

GESTIÓ DE SISTEMES I XARXES
PRÀCTICA 2



UNIVERSITAT ROVIRA I VIRGILI
Escola Tècnica
Superior d'Enginyeria

Genís Martínez Garrido
2023 – 2024
GEI

ÍNDEX

ADMINISTRACIÓ.....	3
SERVEIS.....	8
PROCESSOS.....	13

ADMINISTRACIÓ

Durant aquesta pràctica del curs de Gestió de Sistemes i Xarxes, m'he centrat en el desenvolupament de scripts en Bash i en la administració de sistemes Linux. Els objectius d'aquest laboratori han estat establir una metodologia de treball eficaç, implementar scripts de manera correcta, verificar la seva funcionalitat i aprendre tècniques d'instal·lació i actualització de programari.

En aquesta sessió, hem explorat diverses tasques relacionades amb la gestió de paquets i la monitorització del sistema. Primerament, he après els diferents mètodes d'instal·lació de programari en sistemes Debian i RPM, així com la configuració dels seus repositoris i l'ús de comandes com apt, apt-get, dpkg i alien.

Després, ens hem endinsat en el sistema de log de Linux per comprendre quins registres s'emmagatzemen, per què són importants i com consultar-los per detectar problemes, atacs informàtics o comportaments anòmals del sistema.

Finalment, m'he centrat en el desenvolupament de dos scripts en Bash: un per consultar la versió i les dependències dels paquets Debian instal·lats, i l'altre per desinstal·lar paquets mantenint-ne la configuració i les dependències.

Consulta paquet:

```
#!/bin/bash

# Autor (Genís Martínez Garrido - GSX)

date=$(date +"%Y-%m-%d %H:%M:%S")
if [ $# -gt 0 ]; then
    # Iterem sobre cada argument
    for paquet in "$@"
    do
        # Comprovo si el paquet ja està instal·lat
        if apt list --installed 2>&1 | grep -w "$paquet" >/dev/null 2>&1; then
            echo "Informació del paquet $paquet:"
            # Versió actualment instal·lada
            installed_version=$(dpkg -l | grep -w "$paquet" | awk '{print $3}')
            echo "Versió actualment instal·lada: $installed_version"

            # Data i hora d'instal·lació
            install_date=$(stat -c %y /var/lib/dpkg/info/"$paquet".list)
            echo "Data i hora d'instal·lació: $install_date"

            # Disponibilitat d'actualització
            update_info=$(apt list --upgradable 2>&1 | grep -w "$paquet" | awk -F/ '{print $1}')
            if [ -n "$update_info" ]; then
```

```

        echo "Disponibilitat d'actualització: Hi ha una nova versió disponible"
    else
        echo "Disponibilitat d'actualització: No hi ha cap nova versió disponible"
    fi

    # Llista de dependències del paquet
    dependencies=$(apt-cache depends "$paquet" | grep "Depends" | awk '{print
$2}')
    echo "Llista de dependències del paquet:"
    echo "$dependencies"

    # Fitxers de configuració associats amb el paquet
    config_files=$(dpkg -L "$paquet")
    echo "Fitxers de configuració associats amb el paquet:"
    echo "$config_files"
else
    # Si no està instal·lat, mostro per pantalla que no està instal·lat
    echo "$date $paquet no està instal·lat."
fi
done
else
    echo "No s'ha passat cap argument com a paràmetre."
    exit 0
fi

```

Aquest script en Bash, que he anomenat "**consulta_paquet.sh**", té com a objectiu proporcionar informació detallada sobre els paquets Debian instal·lats al sistema. El que faig és el següent:

1. **Verifico els arguments rebuts:** He començat el script verificant si s'han passat arguments. Si no, mostro un missatge d'error i finalitzo l'execució.
2. **Itero sobre els paquets rebuts:** Si s'han rebut arguments, el script itera sobre cada paquet rebut com a argument.
3. **Comprovo si el paquet està instal·lat:** Per a cada paquet, comprovo si està instal·lat al sistema utilitzant la comanda `apt list --installed`. Si el paquet està instal·lat, mostro informació rellevant; si no, indico que el paquet no està instal·lat.
4. **Informació del paquet instal·lat:**
 - Versió actualment instal·lada: Utilitzo la comanda `dpkg -l` per obtenir la versió del paquet.
 - Data i hora d'instal·lació: Faig servir la comanda `stat` per obtenir la data i hora d'instal·lació del fitxer associat al paquet.
 - Disponibilitat d'actualització: Utilitzo `apt list --upgradable` per verificar si hi ha una nova versió disponible.
 - Llista de dependències del paquet: Utilitzo `apt-cache depends` per mostrar les dependències del paquet.

- Fitxers de configuració associats amb el paquet: Faig servir dpkg -L per obtenir els fitxers de configuració.
5. **Sortida d'informació detallada:** Mostro per pantalla tota la informació recopilada per a cada paquet.

Desinstala paquet:

```
#!/bin/bash

# Author (Genís Martínez Garrido - GSX)

# Verificar que s'ha proporcionat almenys un argument
if [ $# -eq 0 ]; then
    echo "Mode d'ús: $0 <package_name>"
    exit 1
fi

package_name="$1"

# Comprovar si el paquet està instal·lat
if dpkg -l | grep -q "^ii.*$package_name "; then
    # Desinstal·lar el paquet
    apt-get remove "$package_name"
    # Comprovar si s'ha produït algun error durant l'execució de apt-get remove
    if [ $? -eq 0 ]; then
        echo "El paquet $package_name s'ha desinstal·lat amb èxit."
    else
        echo "S'ha produït un error durant la desinstal·lació del paquet $package_name."
        exit 1
    fi
else
    echo "El paquet $package_name no està instal·lat."
    exit 1
fi
```

Aquest script, que he nomenat "**desinstala_paquet.sh**", té com a funció principal desinstal·lar un paquet del sistema. Aquí detallo com funciona:

1. **Verifico els arguments rebuts:** El script comença verificant si s'ha proporcionat almenys un argument a la línia de comandes. Si no s'ha proporcionat cap argument, mostra un missatge d'ús del script i finalitza l'execució.
2. **Obtinc el nom del paquet:** Si s'ha proporcionat almenys un argument, el script guarda el primer argument com a nom del paquet que es vol desinstal·lar.
3. **Comprovo si el paquet està instal·lat:** Utilitzo la comanda dpkg -l per llistar tots els paquets instal·lats al sistema i comprovar si el paquet que es vol

4. **Desinstalo del paquet:** Utilitzo `apt-get remove` per desinstal·lar el paquet del sistema. Posteriorment, comprovo si s'ha produït algun error durant l'execució d'aquesta comanda.
5. **Gestió d'errors:** Si la desinstal·lació s'ha completat sense errors, mostro un missatge indicant que el paquet s'ha desinstal·lat amb èxit. En cas contrari, mostro un missatge d'error informant que s'ha produït un error durant la desinstal·lació.

```
milax@casa: ~/Escriptori/GSX_Labs/Prac2/Administracio
milax@casa:~/Escriptori/GSX_Labs/Prac2/Administracio$ ./consulta_paquet.sh gedit
Informació del paquet gedit:
Versió actualment instal·lada: 3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.38.1-1
3.0.3-6
0.18-2
Data i hora d'instal·lació: 2023-02-20 19:29:11.853478437 +0100
Disponibilitat d'actualització: No hi ha cap nova versió disponible
Llista de dependències del paquet:

Fitxers de configuració associats amb el paquet:
/.
/usr
/usr/bin
/usr/bin/gedit
/usr/lib
/usr/lib/python3
/usr/lib/python3/dist-packages
/usr/lib/python3/dist-packages/gi

milax@casa:~/Escriptori/GSX_Labs/Prac2/Administracio$ ./consulta_paquet.sh acct
2024-03-08 09:51:04 acct no està instal·lat.
```

```
milax@casa: ~/Escriptori/GSX_Labs/Prac2/Administracio
milax@casa:~/Escriptori/GSX_Labs/Prac2/Administracio$ sudo ./desinstal_paquet.sh gedit
[sudo] contrasenya per a milax:
S'està lligint la llista de paquets... Fet
S'està construint l'arbre de dependències... Fet
S'està lligint la informació de l'estat... Fet
El paquets següents s'han instal·lat automàticament i ja no són necessaris:
accountsservice acpi-support-base acpid aglfn analisi-i-disseny-daplicacions apache-netbeans apg
arquitectura-de-computadors architectures-per-a-aplicacions-en-xarxa
arquitectures-segures-de-sistemes-distribuits architectures-segures-de-sistemes-distribuits-extra
asciinema atom autoconf automake autotools-dev bases-de-dades bases-de-dades-avancades
bases-de-dades-avancades-extra bind9-dnsutils bind9-host bind9-libs binutils-alpha-linux-gnu
bison bison-doc cacti-hplabs cgdb cheese-common chronograf cm-super-minimal code codeblocks
codeblocks-common codeblocks-contrib coinor-libc3 coinor-libcgl1 coinor-libclp1
coinor-libcoinplv5 coinor-libcoinutils3v5 coinor-libosilv5 colas colord-data
computacio-algebraica containerd cpp-10-alpha-linux-gnu cpp-alpha-linux-gnu cysignals-tools
cython3 ddd ddd-doc debian-handbook default-jdk default-jdk-doc default-jdk-headless
desenvolupament-avancat-daplicacions-web dhcpcd dnsutils docker.io docutils-common doxygen
doxygen-doc doxygen-gui doxygen-latex eclib-tools eclipse-java eclipse-jee erlang-asn1
erlang-base erlang-crypto erlang-eldap erlang-ftp erlang-inets erlang-mnesia erlang-os-mon
erlang-parsetools erlang-public-key erlang-runtime-tools erlang-snmp erlang-ssl
erlang-syntax-tools erlang-tftp erlang-tools erlang-xmerl estadistica estadistica-extra
estructura-de-computadors estructura-de-computadors-extra estructures-de-dades ethtool ff
fflas-ffpack fflas-ffpack-common file-roller filezilla filezilla-common firefox-esr-l10n-ca
firefox-esr-l10n-es-es flex flex-doc flintqs fluid-soundfont-gs fonaments-de-programacio
fonaments-de-programacio-extra fonaments-de-sistemes-operatius fonts-font-awesome
fonts-glyphicons-halflings fonts-inconsolata fonts-lyx fonts-mathjax g++ g++-10 gamin
gap-atlasrep gap-core gap-dev gap-gapdoc gap-libs gap-online-help gap-primgrp gap-smallgrp
gap-table-of-marks gap-transgrp gcc gcc-10-alpha-linux-gnu gcc-10-alpha-linux-gnu-base
gcc-10-cross-base-ports gcc-alpha-linux-gnu gconf2-common gdb gdb-doc gdebi gdebi-core geany
geany-common gedit-common gedit-plugins-common gestio-de-sistemes-i-xarxes-extra
gestio-dinfraestructures-per-a-la-salut gfan gfortran-10 glib2.0-glib2.0 glib2.0-gtksource-3.0
glib2.0-gtksource-4 glib2.0-gucharmap-2.90 glib2.0-handy-0.0 glib2.0-javascriptcoregtk-4.0
glib2.0-lightdm-1 glib2.0-nautilus-3.0 glib2.0-notify-0.7 glib2.0-vte-2.91 glib2.0-webkit2-4.0 git
git-doc git-gui git-man gitk glpk-utils glx-alternative-mesa glx-alternative-nvidia
glx-diversions gmp-ecm gnome-applets gnome-applets-data gnome-calculator
gnome-control-center-data gnome-disk-utility gnome-packagekit gnome-packagekit-data gnome-panel
gnome-panel-data gnome-screenshot gnome-session-bin gnome-session-common gnome-session-flashback
gnome-shell-extension-prefs gnome-system-monitor gnome-tweaks gnuplot gnuplot-data gnuplot-doc
gnuplot-qt gromit-mpx gstreamer1.0-clutter-3.0 gyp hping3 ibverbs-providers influxdb
infraestructures-per-al-big-data infraestructures-per-al-big-data-extra intellij-idea-community

S'està desinstal·lant architectures-per-a-aplicacions-en-xarxa-extra (20230217-1)...
S'està desinstal·lant computacio-paral.lela-i-massiva (20230217-1)...
S'està desinstal·lant computacio-paral.lela-i-massiva-extra (20230217-1)...
S'està desinstal·lant desenvolupament-avancat-daplicacions-web-extra (20230217-1)...
S'està desinstal·lant estructures-de-dades-extra (20230217-1)...
S'està desinstal·lant fonaments-de-sistemes-operatius-extra (20230217-1)...
S'està desinstal·lant sistemes-multiagent (20230217-1)...
S'està desinstal·lant llenguatges-formals (20230217-1)...
S'està desinstal·lant sistemes-multiagent-extra (20230217-1)...
S'està desinstal·lant llenguatges-formals-extra (20230217-1)...
S'està desinstal·lant gedit-plugins (3.38.1-1)...
S'està desinstal·lant gedit-plugin-bookmarks (3.38.1-1)...
S'està desinstal·lant gedit-plugin-bracket-completion (3.38.1-1)...
S'està desinstal·lant gedit-plugin-character-map (3.38.1-1)...
S'està desinstal·lant gedit-plugin-code-comment (3.38.1-1)...
S'està desinstal·lant gedit-plugin-color-picker (3.38.1-1)...
S'està desinstal·lant gedit-plugin-color-schemer (3.38.1-1)...
S'està desinstal·lant gedit-plugin-commander (3.38.1-1)...
S'està desinstal·lant gedit-plugin-draw-spaces (3.38.1-1)...
S'està desinstal·lant gedit-plugin-find-in-files (3.38.1-1)...
S'està desinstal·lant gedit-plugin-git (3.38.1-1)...
S'està desinstal·lant gedit-plugin-join-lines (3.38.1-1)...
S'està desinstal·lant gedit-plugin-multi-edit (3.38.1-1)...
S'està desinstal·lant gedit-plugin-session-saver (3.38.1-1)...
S'està desinstal·lant gedit-plugin-smart-spaces (3.38.1-1)...
S'està desinstal·lant gedit-plugin-synctex (3.38.1-1)...
S'està desinstal·lant gedit-plugin-terminal (3.38.1-1)...
S'està desinstal·lant gedit-plugin-text-size (3.38.1-1)...
S'està desinstal·lant gedit-plugin-translate (3.38.1-1)...
S'està desinstal·lant gedit-plugin-word-completion (3.38.1-1)...
S'està desinstal·lant gedit-source-code-browser-plugin (3.0.3-6)...
S'està desinstal·lant rabbitvcs-gedit (0.18-2)...
S'està desinstal·lant gedit (3.38.1-1)...
S'estan processant els activadors per a gnome-menus (3.36.0-1)...
S'estan processant els activadors per a libglib2.0-0:amd64 (2.66.8-1)...
S'estan processant els activadors per a libglib2.0-0:i386 (2.66.8-1)...
S'estan processant els activadors per a mailcap (3.69)...
S'estan processant els activadors per a desktop-file-utils (0.26-1)...
El paquet gedit s'ha desinstal·lat amb èxit.
milax@casa:~/Escriptori/GSX_Labs/Prac2/Administracio$
```

SERVEIS

En aquest laboratori de "Boot i Serveis", l'objectiu principal és comprendre el procés d'arrencada del sistema i la gestió dels serveis en un entorn Linux. Aprofundirem en els següents punts:

1. **Seqüència de boot del sistema Linux:** S'analitzarà la seqüència d'arrencada d'un sistema Linux utilitzant les eines proporcionades pel systemd, com ara systemd-analyze blame i systemd-analyze critical-chain. A més, es generarà un diagrama visual de la seqüència d'arrencada per a una millor comprensió.
2. **Comportament dels serveis Linux:** Es realitzarà un estudi detallat dels serveis Linux, incloent la seva configuració i gestió. Es revisarà l'estructura dels directoris dels serveis tant per systemd com per SysV, com ara /etc/systemd/system/, /etc/init.d/, entre d'altres. A més, es consultaran recursos en línia per comprendre millor com utilitzar systemctl per gestionar els serveis.

En aquesta sessió pràctica, s'executaran els següents exercicis:

1. **Programació d'un servei estil SysV (consultaSYSV):** He desenvolupant un servei que, en el nivell d'inicialització per defecte, executa l'script anomenat "consulta_servei.sh" en el moment de l'arrencada del sistema. Aquest servei ha de ser capaç de rebre paràmetres d'entrada des d'un fitxer i registrarà els errors en un fitxer de registre en cas que no s'especifiquin paràmetres o no existeixi el fitxer de paràmetres.
2. **Programació d'un servei estil systemd (consultaSYSD):** He desenvolupat un servei systemd que s'iniciarà en el nivell multiusuari i executarà el mateix script "consulta_servei.sh" adaptat a les necessitats del servei.

Consulta servei (modificat):

```
#!/bin/bash

# Autor (Genís Martínez Garrido - GSX)

date=$(date +"%Y-%m-%d %H:%M:%S")

if [ $# -eq 0 ]; then
    if [ -f "/tmp/paquets" ]; then
        # Si no se pasan argumentos pero existe el archivo /tmp/paquets,
        # se toman los parámetros del archivo
        args=$(cat /tmp/paquets)
    else
        # Si no se pasan argumentos y no existe el archivo /tmp/paquets,
        # se indica en el archivo de registro
        logger -s "No s'han passat arguments i el fitxer /tmp/paquets no existeix."
```



```

        exit 1
    fi
else
    args="$@"
fi

# Iterem sobre cada argument
for paquet in $args
do
    # Comprovo si el paquet ja està instal·lat
    if apt list --installed 2>&1 | grep -w "$paquet" >/dev/null 2>&1; then
        echo "Informació del paquet $paquet:"
        # Versió actualment instal·lada
        installed_version=$(dpkg -l | grep -w "$paquet" | awk '{print $3}')
        echo "Versió actualment instal·lada: $installed_version"

        # Data i hora d'instal·lació
        install_date=$(stat -c %y /var/lib/dpkg/info/"$paquet".list)
        echo "Data i hora d'instal·lació: $install_date"

        # Disponibilitat d'actualització
        update_info=$(apt list --upgradable 2>&1 | grep -w "$paquet" | awk -F/ '{print $1}')
        if [ -n "$update_info" ]; then
            echo "Disponibilitat d'actualització: Hi ha una nova versió disponible"
        else
            echo "Disponibilitat d'actualització: No hi ha cap nova versió disponible"
        fi

        # Llista de dependències del paquet
        dependencies=$(apt-cache depends "$paquet" | grep "Depends" | awk '{print $2}')
        echo "Llista de dependències del paquet:"
        echo "$dependencies"

        # Fitxers de configuració associats amb el paquet
        config_files=$(dpkg -L "$paquet")
        echo "Fitxers de configuració associats amb el paquet:"
        echo "$config_files"
    else
        # Si no està instal·lat, mostro per pantalla que no està instal·lat
        echo "$date $paquet no està instal·lat."
    fi
done

```

Consulta SYSV:

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

```

#### BEGIN INIT INFO
# Provides:      consultaSYSV
# Required-Start: $local_fs $remote_fs
# Required-Stop:  $local_fs $remote_fs
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Start consulta script at boot time
# Description:    Enable service provided by consulta script.
#### END INIT INFO

# Path absolut de l'arxiu a executar
SCRIPT="/home/milax/Escriptori/GSX_Labs/Prac2/Serveis/consulta_paquet_modi
ficat.sh"

# Funció per comprovar si l'arxiu /tmp/paquets existeix i no està buit
check_file() {
    if [[ -f /tmp/paquets ]]; then
        return 0 # L'arxiu existeix i té contingut
    else
        return 1 # L'arxiu no existeix o està buit
    fi
}

case "$1" in
    start)
        # Comprovem si s'han pasat paràmetres al script
        if [[ $# -eq 1 ]]; then
            # No s'han passat arguments
            if check_file; then
                # L'arxiu /tmp/paquets existeix i té contingut
                $SCRIPT $(cat /tmp/paquets)
            else
                # L'arxiu /tmp/paquets no existeix o està buit
                logger -f "El fitxer /tmp/paquets no existeix o és buit."
            fi
        else
            # S'han passat arguments al script
            shift
            $SCRIPT "$@"
        fi
        exit 0
        ;;
    stop)
        # No es necessari fer res per parar el servei
        exit 0
        ;;
)

```

```
*)
    echo "Ús: $0 {start|stop}"
    exit 1
;;
esac

exit 0
```

L'anterior script conté el següent:

1. **Metadades d'inicialització:** Les primeres línies del script contenen informació sobre el servei, com ara el nom del servei, les dependències del sistema, els nivells d'inici i finalització, i una descripció curta i llarga del servei.
2. **Definició del camí de l'arxiu a executar:** S'especifica el camí absolut de l'arxiu que s'executarà com a servei. En aquest cas, és l'arxiu "consulta_paquet_modificat.sh".
3. **Funció per comprovar l'existència de l'arxiu /tmp/paquets:** He definit una funció anomenada "check_file" que comprova si l'arxiu "/tmp/paquets" existeix i té contingut. Aquesta funció retorna un valor 0 si l'arxiu existeix i té contingut, i un valor 1 si l'arxiu no existeix o està buit.
4. **Bloc de comandaments principal (case statement):** Aquest bloc de comandaments processa els arguments passats al script i executa les accions adequades. Si l'argument és "start", el script comprova si s'han passat paràmetres i executa l'script corresponent, tenint en compte l'existència de l'arxiu "/tmp/paquets". Si l'argument és "stop", no es realitza cap acció ja que no és necessari parar el servei.
5. **Gestió de les opcions d'inici i finalització del servei:** El script finalitza amb un codi d'èxit 0 si s'executa correctament.

Consulta SYSD:

```
[Unit]
Description=Servei de consultaSYSD
After=network-online.target

[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/consulta_paquet.sh

[Install]
WantedBy=multi-user.target
```

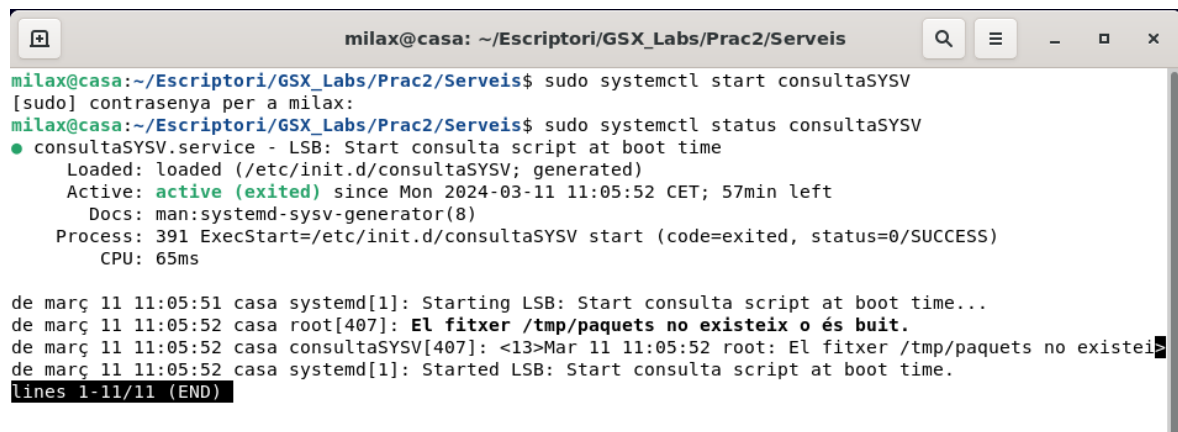
Les seccions d'aquest fitxer:

1. **Metadades del servei (Unit):** Aquesta secció conté metadades sobre el servei, com ara la descripció del servei i les dependències. En aquest cas,

he especificat que el servei "consultaSYSD" s'iniciarà després que la xarxa estigui disponible ("network-online.target").

2. **Definició del servei (Service):** En aquesta secció definirem el comportament del servei. El tipus de servei s'especifica com a "simple", indicant que s'executarà un procés senzill. L'acció a realitzar durant l'inici del servei es defineix amb "ExecStart", on indico que s'executarà el script "consulta_paquet.sh" mitjançant /bin/bash.
3. **Instal·lació del servei (Install):** En aquesta secció especifico com s'instal·larà el servei. El servei s'instal·larà per ser executat en el nivell d'arrencada "multi-user.target".

Joc de proves:



```
milax@casa: ~/Escriptori/GSX_Labs/Prac2/Serveis
milax@casa:~/Escriptori/GSX_Labs/Prac2/Serveis$ sudo systemctl start consultaSYSV
[sudo] contrasenya per a milax:
milax@casa:~/Escriptori/GSX_Labs/Prac2/Serveis$ sudo systemctl status consultaSYSV
● consultaSYSV.service - LSB: Start consulta script at boot time
   Loaded: loaded (/etc/init.d/consultaSYSV; generated)
   Active: active (exited) since Mon 2024-03-11 11:05:52 CET; 57min left
     Docs: man:systemd-sysv-generator(8)
  Process: 391 ExecStart=/etc/init.d/consultaSYSV start (code=exited, status=0/SUCCESS)
    CPU: 65ms

de març 11 11:05:51 casa systemd[1]: Starting LSB: Start consulta script at boot time...
de març 11 11:05:52 casa root[407]: El fitxer /tmp/paquets no existeix o és buit.
de març 11 11:05:52 casa consultaSYSV[407]: <13>Mar 11 11:05:52 root: El fitxer /tmp/paquets no existeix
de març 11 11:05:52 casa systemd[1]: Started LSB: Start consulta script at boot time.
lines 1-11/11 (END)
```

He executat i parat els serveis varies vegades utilitzant els dos sistemes d'inicialització per a comprovar el seu correcte funcionament. He utilitzat també el journalctl per comprovar el que estava passant en cada moment i si s'estava realitzant tot bé.

PROCESOS

Els objectius d'aquest laboratori inclouen l'anàlisi de l'ocupació del processador i les tasques que s'estan duent a terme en un moment donat, l'aprenentatge de l'ús dels senyals en Bash, la implementació de processos periòdics i l'estudi detallat dels cgroups i de l'accounting de processos.

En aquest laboratori explorarem les tasques assignades, que inclouen l'observació dels processos en curs i l'anàlisi del directori /proc, així com l'ús dels senyals en Bash i la comprensió dels límits i restriccions dels processos.

La part fonamental d'aquest laboratori implica la realització de dos exercicis pràctics, com la creació d'un script per a la gestió de senyals i l'exploració dels cgroups. A través d'aquests exercicis, estem reforçant els nostres coneixements i habilitats en la gestió de processos en sistemes Linux.

Cgroup:

```
#!/bin/bash

# Obté el cgroup de la shell actual
cgroup=$(awk -F: '/cpu|cpuacct/ {print $3}' /proc/self/cgroup)

# Troba els processos en el cgroup
pids=$(grep -l $cgroup /proc/*/cgroup | cut -d/ -f3)

# Comprova si no s'han trobat PIDs
if [ -z "$pids" ]; then
    echo "No s'han generat processos en aquest cgroup."
else
    # Itera sobre els PIDs trobats
    for pid in $pids; do
        # Obté el PID i el PPID del procés
        pid_val=$(awk '/Pid/ {print $2}' /proc/$pid/status)
        ppid_val=$(awk '/PPid/ {print $2}' /proc/$pid/status)

        # Obté la comanda del procés
        command_val=$(tr -d '\0' < /proc/$pid/cmdline)

        # Imprimim la informació del procés
        echo "PID: $pid_val, PPID: $ppid_val, Comanda: $command_val"
    done
fi
```

Funcionament:

1. **Obtenció del cgroup de la meua shell actual:** Utilitzo la comanda awk per llegir el directori /proc/self/cgroup i extreure la ruta del cgroup associat amb

els comptadors de temps de CPU (cpu) i els comptadors de temps de CPU acumulat (cpuacct). Aquesta informació es guarda en la variable cgroup.

2. **Recerca dels processos en el cgroup:** Faig servir la comanda grep per trobar els PIDs dels processos que pertanyen al cgroup identificat anteriorment. Aquesta informació es guarda en la variable pids.
3. **Comprovació de la presència de PIDs:** Verifico si s'han trobat PIDs relacionats amb el cgroup. Si la variable pids està buida, el script mostra un missatge indicant que no s'han generat processos en aquest cgroup.
4. **Iteració sobre els PIDs trobats:** Si s'han trobat PIDs, itero a través de cadascun d'ells utilitzant una instrucció for.
5. **Obtenció del PID, PPID i comanda del procés:** Per a cada PID, utilitzo les comandes awk i tr per llegir el fitxer /proc/\$pid/status i obtenir la informació relativa al PID, PPID (ID del procés pare) i la comanda associada al procés. Aquesta informació es guarda en les variables pid_val, ppid_val i command_val.
6. **Impressió de la informació del procés:** Imprimeixo la informació obtinguda per a cada procés, incloent el PID, el PPID i la comanda associada.

Cgroup Father:

```
#!/bin/bash

# Obté el cgroup de la shell actual
cgroup=$(awk -F: '/cpu|cpuacct/ {print $3}' /proc/self/cgroup)

# Troba el PID del procés actual
current_pid=$(echo $$)

# Funció per obtenir el PID del procés pare
get_parent_pid() {
    local pid=$1
    # Obté el PPID del procés
    ppid=$(awk '/PPid/ {print $2}' /proc/$pid/status)
    echo "$ppid"
}

# Funció per obtenir la comanda associada amb un PID
get_command() {
    local pid=$1
    # Obté la comanda del procés
    command_val=$(tr -d '\0' < /proc/$pid/cmdline)
    echo "$command_val"
}

# Obté el PID del procés pare inicial
parent_pid=$(get_parent_pid $current_pid)
```

```
# Imprimeix la informació del procés actual
echo "PID: $current_pid, PPID: $parent_pid, Comanda: $(get_command
$current_pid)"

# Itera fins arribar al primer procés generat en el mateix cgroup
while [ $parent_pid -ne 1 ]; do
    # Obté la comanda i el PPID del procés pare
    parent_command=$(get_command $parent_pid)
    echo "PID: $parent_pid, PPID: $(get_parent_pid $parent_pid), Comanda:
$parent_command"
    # Obté el PID del següent procés pare
    parent_pid=$(get_parent_pid $parent_pid)
done
```

Funcionament:

1. **Obtenció del cgroup de la shell actual:** Utilitzo la comanda `awk` per llegir el directori `/proc/self/cgroup` i extreure la ruta del cgroup associat amb els comptadors de temps de CPU (`cpu`) i els comptadors de temps de CPU acumulat (`cpuacct`). Aquesta informació es guarda en la variable `cgroup`.
2. **Obtenció del PID del procés actual:** Utilitzo `echo $$` per obtenir el PID del procés actual de la shell i guardar-lo a la variable `current_pid`.
3. **Funcions per obtenir el PID del procés pare i la comanda associada a un PID:** Defineixo dues funcions: `get_parent_pid` per obtenir el PPID (ID del procés pare) i `get_command` per obtenir la comanda associada a un PID donat.
4. **Obtenció del PID del procés pare inicial:** Utilitzo la funció `get_parent_pid` per obtenir el PPID del procés pare inicial i guardar-lo a la variable `parent_pid`.
5. **Impressió de la informació del procés actual:** Mostro la informació del procés actual, incloent el PID, el PPID i la comanda associada, utilitzant la comanda `echo`.
6. **Iteració fins al primer procés generat en el mateix cgroup:** Utilitzo un bucle `while` per iterar fins al primer procés que té un PPID igual a 1, el qual indica el primer procés del sistema. En cada iteració, mostro la informació del procés pare utilitzant les funcions `get_command` i `get_parent_pid`.

Joc de proves:

He comprovat desde diferents terminals amb processos actius i altres amb processos acabats el funcionament dels scripts utilitzant l'identificador dels terminals.