

Lab Gestió del Sistema de Fitxers:

Nom:	swap_memory file_system	Grau d'assoliment	Alt
Descripció			
S'estableix un sistema de fitxers en la partició destinada a la regió de swap , on se li assigna un 10% d'espai reservat per a root. A continuació s'hi assignen tot l'arbre de directoris per baix de /home , així com /usr/local i de /var/local . Finalment s'estableixen els canvis realitzats permanents entre els diferents boots del sistema.			
Procediment			
S'identifica la partició de swap (swapon -s), que en el cas de la màquina virtual amb que s'ha treballat correspon a /sda5 .			
Es trasllada l'àrea de swap al directori arrel localitzat en la partició principal. Per això copiem un arxiu de zeros, tot formatant-lo a la mida necessària de la regió de swap (16 MB). A continuació desactiven l'antiga regió de swap de la partició específica per aquest (swapoff /sda5) i activem la nova assignant-la al path de l'arxiu buit al directori arrel (swapon /swap).			
Es crea el sistema de fitxers emprant la comanda mkfs -t , se li assigna el tipus de sistema de fitxers ext2 , i s'indica el disc (partició), /sda5 . Enlloc de ext2 també pot ser vfat .			
Per establir l'espai reservat per a root s'afegeix l'argument -m a la comanda tune2fs , tot indicant seguit el percentatge de memòria reservada i en quin disc o partició.			
<pre>tune2fs -m 10 /dev/sda5</pre>			
Per assignar tot l'arbre de directoris per sota de /home deixant solament l'usuari root copiem tot el contingut de /home mitjançant la comanda tar . Copiar un arbre de directoris i els seus continguts a un altre sistema de fitxers mitjançant tar preserva la propietat, els permisos i les marques de temps. Es permet realitzar una còpia recursiva sense crear un fitxer tar intermedi. El destí on es copia es el directori /mnt on s'ha montat el sistema d'arxius de la partició sda5 .			
<pre>cd /home tar cf - * (cd /mnt; tar xfp -)</pre>			
Es realitza el mateix tipus de copia per establir tots els directoris que estan per sota de /usr/local i de /var/local sota /mnt .			
<pre>cd /usr/local tar cf - * (cd /mnt; tar xfp -)</pre>			
<pre>cd /var/local</pre>			

```
tar cf - * | ( cd /mnt; tar xfp -)
```

En aquesta partició també **crearem un fitxer de swap**, seguint el mateix procés que en el trasllat de l'àrea de swap.

Creem **àrea de treball temporal** anomenada **tmpusr**, que es crearà cada cop que l'usuari entri, mitjançant l'adició de les comandes **mkdir** i **mount** al fitxer `~/profile`

Es crea el directori on es montara l'àrea temporal.

```
mkdir ~/tmpusr
```

Es monta al path del directori creat (`~/tmpusr`) emprant la comanda **mount**, formatant-lo com TMP File System **tmpfs**, i sense assignar-li la mida limit.

```
mount -t tmpfs -o tmpfs ~/tmpusr
```

El contingut es destrueix de cada cop que es realitza un reboot, ja que es tracta d'un sistema de fitxers temporals.

Per fer que tots els canvis siguin permanents entre els diferents boots del sistema s'edita la configuració de `/etc/tmpusr`. Les particions que figuren en **fstab** es poden configurar per muntar automàticament durant el procés d'arrencada seguint la següent sintaxis:

```
# <file system> <mount point> <type> <options> <dump> <pass>
```

Les entrades que s'afegeixen per cada partició configurada son les següents:

```
/dev/sda5      ~/tmpusr      tmpfs  tmpfs    0      0
/dev/sda5      /mnt           ext2   default  0      0
```

Joc de Proves

Per anar realitzant les comprovacions necessaries alhora de crear i formatar els sistemes de fitxers s'empren les comandes que llisten informació sobre discs i particions.

fdisk -l Es llista la informació de les particions del sistema.

df Es llista tota la informació dels sistemes de fitxers.

dumpe2fs /dev/sda5 Es mostra tota la informació d'un disc.

En el cas de la copia de l'arbre de directoris, es comprova que en el directori destí es trobi el contingut, mentre que en el de la reserva d'espai per root s'observa com la sortida de la comanda indica: Setting reserved blocks percentage to 10% (52352 blocks)

Nom:	NIS i NFS	Grau d'assoliment	Alt
Descripció			
En aquesta pràctica, configurem un servidor NFS i un NIS. Es donaran d'altra 3 usuaris remots de forma que puguin entrar en qualsevol màquina i que tinguin el directori d'entrada allotjat en el servidor NFS.			

Decisions de Disseny

Bàsicament hem seguit els passos especificats en la pàgina web d'oracle: (https://docs.oracle.com/cd/E52668_01/E54669/html/ol7-s15-auth.html).

A l'hora de crear els usuaris, hem decidit anomenar-los user1, user2 i user3, amb contrasenya el mateix nom de l'usuari.

Procediment

Primer configurem el servidor:

Instal·lem els paquets necessaris per crear el servidor NFS i NIS

```
apt-get install nfs-common nfs-kernel-server nis portmap
```

en els laboratoris, usem com a nom del domini l'indicat i l'anotem (p.e. d119)

Creem la carpeta on afegirem els directoris a compartir

```
mkdir /home/remots
```

Escribim en el /etc/export el directori que compartirem, sobre quina xarxa i amb quins permisos, de forma que quedarà de la següent forma:

```
/home/remots *(rw, sync)
```

Creem la taula NFS amb els directoris que exportem

```
exportfs -a
```

Engueguem el servei de NFS

```
/etc/init.d/nfs-kernel-server restart
```

-----FINS AQUÍ HEM CONFIGURAT EL NFS, ARA CONFIGUREM EL NIS-----

Creem els Usuaris per al servidor NIS

```
useradd -m -d /home/remots/user1 -b /home/remots -s /bin/bash user1
```

```
useradd -m -d /home/remots/user2 -b /home/remots -s /bin/bash user2
```

```
useradd -m -d /home/remots/user3 -b /home/remots -s /bin/bash user3
```

```
passwd user1
```

```
passwd user2
```

```
passwd user3
```

En el /etc/ypserv.securenets, indiquem la xarxa a la que donem servei:

```
MASCARA_XARXA SERVER_IP
```

En el fitxer /etc/default/nis modifiquem els següents camps:

```
NISSERVER=true
```

```
NISCLIENT=false
```

```
MASTER=SERVER_IP
```

En el /etc/hosts, afegim l'adreça IP del client:

```
afegir el client
```

Indiquem en el fitxer /etc/yp.conf la IP on es troba el servidor (la pròpia):

```
ypserver SERVER_IP
```

En el fitxer de configuració /var/yp/Makefile hem de indicar el valor mínim i màxim que identificaràn els usuaris que compartirem (Min UID: 1001)

Iniciem els diferents serveis:

```
systemctl start ypserv
```

```
systemctl start yppasswdd
```

```
systemctl restart rpcbind
```

```
systemctl restart nis
```

Arranquem el servidor NIS (no hem de afegir cap altre host)

```
/usr/lib/yp/ypinit -m
```

Update the NIS maps:

```
make -C /var/yp
```

Ara configurem el client:

Instal·lem els paquets necessaris

```
apt-get install nfs-common nis portmap
```

com a nom de domini usarem el que hem indicat en el servidor.

Creem el directori /home/remot

```
mkdir /home/remot/
```

Montem el sistema de fitxers ofert pel NFS

```
mount -t nfs NFS_SERVER_IP:/home/remots /home/remot
```

Modifiquem el fitxer /etc/yp.conf:

```

ypserver SERVER_IP
Modifiquem el fitxer /etc/nsswitch.conf:
passwd: files compat nis
shadow: files compat nis
group: files compat nis
netgroup: nis
hosts: files dns nis
Afehim al final dels fitxers:
echo "+::::::" >> /etc/passwd
echo "+:::" >> /etc/group
echo "+::::::" >> /etc/shadow
Reiniciem els serveis
systemctl restart ypbind
systemctl restart nis

```

Joc de Proves

Hem comprovat que el NFS funciona correctament creant un fitxer anomenat prova.txt en la carpeta exportada pel servidor i muntada en el client i després llegint-lo en la mateixa carpeta en el servidor.

En el client hem usat les següents comandes ypwhich i rpcinfo per comprovar que el servidor NIS està correctament configurat.

A l'hora de loguejar-nos en la màquina client (en els laboratoris) usant un dels usuaris exportats pel servidor NIS, ens detecta un error en el servei yppasswdd. Hem buscat com solucionar-lo però cap de les propostes trobades ens ha funcionat.

Després, a casa hem repetit els mateixos passos per dur a terme la configuració del servidor NIS i ens ha funcionat sense cap problema, com es pot observar en la imatge, ens hem pogut loggejar com a "user1" des del client, creiem que la causa és un inici amb un ordre no correcte dels diferents serveis que hi participen:

```

root@client:/home/client/GSX_Labs/Prac1Xarxes# ypwhich
ns.interna
root@client:/home/client/GSX_Labs/Prac1Xarxes# rpcinfo
  program version netid      address                service  owner
  100000    4      tcp6      :::.0.111              portmapper superuser
  100000    3      tcp6      :::.0.111              portmapper superuser
  100000    4      udp6      :::.0.111              portmapper superuser
  100000    3      udp6      :::.0.111              portmapper superuser
  100000    4      tcp      0.0.0.0.0.111          portmapper superuser
  100000    3      tcp      0.0.0.0.0.111          portmapper superuser
  100000    2      tcp      0.0.0.0.0.111          portmapper superuser
  100000    4      udp      0.0.0.0.0.111          portmapper superuser
  100000    3      udp      0.0.0.0.0.111          portmapper superuser
  100000    2      udp      0.0.0.0.0.111          portmapper superuser
  100000    4      local    /run/rpcbind.sock      portmapper superuser
  100000    3      local    /run/rpcbind.sock      portmapper superuser
  100007    2      udp      0.0.0.0.2.185          ypbind   superuser
  100007    1      udp      0.0.0.0.2.185          ypbind   superuser
  100007    2      tcp      0.0.0.0.2.186          ypbind   superuser
  100007    1      tcp      0.0.0.0.2.186          ypbind   superuser

```

```

Debian GNU/Linux 9 client tty1

Hint: Num Lock on

client login: user1
Password:
Linux client 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No directory, logging in with HOME=/
user1@client:/$ _

```

Lab Dispositius:

Nom:	Virtual Disk tmpfs	Grau d'assoliment	Alt
Permisos:	Root	Propietari:	-
Descripció			
Es requereix crear un disc virtual a memòria usant el format tmpfs de mida 100 MB, tot comprovant si s'ha creat correctament revisant els discs del sistema.			
Decisions de Disseny			
Es segueix el procediment per crear un sistema de fitxers en format tmpfs , tal i com es realitza en l'anterior apartat, indicat en documentat en la guia d'Oracle i altres fonts referenciades a continuació:			

<https://docs.oracle.com/cd/E19683-01/817-3814/6mjcp0r0l/index.html>
<https://www.jamescoyle.net/how-to/943-create-a-ram-disk-in-linux>

També es podria formatar com **ramfs**. Com que es tracta d'un sistema d'arxiu temporal, el contingut es borrarà cada cop que es reinici el sistema.

tmpfs apareix com un sistema d'arxius muntat encara que utilitza memòria volàtil, similar als discs RAM, que apareixen com discs virtuals.

Procediment

Es crea el directori on es montara el disc virtual (assumint permisos de root ja que es un directori protegit).

```
mkdir /mnt/mem
```

Es monta al path del directori creat (/mnt/mem) emprant la comanda **mount**, formatant-lo com TMP File System **tmpfs**, i assignant-li la mida limit requerida.

```
mount -t tmpfs -o size=100 tmpfs /mnt/mem
```

Per corroborar la creació del **tmpfs** creat, s'utilitza la comanda **df**, la qual reporta els sistemes d'arxius. Per llistar els discs físics i/o les seves particions s'utilitzarien comandes com **fdisk -l** o **parted -l**.

Joc de Proves

La comanda **df** reporta en la última entrada la creació del **tmpfs**.

```
tmpfs                4          0          4    0% /mnt/mem
```

Nom:	cups	Grau d'assoliment	Alt
Descripció			
<p>Es configura el sistema d'impressió CUPS tot generant una impressora virtual que genera fitxers pdf, la qual s'ha configurat de manera que el directori, per defecte, on es guardin els documents generats sigui a partir d'un directori anomenat DocsPDF sota un directori de l'usuari que imprimeix.</p>			
Decisions de Disseny			
<p>Es decideix implementar la configuració amb comandes de la shell (o editant fitxers de cofiguració), no amb altres configuracions manuals, en particular alhora d'afegir la impresora virtual amb la comanda lpadmin. Es segueix el procediment indicat en documentat en la guia d'Oracle envers el sistema CUPS i altres fonts referenciades a continuació.</p> <p>https://docs.oracle.com/cd/E23824_01/html/821-1451/gllgm.html</p>			

<http://blackhold.nusepas.com/2010/04/02/impresora-virtual-pdf-en-cups/>
<https://enavas.blogspot.com.es/2007/11/cups-pdf-una-impresora-virtual-para.html>

Procediment

S'instalen (`apt-get install`) els packages `cups` i `cups-pdf` la qual proporcionen el sistema d'impressió CUPS i la funcionalitat de produir fitxers PDF mitjançant el sistema d'impressió respectivament.

Per establir les configuracions s'edita el fitxer `cups-pdf.conf` localitzat al path `/etc/cups`. En aquest es canvien les variables del Default path per establir el directori per defecte (**DocsPDF**) de la sortida dels pdf per sota de `/mnt/mem`. Per que sigui específic per l'usuari actual es defineix el path en funció de `${USER}`. Es crea el directori prèviament amb `mkdir`.

```
#####  
##  
#  
# Path Settings  
#  
#####  
##  
  
### Key: Out  
## CUPS-PDF output directory  
## special qualifiers:  
##     ${HOME} will be expanded to the user's home directory  
##     ${USER} will be expanded to the user name  
## in case it is an NFS export make sure it is exported without  
## root_squash!  
## on Ubuntu, the cupsd AppArmor profile needs to be updated  
## to match the output path as per instructions in LP:147551  
### Default: /var/spool/cups-pdf/${USER}
```

Out `/mnt/mem/${USER}/DocsPDF`

S'utilitza la comanda `lpadmin` amb l'opció `-p` per afegir una impressora a CUPS, tot assignant el nom i que es tracta d'una impressora virtual enllaçada a `cups-pdf`:

```
/usr/sbin/lpadmin -p virtualImpre -E -v cups-pdf:/Generic/PostScript
```

Finalment s'habilita l'impressora:

```
cupsaccept virtualImpre  
cupsenable virtualImpre
```

Joc de Proves




Es comprova si l'impressora s'ha habilitat amb la comanda següent:

```
lpstat -p printer-name -l
```

El qual retorna que si ha estat habilitada.

```
printer virtualImpre is idle.  enabled since Tue 08 May 2018 07:19:32 AM CEST
```

També s'observa com apareix al llistat d'impresores alhora de triar-ne una per fer l'impressió:

	Printer	Location	Status
	Print to File		
	PDF		
	virtualImpre		

Finalment es comprova si es generen els fitxers PDF en la carpeta de destí realitzant la impressió virtual.

```
root@debian-gnu-linux-8:/mnt/mem/parallels/DocsPDF# ls
DocsPDF.pdf
-
```


Nom:	lp, lp_config.sh	Grau d'assoliment	Alt
Permisos:	--X--X--X	Propietari:	root
Ubicació dels fitxers:	GSX/Prac2Sistemes	Grup:	root
Descripció			
<p>L'script lp_config.sh s'encarrega d'instalar l'script lp que extindrà la funcionalitat de la comanda lp. Aquest script posa a punt tot el necessari i permet escollir la contrassenya mestra del fitxer de claus (per a desenscriptar-lo) i també permet registrar un nombre indeterminat d'usuaris per a que utilitzin la impressora virtual.</p> <p>Aquesta funcionalitat extesa consisteix en que cada cop que la comanda lp sigui cridada amb la cadena "virtualImpre" com a argument, serà el nostre propi script qui sigui executat aquesta crida i demanarà a l'usuari el seu password per a utilitzar el dispositiu virtualImpre. La verificació del password es duu a terme utilitzant el fitxer /usr/local/secret, que actua com una base de dades que relaciona usuaris amb passwords. Sense un password correcte no es pot utilitzar l'impressora.</p>			
Decisions de Disseny			
<p>lp_config:</p> <p>Per a dur a terme aquest efecte d'intercepció el que hem fet ha estat instal·lar l'script lp a /usr/local/lp i afegir a la variable PATH aquest nou camí. Concretament, s'afegeix al principi de la variable PATH per a que sigui el primer lloc on es busquin els executables, de tal manera que si volem utilitzar la comanda lp, serà el nostre script lp qui sigui trobat primer, i en conseqüència, el que s'executa.</p> <p>Per a fer permanent la modificació de la variable PATH cal manipular algun fitxer de configuració. Jo he decidit fer servir el fitxer /etc/bash.bashrc per a tal fi. Aquest fitxer es caracteritza per executar-se cada cop que s'obre un terminal interactiu sense inici de sessió, pel que resulta adequat pel nostre objectiu. Cada cop que s'obre un terminal, la variable PATH es modifica. He escrit la instrucció que modifica el PATH al final del fitxer .bashrc.</p> <p>Per últim copiem l'script lp a la carpeta /usr/local/lp.</p> <p>Es duen a terme aquelles comprovacions d'existència, contingut, etc. necessàries per a donar una cobertura mínima front als errors.</p> <p>lp:</p> <p>Es miren tots els arguments que ha rebut la comanda i es busca "virtualImpre". No hem aconseguit cercar amb grep "-d virtualImpre" degut a la interpretació que fa la comanda del guió com a argument. Per tant, sempre que aparegui el terme virtualImpre al cridar a la comanda lp ens demanarà usuari i contrassenya.</p> <p>Si no es troba aquesta cadena, se li passen els mateixos arguments al programa lp del sistema, fent que l'usuari no s'adoni que ha executat el nostre codi.</p>			

Si pel contrari, aquesta cadena es troba; es consulta el password de l'usuari que fa la petició al fitxer /usr/local/secret i segons si el password es correcte o incorrecte es permet executar la verdadera comanda lp amb els arguments rebuts.

El fitxer de passwords es troba encriptat i dins de l'script lp s'hi pot llegir en clar la contrassenya mestra del fitxer de claus, és per això que aquest script només tindrà permisos d'execució (222). Es podria haver contemplat l'ús d'una signatura, però fent-ho d'aquesta manera l'administrador del sistema pot trobar fàcilment la contrassenya utilitzant el superusuari.

Codi

```
{lp_config.sh}

#!/bin/bash

PASSWDFILE=/usr/local/secret
LP_PATH=/usr/local/lp

#Mirem parametre d'ajuda
if [ "$1" = "-h" ]; then
    ajuda
    exit 0
fi

# Comprovem existencia de directoris
# considerem que el directori /usr/local ja existeixen
if [ ! -f "/usr/local/secret" ]; then
    echo -e "\nError, file /usr/local/secret does not exist. Creating..."
    touch /usr/local/secret
fi

# Comprovem existencia de directoris
if [ ! -d "/usr/local/lp" ]; then
    echo -e "\nError, directory /usr/local/lp does not exist. Creating..."
    mkdir /usr/local/lp
fi

# Modifiquem la variable PATH si cal. Ara buscare la comanda lp en /usr/local/lp en primer lloc
if [ $(echo $PATH | cut -d ':' -f1) != $LP_PATH ]
then
    export PATH="$LP_PATH:$PATH"
    echo -e "\nexport PATH=$PATH" >> /etc/bash.bashrc
fi

# copiem l'script lp a LP_PATH
cp lp $LP_PATH

echo "Introdueix la clau mestra del fitxer de contrassenyes"
read PSW
if [ -z $PSW ]
then
    echo "has deixat el password buit. S'assignara automaticament a \"gsx2018\""
    PSW="gsx2018"
fi

newusers=""

echo "Introdueix ara els usuaris i passwords. Deixals buits per a parar"
while :
```

```

do
    echo "Introdueix nom d'usuari: "
    read user
    echo "Introdueix password: "
    read pass
    if [ -z $user ] || [ -z $pass ]; then
        break
    fi
    newusers=$(echo "$newusers
$user;$pass")
done

echo -e "$newusers" >> secret.decrypt
more secret.decrypt | openssl enc -aes-128-cbc -a -salt -pass pass:$PSW > /usr/local/secret
rm secret.decrypt

# Afegim el password al script per a pugui descriptar
sed -i "s/-pass pass:/-pass pass:$PSW/g" /usr/local/lp/lp

{lp}

#!/bin/bash

UBICACIO=$(whereis lp | cut -d ' ' -f2)/lp

if [ "$(echo -e "$*" | grep -e "virtualImpre")" == "" ]
then
    # si no troba aquestes opcions
    $UBICACIO $* #li passem la crida a la comanda real
else
    # si troba les opcions
    # Comprovem existencia de directoris
    # considerem que el directori /usr/local ja existeixen
    if [ ! -f "/usr/local/secret" ]
    then
        # el fitxer de credencials no existeix
        echo -e "\nError, file /usr/local/secret does not exist"
        exit 1
    else
        # el fitxer de credencials existeix
        echo -e "Introdueix el teu usuari"
        read USER
        CREDENTIALS=$(more /usr/local/secret | openssl enc -aes-128-cbc -a -d -salt
-pass pass: | grep -E "$USER")
        if [ -z "$CREDENTIALS" ]
        then
            # l'usuari actual no te credencials per utilitzar limpresora
            echo -e "\nError, you are not registered, so you cannot use
virtualImpre\nAdd your credentials into /usr/local/secret and try again later"
            exit 2
        else
            # l'usuari te credencials
            echo -e "Introdueix la teva contrassenya"
            stty -echo
            # apaguem el echo
            stty echo
            # l'encenem
            REAL_PASSWORD=$(echo $CREDENTIALS | cut -d ';' -f2)
            if [ "$REAL_PASSWORD" == "$PASSWORD" ]
            then
                # si el pass coincideix llavors cridem a la comanda
                $UBICACIO $* #li passem la crida a la comanda real
            else
                # si no coincideix
                echo -e "\nEl password no es correcte"
            fi
        fi
    fi
fi

```

Joc de Proves

Podem provar que s'ha afegit la línia que fa permanent la modificació del PATH amb la següent comanda:

```
more /etc/bash.bashrc | grep /usr/local/lp
```

o comprovar que la variable PATH ha estat modificada

```
echo $PATH
```

Comprovem que el programa lp està a la seva ubicació:

```
ls /usr/local/lp
```

Podem comprovar el contingut de la base de dades amb la següent comanda

```
more /usr/local/secret | openssl enc -aes-128-cbc -a -d -salt -pass  
pass:EL_TEU_PASSWORD
```

S'ha afegit un petit snippet que et permet afegir usuaris a la base de dades en el moment d'instalar-la, per tal de fer un petit joc de proves i provar amb diferents usuaris.

Hem dut a terme proves amb diferents usuaris i en tots la verificació del password es adequat.

Lab Accounting i Monitorització:

Nom:	infomem_config.sh infomem.sh	Grau d'assoliment	Alt
Permisos:	rwxr-xr-x	Propietari:	Creador de l'script
Ubicació dels fitxers:	Prac2Sistemes	Grup:	Creador de l'script
Descripció			
<p>Es tracta d'una parella d'scripts que posen a punt una funcionalitat afegida d'avisar a l'usuari actual en el moment de que inicia sessió de la quantitat d'espai en disc que està gastant el seu directori \$HOME.</p> <p>S'utilitzen dos scripts, <code>infomem_config.sh</code> que duu a terme les tasques de configuració de la funcionalitat (modificant fitxers de configuració i carregant els nous fitxers) i l'script <code>infomem.sh</code> que fa la funcionalitat en si: s'encarrega d'obtenir la mida de \$HOME i mostrar-la a través d'un missatge.</p>			
Decisions de Disseny			
<p>Per a deixar instal·lat l'script s'ha utilitzat la carpeta <code>/usr/local/infomem</code>, a on podem trobar l'executable <code>infomem.sh</code>.</p> <p>Per a que l'script s'executi cada cop que un usuari entri al sistema s'ha decidit modificar el fitxer <code>/etc/profile</code>. Aquest script s'executa cada cop que un usuari fa login en el sistema.</p> <p>Per a mostrar el missatge ens ha semblat que invocar a un terminal o obrir un fitxer de text amb algun editor era una opció poc elegant així que hem cercat informació sobre llibreries que permetessin l'ús de pop-ups i quadres de diàleg. Tot i que hi havia una gran quantitat d'opcions ens hem decidit per utilitzar el programa <code>xmessage</code>. Aquesta llibreria és força antiga però segueix estant actualitzada, de tal manera que s'ha convertit en un estàndard i està present a la majoria de sistemes operatius Unix.</p>			
Codi			
<pre>infomem_config.sh # Comprovem existencia de directoris if [! -d "/usr/local/infomem"]; then echo -e "\nWarning, directory /usr/local/infomem does not exist. Creating..." mkdir /usr/local/infomem fi # copiem s'cript en aquest directori mantenint permisos cp -p infomem.sh /usr/local/infomem # carreguem execució de l'script a ~/.profile si no es troba ja carregada if [-z "\$(more /etc/profile grep /usr/local/infomem/infomem.sh)"] then echo '/usr/local/infomem/infomem.sh' >> /etc/profile fi infomem.sh</pre>			

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

```
xmessage -buttons Ok:0 -default Ok -nearmouse "Hello $USER, your home folder weights $(du -s $HOME | cut -d$'\t' -f1) Bytes ($(du -sh $HOME | cut -d$'\t' -f1))" -timeout 30
```

Joc de Proves

Per a comprovar que els canvis s'han dut a terme satisfactoriament podem dur a terme un reinici `shutdown -r now` o fer que es recarreguin els fitxers de configuració `kill -SIGHUP -1`.

Si hi ha algun problema, podem comprovar que s'han dut a terme els canvis satisfactoriament amb la següent comanda:

```
more /etc/profile | grep /usr/local/infomem/infomem.sh
```

així comprovem que la línia que ha d'haver-hi al `/etc/profile` està present. També podem comprovar que l'script `infomem.sh` es troba en el seu directori:

```
ls -al /usr/local/infomem
```

Hem dut a terme proves en la màquina milax i en Ubuntu i funciona en els dos sistemes.

Nom:	limit nombre processos usuari	Grau d'assoliment	Alt
Permisos:	Root	Propietari:	
Descripció			
Es demana limitar el número de processos màxims a executar per un usuari a 512.			
Decisions de Disseny			
S'utilitza la comanda ulimit, la qual permet consultar i establir els límits de l'usuari; concretament s'afegeix l'argument <code>-u</code> , amb el que s'assignen el nombre màxim de processos disponibles per a una sol usuari. Es requereix adquirir permisos d'administrador.			
Codi			
<code>ulimit -u 512</code>			
Joc de Proves			
Es comanda la mateixa instrucció (ulimit) mb l'argument <code>-a</code> , amb el qual es llisten totes les limitacions de l'usuari actual, permetent corroborar si s'ha assignat correctament la limitació.			

Nom:	monitoritzacio.sh	Grau d'assoliment	Alt
Permisos:	rw-rw-rw-r	Propietari:	Creador de l'script
Ubicació dels fitxers:	Prac2Sistemes/vmstat	Grup:	Creador de l'script
Descripció			
<p>Aquest script compila i executa tres benchmarks en C diferents, amb l'objectiu de testejar la RAM, el disc i la CPU.</p> <p>Durant l'execució posa en marxa vmstat i iostat amb els paràmetres adequats, segons el que s'estigui testejant. La sortida d'aquests es tracta i s'escriu sobre el fitxer log.txt.</p>			
Decisions de Disseny			
<p>S'ha decidit utilitzar aquests programes en C senzills per a veure fins a quin punt el sistema es veu afectat.</p> <p>L'script monitoritzacio.sh simplement s'encarrega de compilar els benchmarks, executar cada benchmark dos cops durant 10 segons amb ràfegues de captura d'1 segon, posar en marxa el vmstat i el iostat amb els arguments adequats per a cada execució del benchmark respectivament i emmagatzemar les dades filtrades a log.txt.</p> <p>Això permet ens permetria obtenir dades de manera ràpida d'un sistema.</p>			
Codi			
<pre> cpu.c #include <stdio.h> void main(){ int i; for(i=0;i<100000000000;i++){ } disk.c #include <stdlib.h> #include <stdio.h> #include <fcntl.h> #include <sys/timeb.h> #define BUFFERSIZE (64*1024) #define MEGABYTE (1024*1024) /* Takes two arguments. Arg 1 is file name. arg 2 is number of megabytes to write. */ main(int argc, char * argv[]) { long megabytes, j, i; int outputfile; char buffer[BUFFERSIZE]; struct timeb before,after; </pre>			


```

if (argc !=3)
{
    printf("Arguments are filename to create and size in megabytes\n");
    return -1;
}

for (j=0 ; j < BUFFERSIZE ;j++)
    buffer[j]='X';

outputfile = open(argv[1], O_CREAT | O_WRONLY | O_SYNC);
megabytes = atol(argv[2]);

ftime(&before);

for (i=0 ; i<((MEGABYTE/BUFFERSIZE)*megabytes); i++)
    write(outputfile,buffer,BUFFERSIZE);

ftime(&after);

long timetowrite=after.time-before.time;

if (timetowrite > 0)
    printf("Bytes per second written = %d\n",
        (MEGABYTE*megabytes)/timetowrite);

close(outputfile);

return 0;
}

```

ram.c

```

#include <stdio.h>
#include <unistd.h>
#include <memory.h>
#define MB 1024 * 1024

void main(){
    int counter = 0;
    while(counter < 10){
        void *p = malloc(100*MB);
        memset(p,0, 100*MB);
        sleep(1);
        counter++;
    }
}

```

#!/bin/bash

<<INSTRUCCIONS

Monitoritzeu els recursos (memòria, cpu i disc) amb la comanda vmstat i engegant diferents programes per a fer un ús intensiu de memòria, de processador i de disc. Creeu diferents programes en C o scripts per tal de fer un ús intensiu del processador, de la memòria i del disc de manera individual.

Monitoritzeu el sistema, executant cada un dels programes de manera individual i comprovant com afecten a les comandes de monitorització (vmstat, iostat)

INSTRUCCIONS

```

#####
#####

```

```

# Autors: Cristòfol Daudèn, Aleix Marine i Josep Marín Llaó
# Data d'implementació: 27/4/2018
# Versio 1.0

```

```

# Permisos:      No necessita permisos especiales.
# Descripció i paràmetres: Es vol implementar un conjunt de benchmarks programats
# en C per a que serveixin de prova per a la monitorització amb la màquina amb la
# que estem treballant.
#
# Tindrem un benchmark per a memòria, cpu i disc. Duen a terme una reserva de
# memòria intensiva utilitzant el programa malloc, duent a terme un for de milions
# d'iteracions i finalment el programa de disc està continuament escrivint a disc
# amb la mida de bloc passada per paràmetre.
#
# Després d'executar aquests benchmark de manera individual (durant 10 segons) iniciem
# els programes vmstat i iostat que comencen a monitoritzar diferents paràmetres del
# sistema.
#
# Aquest script automatitza i refina les dades obtingudes en aquest últim procés,
# mostrant aquells camps més interessants, escrivint-los de manera ordenada en el
# fitxer log.txt.
#
# L'únic paràmetre disponible és el de l'ajuda
#####
#####

#Mirem parametre d'ajuda
if [ "$1" = "-h" ]; then
    ajuda
    exit 0
fi

echo "          RESULTATS" > log.txt
echo "....." >> log.txt
make all

echo "CPU TEST" >> log.txt
echo "VMSTAT" >> log.txt
timeout 10 ./cpu &
vmstat -an 1 10 | tail -n +3 | awk {'print $13 " % CPU"}' >> log.txt
echo "IOSTAT" >> log.txt
timeout 10 ./cpu &
iostat -c -y 1 10 | tail -n +3 | tr -s '\n' | grep -v -e "^avg-cpu" | awk {'print $1 " % CPU"}' >> log.txt

echo "RAM TEST" >> log.txt
echo "VMSTAT" >> log.txt
timeout 10 ./ram &
vmstat -an 1 10 | tail -n +3 | awk {'print $4 " kB lliures"}' >> log.txt

echo "DISK TEST" >> log.txt
echo "VMSTAT" >> log.txt
timeout 10 ./disk test 5120 &
vmstat -dn 1 10 | tail -n +3 | awk {'print $1"\tReads: "$2" operacions read\t" "Writes: "$6"
operacions write"}' >> log.txt
echo "IOSTAT" >> log.txt
timeout 10 ./disk test 5120 &
iostat -d -y 1 10 | tail -n +3 | grep -v -e "^Device" | tr -s '\n' | awk {'print $1"\tReads: "$5" kB\t"
"Writes: "$6 " kB"}' >> log.txt

echo "Finished"

```

Joc de Proves

Per a provar aquest conjunt de benchmarks podem executar l'script i posar en marxa el vmstat i l'iostat amb ràfegues indefinides i comparar els resultats.

Després de veure les dades al fitxer log.txt observem que l'ús de memòria, CPU i disc es dispara en cadascun dels benchmarks.