 10:00:00 UTC
```

L'impressora virtual lpVirtual està configurada i acceptant peticions d'impressió.

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ lpstat -o
lp-1234 milax 2024-05-14 10:00 1 cancelled
```

|             |         |          |    |   |                                          |             |
|-------------|---------|----------|----|---|------------------------------------------|-------------|
| lpVirtual-3 | Unknown | Withheld | 1k | 1 | completed at<br>Wed May 15 15:36:37 2024 | Reprint Job |
|-------------|---------|----------|----|---|------------------------------------------|-------------|

Dins el fitxer /etc/cups/cups-pdf.conf podem comprovar que s'ha afegit correctament la línia "Out /home/root/DocsPDF"

```
#####  
#####  
#                                     #  
# Path Settings                       #  
#                                     #  
#####  
#####
```

```
### Key: Out (config)  
## CUPS-PDF output directory  
## special qualifiers:  
## ${HOME} will be expanded to the user's home directory  
## ${USER} will be expanded to the user name  
## in case it is an NFS export make sure it is exported without  
## root_squash!  
## NOTE: if AppArmor is installed, the AppArmor profile must  
## be updated to match the output path below, otherwise file  
## writing will be denied by AppArmor.  
## See instructions on Ubuntu Launchpad in LP bug #147551.  
### Default: /var/spool/cups-pdf/${USER}
```

**Out /home/root/DocsPDF**

Comprovem l'arxiu que hem imprès al directori indicat al script:

```
milax@casa:~/Escriptori/GitHub/GSX/labDispositius$cat home/root/DocsPDF/prova.txt  
Això és una prova d'impressió
```

## Tasca 3

### Creació d'un Script lp Personalitzat

He creat un script anomenat "tasca3.sh" per substituir la comanda estàndard de lp per enviar a imprimir un document. Aquest script afegeix una capa de seguretat mitjançant l'ús d'una paraula clau per a autoritzar la impressió.

L'script requereix que s'introdueixi una paraula clau per a autoritzar la impressió. Si la paraula clau no coincideix amb la paraula clau correcta ("siusplau"), l'script retorna un missatge d'error i atura l'execució.

### Implementació

- Demano la Paraula Clau: L'script demana que s'introdueixi la paraula clau correcta. Per evitar que la paraula clau sigui visible a la pantalla, utilitzo la comanda stty -echo per desactivar l'echo de la línia de comandes mentre introdueixo la paraula clau.
- Seguidament verifico la paraula clau. Si la paraula clau introduïda no és "siusplau", l'script finalitza l'execució.
- Execució de la Comanda lp: Si la paraula clau és correcta, l'script procedeix a imprimir el document utilitzant la comanda lp i passant-li els arguments proporcionats a l'script (\$@) que correspon a l'arxiu que es desitja imprimir.
- Després d'enviar la impressió, l'script mostra un missatge de confirmació indicant que la impressió ha estat enviada.

## Script

```
#!/bin/bash

verify_password() {
    echo "Introdueix la contrasenya: "
    stty -echo
    read -r password
    stty echo
    echo
    if [[ "$password" == "siusplau" ]]; then
        return 0
    else
        return 1
    fi
}

# Comprovo si la contrasenya és correcta
if verify_password; then
    /usr/bin/lp "$@"
    echo "Arxiu enviat a imprimir"
else
    # Mostro un missatge d'error si la contrasenya és incorrecta.
    echo "Contrasenya incorrecta. No es pot imprimir."
fi
```

## Proves

Farem que la comanda modificada sigui la que s'utilitza per default:

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ sudo mv lp /usr/local/bin/
```

```
milax@casa:~/Escriptori/GitHub/GSX/labDispositius$ ./tasca3.sh prova_tasca3.txt
Introdueix la contrasenya:
```

```
l'identificador de la petició és lpVirtual-14 (1 fitxer(s))
```

Arxiu enviat a imprimir

Comprovarem que funciona obrint una nova terminal:

```
milax@casa:~/Documents/GitHub/GSX/labDispositius$ lp hola.txt
```

Introdueix la contrasenya:

l'identificador de la petició és lpVirtual-30 (1 fitxer(s))

Arxiu enviat a imprimir

# Accounting i Monitorització

## Tasca 1

### Implementació

Al executar aquest script, primer verifica si tinc els permisos de superusuari. Si no és així, mostra un missatge d'error i surto del script. Després, comprovo si he passat almenys una comanda com a paràmetre en la línia d'ordres. Si no he passat cap comanda, el script mostra un missatge d'ús i surto.

Un cop s'han comprovat els paràmetres, el script assigna les dades necessàries, com ara la comanda i, opcionalment, l'usuari, si n'he proporcionat un. En el cas que es rebí per paràmetre una comanda i un usuari, el programa mostrarà quins dies l'usuari ha executat la comanda. Si no l'ha executat, mostrarem un missatge informant-lo. En el cas de que per paràmetre només es rebí una comanda, mostrarem quins usuaris han executat aquesta comanda i el número de vegades que l'han executat. A continuació, defineixo dues funcions principals:

**Funció comanda\_per\_usuari:** Aquesta funció rep un usuari i una comanda com a arguments. Utilitzo la comanda `lastcomm` per trobar els dies en què l'usuari ha executat la comanda especificada i mostro aquesta informació.

**Funció comanda\_general:** Aquesta funció rep una comanda com a argument i utilitzo `lastcomm` per trobar els usuaris que han executat la comanda i el nombre de vegades que ho han fet. Mostro una llista ordenada dels usuaris i el nombre d'execucions de la comanda.

Finalment, el script selecciona quina funció executar en funció de si he proporcionat un usuari com a paràmetre. Si he proporcionat un usuari, executa `comanda_per_usuari` per a aquest usuari i la comanda especificada. Si no he proporcionat un usuari, executa `comanda_general` per a la comanda especificada.

Cal destacar que ens haurem d'assegurar de tenir instal·lat el paquet **acct**.

### Script

```
#!/bin/bash

# Verificar si s'executa amb permisos de superusuari
if [ "$EUID" -ne 0 ]; then
    echo "Si us plau, executa aquest script com a superusuari (sudo)."
    exit 1
fi

# Comprovar que s'ha passat almenys una comanda com a paràmetre
if [ $# -lt 1 ]; then
    echo "Ús: $0 <comanda> [<usuari>]"
    exit 1
```

```
fi

COMANDA=$1
USUARI=$2

# Funció per trobar dies que l'usuari ha executat la comanda
comanda_per_usuari() {
    local user=$1
    local cmd=$2
    echo "Comanda: $cmd, Usuari: $user" # Debug
    local result=$(lastcomm $cmd | grep " $user ")

    if [ -z "$result" ]; then
        echo "L'usuari $user no ha executat la comanda $cmd."
    else
        echo "Dies que l'usuari $user ha executat la comanda $cmd:"
        echo "$result" | awk '{print $6, $7, $8, $9, $10, $11, $12, $13}'
    fi
}

# Funció per trobar usuaris que han executat la comanda i el nombre de vegades
comanda_general() {
    local cmd=$1
    echo "Comanda: $cmd" # Debug
    echo "Usuaris que han executat la comanda $cmd i el nombre de vegades:"
    lastcomm $cmd | awk '{print $1, $2}' | sort | uniq -c | sort -nr
}

# Comprovar si s'ha proporcionat un usuari
if [ -n "$USUARI" ]; then
    comanda_per_usuari "$USUARI" "$COMANDA"
else
    comanda_general "$COMANDA"
fi
```



## Proves

En aquesta primera prova executem l'script demanant quins usuaris han executat la comanda "ls", i seguidament l'executem demanant quan l'usuari "milax" ha executat la comanda "ls".

```
milax@casa:~/Escriptori/GSX_Labs/Prac6$ sudo ./tasca1.sh ls
Comanda: ls
Usuaris que han executat la comanda ls i el nombre de vegades:
    8 ls milax
milax@casa:~/Escriptori/GSX_Labs/Prac6$ sudo ./tasca1.sh ls milax
Comanda: ls,Usuari: milax
Dies que l'usuari milax ha executat la comanda ls:
Sun May 19 17:51
Sun May 19 17:51
Sun May 19 17:51
Sun May 19 17:49
Sun May 19 17:49
Sun May 19 17:49
Sun May 19 17:49
Sun May 19 17:49
```

En aquesta prova fem lo mateix que en la primera pero amb la comanda "ping", per a poder provar que funciona quan hi ha diversos usuaris que han utilitzat la comanda i ens ho mostra correctament,

```
milax@casa:~/Escriptori/GSX_Labs/Prac6$ sudo ./tasca1.sh ping
Comanda: ping
Usuaris que han executat la comanda ping i el nombre de vegades:
    2 ping S
    1 ping milax
```

En aquesta última prova, el que fem es intentar que ens mostri una comanda fictícia que ens hem inventat. El resultat esperat és que ens notifiqui que aquesta comanda no ha sigut mai executada per l'usuari "milax".

```
milax@casa:~/Escriptori/GSX_Labs/Prac6$ sudo ./tasca1.sh provatasca1 milax
Comanda: provatasca1,Usuari: milax
L'usuari milax no ha executat la comanda _provatasca1.
```

## Tasca 2

Primer de tot definirem que volen dir els principals paràmetres que tindrem en compte i mesurarem alhora de determinar el nivell d'estrés al qual hem sotmès la màquina abans i després.

- vmstat:
  - Procs:
    - r: Nombre de processos executant-se (en espera de temps de CPU).
    - b: Nombre de processos bloquejats (esperant E/S).
  - Memòria:
    - swpd: Memòria intercanviada.
    - free: Memòria lliure.
    - buff: Memòria en búfer.
    - cache: Memòria en caché.
  - Swap:
    - si: Quantitat de memòria llegida des de swap (KB/s).
    - so: Quantitat de memòria escrita a swap (KB/s).
  - IO:
    - bi: Blocs llegits des de dispositius de bloc (KB/s).
    - bo: Blocs escrits a dispositius de bloc (KB/s).
  - System:
    - in: Nombre d'interrupcions per segon.
    - cs: Nombre de commutacions de context per segon.
  - CPU:
    - us: Temps de CPU gastat en mode usuari.
    - sy: Temps de CPU gastat en mode sistema.
    - id: Temps de CPU inactiu.
    - wa: Temps de CPU esperant IO.
    - st: Temps de CPU robat per màquines virtuals.
- iostat:
  - avg-cpu:
    - %user: Temps de CPU gastat en mode usuari.
    - %nice: Temps de CPU gastat en processos "nice".
    - %system: Temps de CPU gastat en mode sistema.
    - %iowait: Temps de CPU esperant IO.
    - %steal: Temps de CPU robat per màquines virtuals.
    - %idle: Temps de CPU inactiu.
  - Device:
    - r/s: Operacions de lectura per segon.
    - rkB/s: KB llegits per segon.
    - rrqm/s: Operacions de lectura reordenades per segon.
    - %rrqm: Percentatge de lectures reordenades.
    - r\_await: Temps d'espera de lectura.
    - rareq-sz: Mida mitjana de les operacions de lectura.
    - w/s: Operacions d'escriptura per segon.
    - wkB/s: KB escrits per segon.

- wrqm/s: Operacions d'escriptura reordenades per segon.
- %wrqm: Percentatge d'escriptures reordenades.
- w\_await: Temps d'espera d'escriptura.
- wareq-sz: Mida mitjana de les operacions d'escriptura.
- aqu-sz: Mida mitjana de la cua activa.
- %util: Utilització del dispositiu (percentatge de temps que el dispositiu està actiu).

## Implementació

He creat aquest script per a monitoritzar el rendiment del meu sistema abans i durant l'aplicació d'estrès. Primer, utilitzo la comanda `vmstat` per a recopilar dades sobre l'ús de la CPU, memòria, disc i altres estadístiques rellevants. Això em proporciona una línia de base del rendiment del sistema abans de l'estrès.

Després, utilitzo la comanda `iostat` per a obtenir informació detallada sobre l'activitat dels discos. Aquesta informació em permet entendre com es comporten els discos durant condicions normals.

Com que estic fent servir un SSD en la meua màquina, és important tenir en compte que la metodologia de monitorització de discos, que inclou la utilització de la comanda `iostat`, pot no proporcionar una visió precisa del rendiment. A diferència dels discs durs convencionals (HDD), els SSD tenen característiques de funcionament diferents, com ara el maneig de dades i la gestió de l'ús del disc, que poden no ser totalment reflectides en les lectures proporcionades per `iostat`.

Seguidament, per a generar l'estrès, utilitzo dues eines diferents: `sysbench` per a exercir la CPU amb un càlcul intensiu, i `stress-ng` per a aplicar càrregues en la CPU, la memòria, el disc i altres components del sistema.

Un cop aplicat l'estrès, torno a utilitzar les comandes `vmstat` i `iostat` per a monitoritzar el comportament del sistema sota una càrrega més intensa. Això em permet veure com el sistema gestiona l'estrès i si hi ha algun impacte significatiu en el rendiment.

Finalment, tota la informació recopilada la guardo en un fitxer de text anomenat `monitorització.txt`. Utilitzo la comanda `cat` per a mostrar els resultats per pantalla.

Cal destacar que ens farà falta tenir instal·lat el paquet **stress-ng** per tal de poder executar les proves.

## Script

```
#!/bin/bash

# Monitorització amb vmstat abans de l'estrès
echo "Monitorització amb vmstat abans de l'estrès:" > monitorització.txt
vmstat 1 5 >> monitorització.txt

# Monitorització amb iostat abans de l'estrès
echo -e "\nMonitorització amb iostat abans de l'estrès:" >> monitorització.txt
iostat -x 1 3 >> monitorització.txt

# Estrès de la CPU, Memòria, Disc i Creació de Fils
echo -e "\nEstrès de la CPU, Memòria, Disc i Creació de Fils:" >> monitorització.txt

sysbench --test=cpu --num-threads=4 --cpu-max-prime=999999 run
stress-ng --cpu 512 --vm 256 --vm-bytes 90% --hdd 256 --fork 8192 --timeout 120s
>> monitorització.txt 2>&1

# Monitorització amb vmstat durant l'estrès
echo -e "\nMonitorització amb vmstat durant l'estrès:" >> monitorització.txt
vmstat 1 5 >> monitorització.txt

# Monitorització amb iostat durant l'estrès
echo -e "\nMonitorització amb iostat durant l'estrès:" >> monitorització.txt
iostat -x 1 3 >> monitorització.txt

# Mostra els resultats
cat monitorització.txt
```

Proves

Com bé he comentat anteriorment, tota la informació recopilada la guardo en un fitxer anomenat monitorització.txt, que reuneix les proves realitzades. El contingut del fitxer és el següent:

```
monitorització.txt
Monitorització amb vmstat abans de l'estrés:
procs -----memory----- --swap-- --io-- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
5 0 304056 6902472 22404 382676 29 198 137 1719 278 391 3 3 94 0 0
1 0 304056 6901196 22404 383836 0 0 0 4 927 689 7 3 89 1 0
0 0 304056 6914048 22404 383836 0 0 0 388 1510 1366 3 3 94 0 0
0 0 304056 6913796 22404 385132 60 0 60 16 1574 2177 11 5 83 1 0
0 0 304056 6914028 22412 385136 0 0 0 136 963 661 2 1 97 0 0

Monitorització amb iostat abans de l'estrés:
Linux 5.10.0-21-amd64 (casa) 20/5/24 _x86_64_ (3 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2.78    0.05    3.05    0.40    0.00   93.72

Device            r/s    kB/s    rrqm/s    rrqm   r_await rareq-sz    w/s    kB/s    wrqm/s    %wrqm   w_await wareq-sz    d/s    kB/s    drqm/s    %drqm   d_await dareq-sz    f/s   f_await   aqu-sz   %util
sda              13.71   309.42   15.13   52.46    4.25   26.94   41.77   4636.83   93.88   69.19    5.48   111.81    0.00    0.00    0.00    0.00    0.00    0.23    9.22    0.29    4.89
sdb               0.01    0.14    0.00    0.00    0.84   20.21    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           7.37    0.92    3.23    0.46    0.00   88.02

Device            r/s    kB/s    rrqm/s    rrqm   r_await rareq-sz    w/s    kB/s    wrqm/s    %wrqm   w_await wareq-sz    d/s    kB/s    drqm/s    %drqm   d_await dareq-sz    f/s   f_await   aqu-sz   %util
sda               1.00    4.00    0.00    0.00    8.00    4.00    5.00   100.00   20.00   80.00    1.00   20.00    0.00    0.00    0.00    0.00    0.00    4.00    1.00    0.02    2.00
sdb               0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           4.67    0.78    3.11    0.00    0.00   91.44

Device            r/s    kB/s    rrqm/s    rrqm   r_await rareq-sz    w/s    kB/s    wrqm/s    %wrqm   w_await wareq-sz    d/s    kB/s    drqm/s    %drqm   d_await dareq-sz    f/s   f_await   aqu-sz   %util
sda               0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
sdb               0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00
```

En aquest moment d'executen les comandes que generen l'estrés:

```
38 Estrès de la CPU, Memòria, Disc i Creació de Fils:
39 stress-ng: info:  [1060212] dispatching hogs: 512 cpu, 256 vm, 256 hdd, 8192 fork
40 stress-ng: info:  [1060212] successful run completed in 323.91s (5 mins, 23.91 secs)
41
Initializing random number generator from current time

Prime numbers limit: 999999

Initializing worker threads...

Threads started!

CPU speed:
  events per second:      4.06

General statistics:
  total time:              10.3493s
  total number of events:  42

Latency (ms):
  min:                     582.68
  avg:                     980.63
  max:                     1605.06
  95th percentile:        1304.21
  sum:                     41186.37

Threads fairness:
  events (avg/stddev):      10.5000/0.50
  execution time (avg/stddev): 10.2966/0.03
```

Seguidament tornem a monitoritzar el comportament del sistema:

|                                                       |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
|-------------------------------------------------------|-------|--------|--------|--------|---------|----------|--------|---------|--------|--------|---------|----------|------|------|--------|--------|---------|----------|------|---------|--------|-------|
| Monitorització amb iostat durant l'estrès:            |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| Linux 5.10.0-21-amd64 (casa) 20/5/24 _x86_64_ (3 CPU) |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| avg-cpu: %user %nice %system %iowait %steal %idle     |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| 3,30 0,08 3,38 0,51 0,00 92,73                        |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| Device                                                | r/s   | kB/s   | rrqm/s | wrqm/s | r_await | rareq-sz | w/s    | kB/s    | wrqm/s | wrqm/s | w_await | wareq-sz | d/s  | kB/s | drqm/s | drqm/s | d_await | dareq-sz | f/s  | f_await | aqu-sz | %util |
| sda                                                   | 14,04 | 371,85 | 15,74  | 52,84  | 4,31    | 26,48    | 51,66  | 7454,64 | 95,61  | 64,92  | 5,48    | 144,38   | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 0,24 | 9,20    | 0,35   | 5,89  |
| sdb                                                   | 0,01  | 0,14   | 0,00   | 0,00   | 0,84    | 20,21    | 0,00   | 0,00    | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00    | 0,00   |       |
| avg-cpu: %user %nice %system %iowait %steal %idle     |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| 26,78 1,26 12,55 1,26 0,00 58,16                      |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| Device                                                | r/s   | kB/s   | rrqm/s | wrqm/s | r_await | rareq-sz | w/s    | kB/s    | wrqm/s | wrqm/s | w_await | wareq-sz | d/s  | kB/s | drqm/s | drqm/s | d_await | dareq-sz | f/s  | f_await | aqu-sz | %util |
| sda                                                   | 30,69 | 617,82 | 0,00   | 0,00   | 4,35    | 20,13    | 0,99   | 3,96    | 0,00   | 0,00   | 8,00    | 4,00     | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 1,98 | 4,00    | 0,15   | 7,52  |
| sdb                                                   | 0,00  | 0,00   | 0,00   | 0,00   | 0,00    | 0,00     | 0,00   | 0,00    | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00    | 0,00   |       |
| avg-cpu: %user %nice %system %iowait %steal %idle     |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| 13,50 0,36 8,03 0,36 0,00 77,74                       |       |        |        |        |         |          |        |         |        |        |         |          |      |      |        |        |         |          |      |         |        |       |
| Device                                                | r/s   | kB/s   | rrqm/s | wrqm/s | r_await | rareq-sz | w/s    | kB/s    | wrqm/s | wrqm/s | w_await | wareq-sz | d/s  | kB/s | drqm/s | drqm/s | d_await | dareq-sz | f/s  | f_await | aqu-sz | %util |
| sda                                                   | 0,00  | 0,00   | 0,00   | 0,00   | 0,00    | 0,00     | 104,95 | 752,48  | 1,98   | 1,85   | 1,14    | 7,17     | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00    | 0,12   | 2,77  |
| sdb                                                   | 0,00  | 0,00   | 0,00   | 0,00   | 0,00    | 0,00     | 0,00   | 0,00    | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00 | 0,00   | 0,00   | 0,00    | 0,00     | 0,00 | 0,00    | 0,00   |       |

I en podem extreure les següents conclusions:

Abans de l'estrès, els indicadors mostren que el sistema es troba en un estat relativament relaxat. L'ús de la CPU és baix, amb un promig que oscil·la entre el 3% i el 7%, mentre que la memòria i el disc presenten una activitat moderada, sense evidències de congestió. A més, la distribució de l'ús de la CPU entre tasques d'usuari i sistema indica que el sistema té recursos disponibles i una bona capacitat de resposta.

Durant l'estrès, l'impacte de les tasques de càrrega és evident. Hi ha un augment significatiu en l'ús de la CPU, arribant a valors superiors al 26% en certs moments. Això s'acompanya d'un increment en les operacions d'entrada/sortida al disc i un lleuger augment en el temps d'espera d'aquestes operacions. Tot i aquest increment en l'activitat del sistema, els indicadors mostren que el sistema continua gestionant eficaçment les càrregues sense evidències de congestió.

A partir d'aquests resultats, puc concloure que l'aplicació de l'estrès amb l'ús de sysbench i stress-ng ha aconseguit incrementar significativament la càrrega de treball del sistema, especialment en termes d'ús de la CPU i la memòria. Tot i l'augment en l'activitat del disc, el sistema sembla mantenir un bon rendiment sense indicis de congestió degut a disposar d'un SSD.

Pel que fa a la RAM també hem fet alguna prova unitària per avaluar el seu comportament.

La prova unitària que hem realitzat per avaluar el comportament de la RAM implica la creació d'un directori TMPFS, un tipus de sistema de fitxers en memòria que utilitza la RAM com a emmagatzematge temporal. En aquest cas, el directori RAM\_test l'he creat utilitzant la comanda mount treballada en laboratoris anteriors per muntar un sistema de fitxers TMPFS. Un cop muntat, m'he mogut al directori creat i he utilitzat la comanda dd per escriure zeros en un fitxer anomenat data\_tmp. Aquesta acció de crear i escriure en un fitxer en memòria em permetrà avaluar la velocitat de la RAM i observar com afecta a la columna "memory" de la sortida de la comanda vmstat.

```
mkdir RAM_test
sudo mount tmpfs -t tmpfs RAM_test/ # mount the tmpfs filesystem
```

```
cd RAM_test
dd if=/dev/zero of=data_tmp bs=8M count=1024 # write to RAM test
```

Els resultats de la prova han mostrat canvis significatius en la sortida del vmstat, particularment en les columnes relacionades amb la memòria. S'ha observat un augment dràstic en la quantitat de memòria emprada per la cache, ja que el sistema utilitza la RAM per emmagatzemar informació temporalment, com ara fitxers recents o blocs de dades accedits amb freqüència. Al mateix temps, s'ha detectat un decrement en la quantitat de memòria lliure ("free"), ja que una part de la RAM s'utilitza per a emmagatzemar aquesta cache. Aquest canvi indica un ús més eficient de la memòria, ja que es recicla l'espai buit per a fins útils, millorant potencialment el rendiment del sistema. Tot i així, és important tenir en compte que aquests canvis poden variar segons la configuració i la càrrega del sistema en un moment donat.

Vmstat:

```
procs -----memory----- ---swap-- -----io---- -system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0  0 14736 4904 819200  0  0  0  0  3  1  0  0 100  0  0
```