



ESCOLA TÈCNICA SUPERIOR  
D'ENGINYERIA  
Universitat Rovira i Virgili



# GSX- Pràctica 6

**Curs 2023-24**

**Estudiant:** Rubén López Martínez

**Data d'entrega:** 22/05/2024

# Índex

<b>1. creaSwap.sh.....</b>	<b>4</b>
<b>1.1. Codi del script.....</b>	<b>4</b>
<b>1.2. Path absolut del fitxer.....</b>	<b>5</b>
<b>1.3. Permisos del fitxer .....</b>	<b>5</b>
<b>1.4. Propietari i grup del fitxer .....</b>	<b>5</b>
<b>1.5. Preguntes script.....</b>	<b>5</b>
<b>1.6. Joc de proves .....</b>	<b>6</b>
<b>2. Sistema d'impressió .....</b>	<b>8</b>
<b>2.1. Codi del script.....</b>	<b>8</b>
<b>2.2. Path absolut del fitxer.....</b>	<b>9</b>
<b>2.3. Permisos del fitxer .....</b>	<b>9</b>
<b>2.4. Propietari i grup del fitxer .....</b>	<b>9</b>
<b>2.5. Preguntes script.....</b>	<b>9</b>
<b>2.6. Joc de proves .....</b>	<b>9</b>
<b>3. lp .....</b>	<b>11</b>
<b>3.1. Codi del script.....</b>	<b>11</b>
<b>3.2. Path absolut del fitxer.....</b>	<b>11</b>

3.3.	Permisos del fitxer .....	11
3.4.	Propietari i grup del fitxer .....	11
3.5.	Joc de proves .....	12
4.	consultaComanda.sh.....	13
4.1.	Codi del script.....	13
4.2.	Path absolut del fitxer.....	13
4.3.	Permisos del fitxer .....	14
4.4.	Propietari i grup del fitxer .....	14
4.5.	Joc de proves .....	14
5.	Monitorització de recursos.....	16
5.1.	Vmstat .....	17
5.1.1.	Estudi .....	17
5.1.2.	Conclusions resultats obtinguts i camps que varien en funció dels recursos explotats .....	18
5.2.	lostat .....	19
5.2.1.	Estudi .....	19
5.2.2.	Conclusions resultats obtinguts i camps que varien en funció dels recursos explotats .....	20

# 1. creaSwap.sh

## 1.1. Codi del script

```
#!/bin/bash

# Afegim PATH per poder executar comandes sense necessitat de sudo
PATH="$PATH:/usr/sbin/"

# Demanem privilegis per poder crear l'area de swap
if [[ $EUID -ne 0 ]] ; then
echo "Necessitem privilegis per executar aquest script"
exit 1
fi

# Comprovem si hi ha àrea de swap
if [[ -z $(swapon -s | grep /var/swap) ]] ; then
echo "Creants fitxer de swap..."

# Creem fitxer de 64 MB (4096k (midaBloc) x 16 (blocs) = 65536k = 64 MB)
dd if=/dev/zero of=/var/swap bs=4096k count=16 &> /dev/null

# Donem els permissos segurs a l'area de swap
chmod 0600 /var/swap

# Crear àrea swap
mkswap /var/swap

# Afegim àrea de swap al sistema
swapon /var/swap
else
echo "Fitxer de swap ja creat!"
fi

# Verificar que s'ha creat el fitxer de swap
echo "-----"
echo "Fitxers de swap actius..."
swapon -s
echo "-----"

echo "Memoria de swap total disponible..."
free -h
echo "-----"

# Moduls previs al montatge de la nova imatge
modulsPre=$(lsmod)

# Montar fitxer .img
if [[ ! $(df -h | grep "loop") ]] ; then
echo "Montant imatge memtest86..."
```

```
mount -o loop /home/milax/Documents/GSX/lab12/memtest86-4.3.7-usb/memtest86-usb.img /mnt
fi
```

```
# Afegim l'entrada al /etc/fstab per manternir-ho quan es faci el boot del sistema
if ! grep -q "/var/swap" /etc/fstab; then
echo "/var/swap none swap defaults 0 0" >> /etc/fstab
systemctl daemon-reload
fi
```

```
echo "Imatge memtest86 montada"
df -h | grep "loop"
```

```
# Moduls previs al montatge de la nova imatge
modulsAct=$(lsmod)
```

```
# Moduls nous instalats per la imatge memtest86
echo "-----"
echo "Moduls nous (memtest86)..."
```

```
diff <(echo "$modulsPre") <(echo "$modulsAct") | grep '^>' | sed 's/^> //'
```

## 1.2. Path absolut del fitxer

➔ /home/milax/Documents/GSX/lab12/creaSwap.sh

## 1.3. Permisos del fitxer

➔ Octal => 700 (-rwx----- 1 root root 1752 16 de maig 00:02 creaSwap.sh)

## 1.4. Propietari i grup del fitxer

- Propietari => root
- Grup => root

## 1.5. Preguntes script

- **Quina mida de fitxer s'ha creat?** 67104768 bytes (64MiB).
- **Com l'afegim al swap que ja existent?** Amb la comanda *swapon /var/swap*.
- **Com ho podem verificar?** Amb la comanda *swapon -s* o amb *free -h*.
- **Com es pot mantenir entre diferents boots de la màquina?** Afegint-ho al fitxer */etc/fstab*: *"/var/swap none swap defaults 0 0"*.
- **Quins mòduls s'han instal·lat en fer el mount?**

```
Moduls nous (memtest86)...
nls_ascii                16384  1
nls_cp437                20480  1
vfat                    24576  1
loop                    32768  2
```

## 1.6. Joc de proves

Al mateix script d'execució, s'han afegit les diferents sortides per tal de comprovar si el que es demanava es correspon amb el que s'ha realitzat:

- Creació de la nova àrea de swap amb la mida corresponent esmentada:

```
Creants fitxer de swap...
Setting up swapspace version 1, size = 64 MiB (67104768 bytes)
sense etiqueta, UUID=588b95a4-a9a5-4cee-9ea6-e12eea9840e7
```

- Verificació de la creació del fitxer de swap a partir de la comanda `swapon -s` (veient així els fitxers de swap actius):

```
Fitxers de swap actius...
Filename                                Type      Size      Used      Priority
/var/swap                              file      65532     0         -3
/dev/sda5                              partition 998396    0         -2
```

- Comprovem la memòria de swap total disponible a partir de la comanda `free -h` (per tal de veure la quantitat total de memòria d'intercanvi disponible, així com quant s'està utilitzant):

```
Memoria de swap total disponible...
              total        used        free      shared  buff/cache   available
Mem:          5,7Gi         1,6Gi         3,6Gi         25Mi         757Mi         4,1Gi
Swap:          1,0Gi           0B          1,0Gi
```

\* Ens indica que el sistema té un total de 1.0 gigabytes d'espai de swap disponible (1,0Gi) i, actualment, no està sent utilitzat (0B). Per tant, podem confirmar que l'àrea de swap està disponible per al seu ús.

- També, en quant a la imatge que es demanava muntar, fent un `df -h` (en aquest cas de la imatge muntada) una vegada muntada, podem comprovar-ho:

```
Montant imatge memtest86...
Imatge memtest86 muntada
/dev/loop0      1,4M  513K   911K  37% /mnt
```

\* Veiem que es mostra que l'arxiu `.img` ha sigut muntat correctament en el directori `/mnt` del sistema amb una mida total de 1,4 megabytes. També veiem que l'espai utilitzat en el dispositiu és d'un 37%.

\* També, s'afegeix una entrada al `/etc/fstab` que garanteix que aquest muntatge es mantingui en els boots del sistema.

```
milax@casa:~/Documents/6SX/lab12$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options>          <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=3070be3e-e632-4417-bec9-229981d8e64b /          ext4      errors=remount-ro 0    1
# swap was on /dev/sda5 during installation
UUID=d36056c7-afc1-4561-9944-95bc2d46bbdf none        swap      sw              0    0
tmpfs /empresa/usuarios/AnnaGS/tmp tmpfs size=100M,defaults,users 0 0
tmpfs /empresa/usuarios/JoanJosepLA/tmp tmpfs size=100M,defaults,users 0 0
tmpfs /empresa/usuarios/DanielSR/tmp tmpfs size=100M,defaults,users 0 0
tmpfs /empresa/usuarios/MariaPG/tmp tmpfs size=100M,defaults,users 0 0
/var/swap none swap defaults 0 0
```

- Finalment, podem veure els mòduls que s'han instal·lat en fer el mount, fent un lsmod abans i després del muntatge i fer la comparació per treure els nous mòduls:

```
Moduls nous (memtest86)...\nnls_ascii          16384  1\nnls_cp437          20480  1\nvfat               24576  1\nloop               32768  2
```

## 2. Sistema d'impressió

### 2.1. Codi del script

Com que he seguit els passos utilitzant la part gràfica, no hi ha script; per tant, explicaré els passos que he dut a terme per configurar la impressora esmentada a l'enunciat.

- 1) Instal·lem els paquets de cups i cups-pdf primerament si no els tenim, i fem un restart de cups després de la instal·lació o actualització:
  - `sudo apt update`
  - `sudo apt install cups cups-pdf`
  - `systemctl restart cups`
- 2) Entrem al <http://localhost:631/> a partir del nostre navegador de la imatge de Debian.
- 3) A la part d'Administration, afegim una nova impressora (Add Printer) amb les següents característiques:

#### lpVirtual (Idle, Accepting Jobs, Not Shared, Server Default)



- 4) Configurem que aquesta nova impressora (lpVirtual) sigui la impressora per defecte:
  - `lpadmin -d lpVirtual`
- 5) Configurem el directori de sortida de l'arxiu `/etc/cups/cups-pdf.conf` a partir d'un editor, per a que deixi tots els documents en format PDF al directori anomenat DocsPDF en el directori d'entrada de l'usuari.

```
#####
#
# 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}/DocsPDF
```

- 6) Fem un restart del cups:
  - `systemctl restart cups`



## 2.2. Path absolut del fitxer

→ -

## 2.3. Permisos del fitxer

→ -

## 2.4. Propietari i grup del fitxer

- Propietari => -
- Grup => -

## 2.5. Preguntes script

- **Es pot enviar a imprimir amb la comanda lp?** Sí, però com que ja he fet també la part 3 de la pràctica, s'ha d'especificar el path absolut de la comanda lp (a la següent part de la pràctica s'explicarà millor el per què):

```
milax@casa:~/Documents/GSX/lab12$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
milax@casa:~/Documents/GSX/lab12$ /usr/bin/lp lab.txt
l'identificador de la petició és lpVirtual-19 (1 fitxer(s))
```

- **Es pot aturar l'enviament cap a la impressora?** Sí, fent un *lpc stop*. Amb això aconseguim que el servei no agafi les dades de l'àrea d'spool i els envii a la impressora.
- **On es queda retingut l'enviament que s'ha fet del fitxer?** Com que a l'aturar, el servei no agafarà les dades de l'àrea d'spool, els documents que l'usuari vulgui imprimir es quedaran retinguts a l'àrea d'spool (/var/spool/cups) fins que el servei es torni a activar (lpc start). Si volem que no es vagin acumulant els documents en aquesta àrea, haurem de bloquejar l'encauament per a que l'usuari no pugui fer peticions, avisant-lo (lpc dissable); d'aquesta manera no arribarà cap petició a l'àrea d'spool.

## 2.6. Joc de proves

- Si tornem a fer l'exemple anterior, veurem que ens notifica de que s'ha realitzat la impressió (indicant-ho amb un identificador) i podem comprovar de que la sortida del document es guarda al directori configurat a l'arxiu /etc/cups/cups-pdf.conf:

```
milax@casa:~/Documents/GSX/lab12$ /usr/bin/lp lab.txt
l'identificador de la petició és lpVirtual-22 (1 fitxer(s))
milax@casa:~/Documents/GSX/lab12$ ls -l /home/milax/DocsPDF/
total 20
-rw----- 1 milax milax 16516 16 de maig 13:20 lab.txt__x_DocsPDF-job_22.pdf
```

- També, podem comprovar totes les peticions que s'han anat realitzant fent un `ls -l` de `/var/spool/cups` (àrea d'spool):

```
milax@casa:~/Documents/GSX/lab12$ sudo ls -l /var/spool/cups
total 84
-rw----- 1 root lp 1086 15 de maig 18:37 c00013
-rw----- 1 root lp 1120 15 de maig 18:44 c00014
-rw----- 1 root lp 1122 15 de maig 18:53 c00015
-rw----- 1 root lp 1122 15 de maig 18:59 c00016
-rw----- 1 root lp 1120 15 de maig 19:04 c00017
-rw----- 1 root lp 1120 15 de maig 19:29 c00018
-rw----- 1 root lp 1120 16 de maig 13:01 c00019
-rw----- 1 root lp 1101 16 de maig 13:14 c00020
-rw----- 1 root lp 1099 16 de maig 13:14 c00021
-rw----- 1 root lp 1120 16 de maig 13:20 c00022
-rw-r----- 1 root lp 716 15 de maig 18:37 d00013-001
-rw-r----- 1 root lp 716 15 de maig 18:43 d00014-001
-rw-r----- 1 root lp 5 15 de maig 18:52 d00015-001
-rw-r----- 1 root lp 5 15 de maig 18:58 d00016-001
-rw-r----- 1 root lp 716 15 de maig 19:04 d00017-001
-rw-r----- 1 root lp 716 15 de maig 19:29 d00018-001
-rw-r----- 1 root lp 716 16 de maig 13:01 d00019-001
-rw-r----- 1 root lp 18 16 de maig 13:14 d00020-001
-rw-r----- 1 root lp 19 16 de maig 13:14 d00021-001
-rw-r----- 1 root lp 716 16 de maig 13:20 d00022-001
drwxrwx---T 2 root lp 4096 15 de maig 19:31 tmp
```

## 3. lp

### 3.1. Codi del script

```
#!/bin/bash

# Comprovem que l'usuari posi parametres
if [[ $# -eq 0 ]] ; then
    echo "No s'ha introduït cap document a imprimir"
    exit 1
fi

# Comprovar paraula clau
function comprovar_clau {
    local paraula_clau="sisplau"
    local error=0
    stty -echo
    read -p "Introdueix la paraula clau: " valor
    stty echo
    echo

    if [ "$valor" = "$paraula_clau" ]; then
        error=0
    else
        error=1
    fi
    return $error
}

if comprovar_clau; then
    echo "Imprimint document ..."
    /usr/bin/lp "$@"
else
    echo "ERROR: La paraula clau no és correcta."
fi
```

### 3.2. Path absolut del fitxer

→ /usr/local/bin/lp

### 3.3. Permisos del fitxer

→ Octal => 755 (-rwxr-xr-x 1 root root 583 15 de maig 17:48 lp)

### 3.4. Propietari i grup del fitxer

- Propietari => root
- Grup => root

### 3.5. Joc de proves

- Primerament, comprovem que al fer un `whereis lp`, podem veure que existeixen dos principalment: el propi del sistema i l'script definit propi realitzat.

```
milax@casa:/usr/local/bin$ whereis lp
lp: /usr/bin/lp /usr/local/bin/lp /usr/share/man/man4/lp.4.gz /usr/share/man/man1/lp.1.gz
```

- El motiu pel que aquest script s'ha ficat en aquest path (`/usr/local/bin/`) és per la raó de suplir l'actual comanda `lp` (per a enviar a imprimir un document). Ho podem mirar millor amb la variable d'entorn `PATH`:

```
milax@casa:/usr/local/bin$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

Com que el path `/usr/local/bin` va abans que el `/usr/bin` (que és on es situa la comanda per defecte del sistema), si fiquem l'script realitzat en el path `/usr/local/bin` abans que l'anterior esmentat, al consultar on es troba `lp`, primer es recorrerà la variable `PATH` començant per la esquerra. Per aquesta raó, primerament, veurà el `lp` situat al `/usr/local/bin` i és el que s'executarà (el nostre script). Si volguéssim executar el `lp` del sistema (fet a l'interior de l'script), hauríem de ficar abans el seu path absolut `/usr/bin/` per tal de que no s'executi el del script `/usr/local/bin`.

- Una vegada vist aquest aspecte, podem fer les proves:
  - Si executem `lp` i fiquem una paraula clau que no és la declarada a l'enunciat ("sisplau"), ens sortirà un error pel termina.

```
milax@casa:~$ lp /home/milax/Documents/GSX/lab12/lab.txt
Introdueix la paraula clau:
ERROR: La paraula clau no és correcta.
```

- Si tornem a executar `lp` i fiquem la paraula clau correcta ("sisplau"), es farà la impressió del document definit al següent path que es passa per paràmetre:

```
milax@casa:~$ lp /home/milax/Documents/GSX/lab12/lab.txt
Introdueix la paraula clau:
Imprimint document ...
l'identificador de la petició és lpVirtual-23 (1 fitxer(s))
milax@casa:~$ ls -l DocsPDF/
total 20
-rw-r--r-- 1 milax milax 16516 16 de maig 14:02 lab.txt__x_DocsPDF-job_23.pdf
milax@casa:~$ sudo ls -l /var/spool/cups
[sudo] contrasenya per a milax:
total 92
-rw-r--r-- 1 root lp 1086 15 de maig 18:37 c00013
-rw-r--r-- 1 root lp 1120 15 de maig 18:44 c00014
-rw-r--r-- 1 root lp 1122 15 de maig 18:53 c00015
-rw-r--r-- 1 root lp 1122 15 de maig 18:59 c00016
-rw-r--r-- 1 root lp 1120 15 de maig 19:04 c00017
-rw-r--r-- 1 root lp 1120 15 de maig 19:29 c00018
-rw-r--r-- 1 root lp 1120 16 de maig 13:01 c00019
-rw-r--r-- 1 root lp 1101 16 de maig 13:14 c00020
-rw-r--r-- 1 root lp 1099 16 de maig 13:14 c00021
-rw-r--r-- 1 root lp 1120 16 de maig 13:20 c00022
-rw-r--r-- 1 root lp 1120 16 de maig 14:02 c00023
-rw-r--r-- 1 root lp 716 15 de maig 18:37 d00013-001
-rw-r--r-- 1 root lp 716 15 de maig 18:43 d00014-001
-rw-r--r-- 1 root lp 5 15 de maig 18:52 d00015-001
-rw-r--r-- 1 root lp 5 15 de maig 18:58 d00016-001
-rw-r--r-- 1 root lp 716 15 de maig 19:04 d00017-001
-rw-r--r-- 1 root lp 716 15 de maig 19:29 d00018-001
-rw-r--r-- 1 root lp 716 16 de maig 13:01 d00019-001
-rw-r--r-- 1 root lp 18 16 de maig 13:14 d00020-001
-rw-r--r-- 1 root lp 19 16 de maig 13:14 d00021-001
-rw-r--r-- 1 root lp 716 16 de maig 13:20 d00022-001
-rw-r--r-- 1 root lp 716 16 de maig 14:02 d00023-001
drwxrwx--T 2 root lp 4096 15 de maig 19:31 tmp
```

- Si ara volem executar la comanda `lp` per defecte directament, podem posar el seu path absolut com ja he esmentat anteriorment, sense executar aquest script:

```
milax@casa:~$ /usr/bin/lp /home/milax/Documents/GSX/lab12/lab.txt
l'identificador de la petició és lpVirtual-24 (1 fitxer(s))
milax@casa:~$ ls -l DocsPDF/
total 40
-rw-r--r-- 1 milax milax 16516 16 de maig 14:02 lab.txt__x_DocsPDF-job_23.pdf
-rw-r--r-- 1 milax milax 16516 16 de maig 14:08 lab.txt__x_DocsPDF-job_24.pdf
```

## 4. consultaComanda.sh

### 4.1. Codi del script

```
#!/bin/bash

PATH="$PATH:/usr/sbin/"

# S'ha de tenir privilegis
if [[ $EUID -ne 0 ]] ; then
    echo "Necessitem privilegis per executar aquest script"
    exit 1
fi

systemctl start acct

if [ $# -eq 2 ]; then
    command="$1"
    user="$2"
    if lastcomm "$command" | awk -v user="$user" '{if ($2 == "S" || $2 == "F" || $2 == "C" || $2 == "D" || $2 == "X") {u=$3} else {u=$2}; if (u == user) print u}' | grep -q "^$user$";
    then
        echo "L'usuari $user ha executat la comanda $command en els següents dies:"
        lastcomm "$command" | awk -v user="$user" '{if ($2 == "S" || $2 == "F" || $2 == "C" || $2 == "D" || $2 == "X") {u=$3; d=$7 " " $8 " " $9 " " $10} else {u=$2; d=$6 " " $7 " " $8 " " $9}; if (u == user) print d}'
    else
        echo "L'usuari $user no ha executat la comanda $command."
    fi
elif [ $# -eq 1 ]; then
    command="$1"
    echo "Els usuaris que han executat la comanda $command i el nombre de vegades que l'han executat són:"
    lastcomm "$command" | awk '{if ($2 == "S" || $2 == "F" || $2 == "C" || $2 == "D" || $2 == "X") {u=$3} else {u=$2}; print u}' | sort | uniq -c
else
    echo "Ús: $0 <comanda> [usuari]"
    echo "Per a veure els dies que un usuari ha executat una comanda: $0 <comanda> <usuari>"
    echo "Per a veure quins usuaris han executat una comanda i el nombre de vegades: $0 <comanda>"
    exit 1
fi
```

### 4.2. Path absolut del fitxer

➔ /home/milax/Documents/GSX/lab12/consultaComada.sh

### 4.3. Permisos del fitxer

➔ Octal => 755 (-rwxr-xr-x 1 root root 1091 16 de maig 15:19 consultaComanda.sh)

### 4.4. Propietari i grup del fitxer

- Propietari => root
- Grup => root

### 4.5. Joc de proves

- Si executem el script on no li passem cap paràmetre, ens mostrarà el següent:

```
milax@casa:~/Documents/GSX/Lab12$ sudo ./consultaComanda.sh
Ús: ./consultaComanda.sh <comanda> [usuari]
Per a veure els dies que un usuari ha executat una comanda: ./consultaComanda.sh <comanda> <usuari>
Per a veure quins usuaris han executat una comanda i el nombre de vegades: ./consultaComanda.sh <comanda>
```

- Si ara li passem una comanda, ens mostrarà els usuaris que l'han executada i el número de vegades:

```
milax@casa:~/Documents/GSX/Lab12$ sudo ./consultaComanda.sh ls
Els usuaris que han executat la comanda ls i el nombre de vegades que l'han executat són:
1 AnnaGS
1 MariaPG
10 milax
9 root
```

- Si fem un `sudo lastcomm | grep ls`, podem veure que els valors coincideixen:

```
milax@casa:~$ sudo lastcomm | grep ls
[sudo] contrasenya per a milax:
ls                root      pts/1    0.00 secs  Fri May 17 01:20
ls                root      pts/1    0.00 secs  Fri May 17 01:20
ls                5        milax    pts/1    0.00 secs  Fri May 17 00:46
ls                milax     pts/1    0.00 secs  Fri May 17 00:46
ls                milax     pts/1    0.00 secs  Fri May 17 00:37
ls                milax     pts/1    0.00 secs  Fri May 17 00:12
ls                MariaPG   pts/0    0.00 secs  Fri May 17 00:06
ls                root      pts/1    0.00 secs  Thu May 16 17:50
ls                root      pts/1    0.00 secs  Thu May 16 17:50
ls                root      pts/1    0.00 secs  Thu May 16 17:50
ls                root      pts/1    0.00 secs  Thu May 16 17:50
ls                milax     pts/1    0.00 secs  Thu May 16 15:25
ls                milax     pts/1    0.00 secs  Thu May 16 15:24
ls                AnnaGS    pts/3    0.00 secs  Thu May 16 14:55
ls                milax     pts/1    0.00 secs  Thu May 16 14:54
ls                5        root      pts/2    0.00 secs  Thu May 16 14:35
ls                milax     pts/1    0.00 secs  Thu May 16 14:35
ls                milax     pts/1    0.00 secs  Thu May 16 14:31
ls                milax     pts/0    0.00 secs  Thu May 16 14:28
```

- Si en aquest cas, li passem com a paràmetres la comanda i un usuari, ens mostrarà els dies en els quals l'ha executada:

```
milax@casa:~/Documents/GSX/Lab12$ sudo ./consultaComanda.sh ls milax
L'usuari milax ha executat la comanda ls en els següents dies:
Fri May 17 00:46
Fri May 17 00:46
Fri May 17 00:37
Fri May 17 00:12
Thu May 16 15:25
Thu May 16 15:24
Thu May 16 14:54
Thu May 16 14:35
Thu May 16 14:31
Thu May 16 14:28
```

Comprovem que també coincideix amb les proves anteriors

\* Si ara, l'usuari milax decideix executar un altre cop la comanda ls i comprovem els dies en els que l'ha executat a partir del script, veurem una nova data a la sortida dels resultats:

```
milax@casa:~/Documents/GSX/Lab12$ ls
consultaComanda.sh  creaSwap.sh  lp  memtest86-4.3.7-usb  prac6_LopezMartinezRuben.tgz
milax@casa:~/Documents/GSX/Lab12$ sudo ./consultaComanda.sh ls milax
L'usuari milax ha executat la comanda ls en els següents dies:
Fri May 17 12:00
Fri May 17 00:46
Fri May 17 00:46
Fri May 17 00:37
Fri May 17 00:12
Thu May 16 15:25
Thu May 16 15:24
Thu May 16 14:54
Thu May 16 14:35
Thu May 16 14:31
Thu May 16 14:28

milax@casa:~/Documents/GSX/Lab12$ sudo ./consultaComanda.sh ls
Els usuaris que han executat la comanda ls i el nombre de vegades que l'han executat són:
1 AnnaGS
1 MariaPG
11 milax
9 root
```

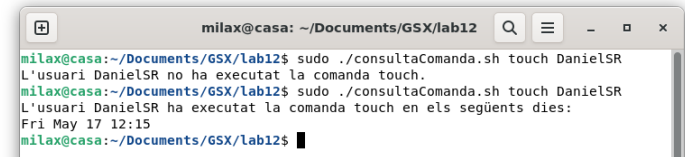
Podem comprovar que el número de cops, fen la comparació de les dues maneres a partir del script, coincideix.

- Si li passem una comanda i un usuari, el qual no ha executat la comanda en qüestió, es mostrarà el següent:

```
milax@casa:~/Documents/GSX/lab12$ sudo ./consultaComanda.sh touch DanielSR
L'usuari DanielSR no ha executat la comanda touch.
```

\* Si ara DanielSR entra al sistema i executa la comanda passada per paràmetre i tornem a fer la prova, veurem el següent:

```
milax@casa:~$ su DanielSR
DanielSR@casa:/home/milax$ cd /empresa/usuarios/DanielSR/bin/
DanielSR@casa:~/bin$ touch hola.sh
DanielSR@casa:~/bin$
```



```
milax@casa:~/Documents/GSX/lab12$ sudo ./consultaComanda.sh touch DanielSR
L'usuari DanielSR no ha executat la comanda touch.
milax@casa:~/Documents/GSX/lab12$ sudo ./consultaComanda.sh touch DanielSR
L'usuari DanielSR ha executat la comanda touch en els següents dies:
Fri May 17 12:15
milax@casa:~/Documents/GSX/lab12$
```

En tots els casos, és molt segur que, per exemple, l'usuari milax hagi executat molts cops la comanda ls (no 10 com indicava inicialment). Això es degut a que el servei acct no estava actiu.

## 5. Monitorització de recursos

Per tal d'estressar el sistema i fer correctament les proves de monitorització dels recursos (memòria, cpu i disc), informant-me i buscant informació, he trobat un paquet que es dedica a explotar els recursos del sistema, creant diferents tasques que requereixen d'una gran capacitat per executar-les segons la quantitat de recursos que volem estressar. Aquest paquet és el stress, desenvolupat originalment per Amos Waterland <apw@debian.org> i, el codi font i les versions més noves estan disponibles a <https://github.com/resurrecting-open-source-projects/stress>.

Estress és una eina utilitzada pels administradors de sistemes per avaluar què tan bé escalaran els nostres sistemes, pels programadors del Kernel per avaluar les característiques de rendiment.

Per tant, abans de començar, he instal·lat aquest paquet:

➤ *sudo apt install stress*

Fem *man* per veure els paràmetres possibles que accepta:

```
stress(1)      tool to impose load on and stress test systems      stress(1)

NAME
  stress - tool to impose load on and stress test a computer system

SYNOPSIS
  stress [OPTIONS]

DESCRIPTION
  stress is a tool that imposes a configurable amount of CPU, memory,
  I/O, or disk stress on a POSIX-compliant operating system and reports
  any errors it detects.

  stress is not a benchmark. It is a tool used by system administrators
  to evaluate how well their systems will scale, by kernel programmers to
  evaluate perceived performance characteristics, and by systems program-
  mers to expose the classes of bugs which only or more frequently mani-
  fest themselves when the system is under heavy load.

OPTIONS
  -?, --help
    Show this help statement.

  --version
    Show version statement.

  -v, --verbose
    Be verbose.

  -q, --quiet
    Be quiet.

  -n, --dry-run
    Show what would have been done.

  -t, --timeout <N>
    Timeout after N seconds. This option is ignored by -n.

  --backoff <N>
    Wait for factor of microseconds before starting work.

  -c, --cpu <N>
    Spawn N workers spinning on sqrt().

  -i, --io <N>
    Spawn N workers spinning on sync().

  -m, --vm <N>
    Spawn N workers spinning on malloc()/free().

  --vm-bytes <B>
    Malloc B bytes per vm worker (default is 256MB).

  --vm-stride <B>
    Touch a byte every B bytes (default is 4096).

  --vm-hang <N>
    Sleep N secs before free (default none, 0 is inf).

  --vm-keep
    Redirty memory instead of freeing and reallocating.

  -d, --hdd <N>
    Spawn N workers spinning on write()/unlink().

  --hdd-bytes <B>
    Write B bytes per hdd worker (default is 1GB). The file will be
    created with mkstemp() in the current directory.

  Note: Numbers may be suffixed with s,m,h,d,y (time) or B,K,M,G (size).

EXAMPLES
  The simple case is that you just want to bring the system load average
  up to an arbitrary value. The following forks 13 processes, each of
  which spins in a tight loop calculating the sqrt() of a random number
  acquired with rand().

  stress -c 13

  Long options are supported, as well as is making the output less ver-
 bose. The following forks 1024 processes, and only reports error mes-
  sages if any.

  stress --quiet --cpu 1k

  To see how your system performs when it is I/O bound, use the -i
  switch. The following forks 4 processes, each of which spins in a tight
  loop calling sync(), which is a system call that flushes memory buffers
  to disk.

  stress -i 4

  Multiple hogs may be combined on the same command line. The following
  does everything the preceding examples did in one command, but also
  turns up the verbosity level as well as showing how to cause the com-
  mand to self-terminate after 1 minute.

  stress -c 13 -i 4 --verbose --timeout 1m

  You can write a file of arbitrary length to disk. The file is created
  with mkstemp() in the current directory.

  stress -d 1 --hdd-bytes 13

  Large file support is enabled.

  stress -d 1 --hdd-bytes 3G

AUTHOR
  stress was originally developed by Amos Waterland <apw@debian.org> and
```

\* Com que tinc molts recursos assignats a la imatge de la màquina virtual per a que funcioni correctament, la prova no ressaltarà molt en quant a l'activitat d'estressar la gran majoria de recursos, però si m'ha servit per veure que variables es modifiquen a cada recursos, el per què d'aquest comportament i entendre que significa cada una d'aquestes variable.



Una vegada tenim els paquets instal·lats, realitzarem l'estudi amb les següents característiques per a vmstat i iostat:

- **stress --cpu 4 --io 2 --vm 1 --vm-bytes 512M --timeout 60s**
  - --cpu 4 -> 4 nuclis estaran sotmesos a estrès.
  - --io 2 -> indica el nivell d'estrès d'entrada/sortida (en aquest cas 2 processos d'I/O).
  - --vm 1 -> s'utilitzarà un procés de memòria virtual en execució.
  - --vm-bytes 512M -> s'utilitzarà 512 megabytes per a cada procés de memòria virtual.
  - --timeout 60s -> l'estrès realitzat al sistema s'acabarà en 60 segons.

## 5.1. Vmstat

```
milax@casa:~$ vmstat 1
procs-----memory-----swap-----io-----system-----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
1 0 0 4343708 63748 918268 0 0 0 26 11 131 146 2 0 98 0 0
1 0 0 4343708 63748 918268 0 0 0 0 0 541 233 0 0 100 0 0
1 0 0 4343708 63748 918268 0 0 0 0 0 537 343 0 0 100 0 0
1 0 0 4343708 63748 918268 0 0 0 0 0 495 214 0 0 100 0 0
1 0 0 4343708 63748 918268 0 0 0 0 0 582 329 0 0 100 0 0
1 0 0 4343708 63748 918268 0 0 0 48 1331 4377 3 1 97 0 0
7 1 0 4343708 63748 918268 0 0 0 40 1079 658 16 5 73 6 0
6 2 0 4343708 63748 918268 0 0 0 0 4071 5228 51 13 12 23 0
7 1 0 4343708 63748 918268 0 0 0 0 5363 7696 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5105 7232 51 13 12 23 0
7 1 0 4343708 63748 918268 0 0 0 0 5261 7376 51 14 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5185 7135 51 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5275 7445 52 13 12 23 0
7 2 0 4343708 63748 918268 0 0 0 0 5451 7831 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5343 7906 52 13 11 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5354 7461 51 13 12 23 0
7 1 0 4343708 63748 918268 0 0 0 0 4628 6142 51 13 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5211 7308 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5253 7438 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5211 7371 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5285 7639 52 13 12 23 0
8 2 0 4343708 63748 918268 0 0 0 416 5667 8534 53 14 11 22 0
6 2 0 4343708 63748 918268 0 0 0 24 5811 8709 53 14 11 22 0
6 2 0 4343708 63748 918268 0 0 0 0 5337 7729 52 13 12 23 0
7 1 0 4343708 63748 918268 0 0 0 0 5259 7389 52 12 12 23 0
9 1 0 4343708 63748 918268 0 0 132 80 5305 7470 52 13 12 23 0
7 1 0 4343708 63748 918268 0 0 0 956 5448 7398 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5464 7839 52 12 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5367 7658 52 12 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5239 7364 52 13 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5285 7457 51 13 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5167 7084 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5106 7179 52 13 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5089 7206 52 12 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5334 7368 52 13 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5207 7326 52 12 13 23 0
7 1 0 4343708 63748 918268 0 0 0 0 5051 7026 52 12 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5252 7293 52 12 13 23 0
6 2 0 4343708 63748 918268 0 0 0 0 5548 8142 52 13 12 23 0
9 2 0 4343708 63748 918268 0 0 0 640 5809 8441 53 14 11 22 0
9 1 0 4343708 63748 918268 0 0 16 1896 10554 16712 66 19 6 9 0
9 2 0 4343708 63748 918268 0 0 284 3228 19325 22809 66 23 5 6 0
7 1 0 4343708 63748 918268 0 0 28 600 7577 11819 57 17 18 9 0
6 2 0 4343708 63748 918268 0 0 0 100 5534 9032 53 13 15 19 0
6 2 0 4343708 63748 918268 0 0 0 24 5457 8550 52 13 12 23 0
6 2 0 4343708 63748 918268 0 0 0 412 5561 8835 52 14 12 23 0
7 2 0 4343708 63748 918268 0 0 0 440 6471 11268 53 14 11 21 0
6 2 0 4343708 63748 918268 0 0 24 376 6465 11519 54 14 11 21 0
8 2 0 4343708 63748 918268 0 0 2224 10951 22803 67 18 6 9 0
procs-----memory-----swap-----io-----system-----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
8 2 0 4343708 63748 918268 0 0 0 0 0 9004 12525 57 15 9 20 0
6 2 0 4343708 63748 918268 0 0 0 192 6999 8375 54 13 11 22 0
```

### 5.1.1. Estudi

#### 1) Primers moments (Inici):

En aquest punt, el sistema està inactiu, amb la CPU principalment lliure.

La columna us (temps de CPU gastat en mode d'usuari) i sy (temps de CPU gastat en mode de sistema) són baixos, la qual cosa indica que no hi ha una càrrega significativa en la CPU.

La columna id (temps de CPU inactiu) és alta, la qual cosa confirma que la CPU no està sent utilitzada.

No hi ha activitat d'intercanvi (swap), ja que tant si com so són 0. Això indica que el sistema no està utilitzant memòria virtual en aquest moment.

### 2) Fluctuacions en l'activitat de la CPU i la E/S:

A mesura que avancen els resultats, s'observen canvis en l'activitat de la CPU i la E/S. Algunes files mostren un augment en el nombre de processos en execució (r augmenta), la qual cosa indica una major càrrega en la CPU.

En altres files, veiem un augment en l'activitat de E/S (bi i bo augmenten), la qual cosa suggereix que s'estan realitzant operacions intensives de lectura i escriptura en els dispositius d'emmagatzematge.

Aquests canvis son causats per l'execució de programes o processos que consumeixen grans recursos, com ara la realització de tasques de processament intensiu o la lectura/escriptura de dades al disc.

### 3) Canvis en l'activitat de la CPU i la memòria:

Les fluctuacions en l'activitat de la CPU (us, sy) i la E/S (bi, bo) indiquen moments de major càrrega en el sistema.

Per exemple, un augment en us i sy pot suggerir que el sistema està executant programes que consumeixen CPU, com ara càlculs intensius o processos de sistema.

Els canvis en la utilització de la memòria (swpd, free, buff, cache) poden reflectir el comportament dels programes en execució i la gestió de la memòria per part del sistema operatiu.

Per exemple, un augment en swpd pot indicar que el sistema està utilitzant més espai d'intercanvi a causa de la manca de memòria física disponible.

## **5.1.2. Conclusions resultats obtinguts i camps que varien en funció dels recursos explotats**

### CPU:

- El temps de CPU gastat en mode d'usuari (us) i en mode de sistema (sy) augmenta quan s'estressa la CPU.
- El temps de CPU inactiu (id) disminueix, ja que la CPU estarà més ocupada.

### Memòria:

- El valor de swpd (ús de l'espai de swap) augmentarà si hi ha una càrrega important en la memòria i es comença a utilitzar l'espai de swap (que en aquest cas no s'arriba a utilitzar).
- El valor de free (memòria física lliure) disminueix si la memòria es va omplint i s'utilitza l'espai de swap.
- Els valors de buff i cache poden augmentar o disminuir depenent de com el sistema gestiona els buffers i la memòria cau.

### E/S:

- Els valors de bi (blocs d'entrada) i bo (blocs de sortida) augmentaran si es realitzen operacions intensives de lectura i escriptura a disc.

## 5.2. iostat

```
milax@casa:~$ iostat 1
Linux 6.1.0-18-amd64 (casa)      16/5/24      _x86_64_      (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1,90    0,01   0,40   0,19    0,00   97,50

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                24,76      122,29      88,89        0,00      1832311    750396      0
sdb                 0,02       0,41        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           3,43    0,00   0,89   0,00    0,00   95,69

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                 0,00        0,00        0,00        0,00         0         0         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1,39    0,00   0,38   0,00    0,00   98,23

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                 0,00        0,00        0,00        0,00         0         0         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,38    0,00   0,50   0,00    0,00   99,12

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                 0,00        0,00        0,00        0,00         0         0         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           28,55    0,00  13,09   6,73    0,00   51,64

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                295,00        0,00      564,00        0,00         0         0         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           52,17    0,51  18,45   9,09    0,00  21,17

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                2217,00        0,00     2880,00        0,00         0        2880         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           51,79    0,13  16,07   9,57    0,00  22,45

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                2375,00        0,00     3532,00        0,00         0        3532         0
sdb                 0,00        0,00        0,00        0,00         0         0         0

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           52,10    0,00  14,78  10,06    0,00  23,06

Device:            tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
sda                1673,00        0,00      24,00        0,00         0         24         0
sdb                 0,00        0,00        0,00        0,00         0         0         0
```

### 5.2.1. Estudi

#### 1) Moments inicials:

En quant a la CPU, el percentatge de temps gastat en mode d'usuari (%user) i en mode de sistema (%system) és baix, la qual cosa indica que la CPU no està sent utilitzada intensivament per processos d'usuari ni per processos del sistema. El percentatge de temps en què la CPU està inactiva (%idle) és alt, confirmant que la CPU no està sent utilitzada. No hi ha esperes significatives d'E/S (%iowait). El percentatge de temps en què la CPU està sent utilitzada per processos de baixa prioritat (%nice) també és baix. Els dispositius d'E/S (sda i sdb), pel que fa el nombre d'operacions d'E/S per segon (tps) és nul, indicant una activitat mínima de lectura/escriptura en els dispositius. Els kilobytes llegits per segon (kB\_read/s), els kilobytes escrits per segon (kB\_wrtn/s) i els kilobytes descartats per segon (kB\_dscd/s) també són baixos.

#### 2) Fluctuacions en l'activitat de la CPU i la E/S:

A la CPU, a mesura que avança el temps, s'observa un augment gradual en el percentatge de temps gastat en mode d'usuari (%user) i en mode de sistema (%system). Això indica una major càrrega a la CPU a causa de l'execució de programes intensius en CPU o processos del sistema. El percentatge de temps d'espera d'E/S (%iowait) es manté baix tot i que va creixen gradualment. El percentatge de temps d'inactivitat de la CPU (%idle) disminueix, ja que la CPU està més ocupada.

En els dispositius d'E/S (sda i sdb), s'observa un augment en el nombre d'operacions d'E/S per segon (tps) en alguns punts, la qual cosa indica una major activitat de lectura/escriptura en els dispositius d'emmagatzematge. Els kilobytes llegits i escrits per segon també augmenten en alguns casos, indicant una major activitat d'E/S.

### 3) Canvis en l'activitat de la CPU:

A la CPU, els augments en els percentatges de temps gastat en mode d'usuari (%user) i en mode de sistema (%system) poden indicar una càrrega intensiva a la CPU a causa de l'execució de programes o processos que consumeixen recursos. El percentatge de temps d'espera d'E/S (%iowait) creix generalment, la qual cosa indica que el sistema està experimentant esperes d'E/S. El percentatge de temps d'inactivitat de la CPU (%idle) disminueix encara més a mesura que augmenta l'activitat de la CPU i les operacions d'E/S.

## **5.2.2. Conclusions resultats obtinguts i camps que varien en funció dels recursos explotats**

### CPU:

- El percentatge de temps gastat en mode d'usuari (%user) i en mode de sistema (%system) tendeix a augmentar amb la càrrega del sistema, indicant una major activitat de la CPU.
- El percentatge de temps d'espera d'E/S (%iowait) creix però no bruscament, la qual cosa pot indicar que l'activitat observada està més relacionada amb la CPU que amb les operacions d'E/S.

### E/S:

- El nombre d'operacions d'E/S per segon (tps) tendeix a augmentar en moments de major activitat del sistema, la qual cosa indica una major activitat de lectura/escriptura en els dispositius d'emmagatzematge.
- Els kilobytes llegits i escrits per segon també poden augmentar en aquests moments de major activitat d'E/S.