

Simulacre Prova Final UF1 Sistemes (Part II)

SYSTEMD

1.-a) Suposant que la comanda `systemctl cat graphical.target | grep -E "(Requires|Wants|After)"` mostrés la següent sortida...

```
Requires=multi-user.target
Wants=display-manager.service
After=multi-user.target display-manager.service
```

...raona què farien els valors concrets que hi ha indicats a cadascuna de les directives anteriors.

aII) Utilitzant aquest "truc" d'observar les línies respectives `Requires=`, `Wants=` i/o `After=`, ordena, per ordre d'assoliment durant l'arranc del sistema, els següents units: "multi-user.target", "basic.target", "sysinit.target", "sockets.target", "timers.target", "tmp.mount" i "local-fs.target" i justifica-ho amb les captures de pantalla pertinents

b) Suposant que la comanda `systemctl cat gdm.service` mostrés la següent sortida...

```
[Unit]
Conflicts=getty@tty1.service
After=getty@tty1.service
OnFailure=reboot.target
[Service]
ExecStart=/usr/sbin/gdm
Restart=always
BusName=org.gnome.DisplayManager
EnvironmentFile=-/etc/locale.conf
ExecReload=/bin/kill -SIGHUP $MAINPID
[Install]
Alias=display-manager.service
```

...raona què farien els valors concrets que hi ha indicats a cadascuna de les directives anteriors.

c) Suposant que tinguéssim el fitxer `/etc/systemd/system/hola.timer` amb el contingut següent, digues què passaria després d'executar la comanda `systemctl enable hola.timer`

```
[Unit]
Description=Hola amigo
[Timer]
OnBootSec=30m
RandomizedDelaySec=10m
Unit=reboot.target
[Install]
WantedBy=timers.target
```

d) Digues el nom i contingut del/s arxiu/s de tipus "mount" que equivaldria tenir activat/s per substituir l'efecte de tenir un arxiu `/etc/fstab` amb les línies següents.

```
/dev/sda1    /      ext4  defaults,noatime 0 0
/dev/sda2    /home  xfs   defaults,noatime 0 0
```

2.-a) Crea una "unit" de tipus "service" que en iniciar-se executi la comanda `rm -rf /opt/*` (per triar el seu *Type* adient, tingues en compte que aquesta comanda finalitza!) i habilita-la per a què es posi en marxa cada cop que el sistema arribi al target "multi-user". Un cop reiniciat el sistema, ¿apareix algun missatge de registre al Journal conforme s'ha posat en marxa aquesta "unit"? (i en tot cas, quin seria?).

b) Crea la "unit" de tipus "timer" necessària per executar la "unit" "service" creada a l'apartat anterior (suposant que aquesta ja no estigui habilitada i el que estigui habilitat sigui el "timer") cada quart d'hora exacte (és a dir, a les :00, :15, :30 i :45 minuts).

c) Atura i deshabilita el "timer" anterior i crea ara un altre "timer" que, un cop habilitat, executi la "unit" "service" creada a l'apartat anterior cada quart d'hora a partir de la 1ª vegada que s'activi el "timer", la qual serà al cap de 10 minuts d'haver arrencat el sistema.

d) ¿Què creus que passaria si afegeixes la línia `RemainAfterExit=yes` a la "unit" "service" quan toqués executar-se per segon cop degut a un "timer"?

3.-PAS PREVI: Escriu un Bash shell script que executi cada 5 segons (en un bucle infinit) això: `curl -s "https://api.chucknorris.io/jokes/random?category=career" | jq ".value"` En executar-lo hauràs de veure un acudit per pantalla cada 5 segons sense parar. Un cop ho tinguis fet, passa a realitzar els següents apartats:

a) Defineix una unit de tipus "service" **a nivell d'usuari** que, en iniciar-se (amb `systemctl --user start`, sense ésser "root"!) executi el shell script que acabes d'escriure (per triar el seu *Type* adient, tingues en compte que l'script no finalitza!). ¿On s'envien ara els acudits i per què?. Fes a més que en aturar-se (amb `systemctl --user stop`) aquesta unit envii amb `systemd-cat` el missatge "NOO!" de tipus "crític" al Journal. Fes, finalment, que la comanda `systemctl enable` no funcioni amb aquesta unit per cap target.

b) Modifica l'arxiu "service" de l'apartat anterior per a convertir-lo en una plantilla de tal manera què es pugui establir un missatge diferent en el moment d'aturar cada instància concreta del servei.

4.-PAS PREVI: Arrenca una màquina virtual que sigui de tipus Server (Ubuntu o Fedora, tant és) i que tingui la seva tarja de xarxa en mode "adaptador pont". Assegura't de tenir el servidor SSH encès i el tallafocs apagat

a) Crea la carpeta `"/etc/systemd/system/getty@tty1.service.d"` i, allà dins, l'arxiu "override.conf" amb el següent contingut:

```
[Service]
ExecStart=
ExecStart=/usr/bin/top
Restart=always
User=usuari #Es pot posar un usuari qualsevol
```

¿Per a què serveix aquest fitxer i quina relació hi té amb el fitxer (ja existent) `"/etc/systemd/system/getty@.service"` ? ¿Per què cal afegir la primera línia `ExecStart=` ?

b) Fes el necessari per deshabilitar l'activació automàtica de la resta de "ttyps" (tty2,tty3,etc) per tal que en iniciar la màquina, només aparegui la comanda top i l'usuari no pugui "escapar" d'allà.

NOTA: És per això que s'ha demanat tenir un servidor SSH encès, per poder entrar a la màquina des d'allà i poder revertir els canvis en cas que es vulgui reutilitzar la màquina virtual per altres tasques

PISTA: Recorda que això es pot aconseguir modificant adientment la configuració del servei Systemd-logind

USB-IP (<https://www.kernel.org/doc/readme/tools-usb-usbip-README>) és un conjunt de mòduls del kernel i utilitats de terminal que permeten compartir dispositius USB a través d'una xarxa IP; així, tot i que físicament estiguin endollats a una màquina, es podran fer servir remotament des d'una altra màquina. A continuació es fa una breu descripció del seu ús:

***En el servidor (qui comparteix el dispositiu USB):**

- + Instal·lar el paquet "linux-tools-common" (a Ubuntu) o "usbip" (a Fedora)
- + Llistar els dispositius disponibles a compartir: `sudo usbip list --local` Cal fixar-se, en la sortida d'aquesta comanda, en el codi "busid" mostrat (format per dos números separats per guió) corresponent al dispositiu que voldrem compartir
- + Carregar el mòdul USBIP necessari: `sudo modprobe usbip_host`
- + Posar en marxa (en segon pla) el servei de compartició : `sudo usbipd --daemon` Es posarà a escoltar al port 3240 (compte, per tant, amb el tallafocs!)
- + Compartir cada dispositiu que es desitja, indicant el seu "busid": `sudo usbip bind --busid=n-n`
Aquest dispositiu deixarà d'estar disponible en la màquina local, atenció

NOTA: Per deixar de compartir un dispositiu (i, per tant, tornar-lo a tenir disponible en local), cal executar la comanda `sudo usbip unbind --busid=n-n`

NOTA: Per saber quins dispositius s'estan compartint en un moment determinat, cal executar la comanda `sudo usbip list --remote=localhost`

***En el client (qui farà servir el dispositiu USB):**

- + Instal·lar el paquet "linux-tools-common" (a Ubuntu) o "usbip" (a Fedora)
- + Llistar els dispositius remots compartits per un determinat servidor: `sudo usbip list --remote=ip.serv` Cal fixar-se, en la sortida d'aquesta comanda, en el codi "busid" mostrat corresponent al dispositiu que voldrem utilitzar
- + Carregar el mòdul USBIP necessari: `sudo modprobe vhci_hcd`
- + "Endollar" el dispositiu desitjat a la màquina local: `sudo usbip attach --remote=ip.serv --busid=n-n`
Si tot ha funcionat correctament, aquest dispositiu apareixerà com un més al llistat mostrat per `lsusb`

NOTA: Per deixar d'utilitzar un dispositiu compartit cal executar la comanda `usbip detach --port=nº`, on el "número de port" caldrà haver-lo esbrinat abans amb la comanda `usbip port`

5.-PASSOS PREVIS: *Crea i afegeix, a una màquina virtual VirtualBox (Ubuntu o Fedora, Server o Desktop, tant és), un nou disc dur d'1GB de mida de tipus USB (és a dir, endollat sota el "controlador USB" disponible al panell de configuració de l'emmagatzematge en VirtualBox).

*Arrenca aquesta màquina i comprova que aquest nou disc dur apareix llistat a la sortida de la comanda `lsusb` Aquest dispositiu serà el que es compartirà amb altres màquines.

*Arrenca una segona màquina virtual VirtualBox (Ubuntu o Fedora, Server o Desktop, tant és) que estigui a la mateixa xarxa que la primera. Realitza els passos indicats als paràgrafs blaus anteriors per tal d'aconseguir que el disc dur físicament endollat a la primera màquina aparegui ara a la sortida de la comanda `lsusb` de la segona màquina (el podràs reconèixer pel seu "VendorID:ProductID" o la seva descripció)

* Després d'haver comprovat que la compartició de dispositius USB funciona correctament entre les dues màquines, recorda ara com es feia per carregar mòduls del kernel de forma automàtica en iniciar el sistema (ho vam veure en el PDF sobre els mòduls del kernel, precisament) i fes-ho a les dues màquines per tal de carregar automàticament en cada inici de cadascuna d'elles els mòduls "usbip_core", "usbip_host" i "vhci_hcd". Podràs comprovar que ho has fet correctament si observes, després de reiniciar-les, la sortida de la comanda `lsmod`. A partir d'aquí, , realitza els apartats següents:

a) Crea l'arxiu `"/etc/systemd/system/usbipd.service"` amb un contingut tal que tingui les següents característiques: que sempre es posi en marxa després d'haver arribat el sistema al target "network-online" (directiva `After=`), que sigui de tipus "simple" (directiva `Type=`), que executi la comanda `usbipd` (sense paràmetres!) en iniciar-se (directiva `ExecStart=`) i que, si s'habilita, que es faci al target "multi-user" (directiva `WantedBy=`) . ¿Per a què servirà, llavors, la comanda `sudo systemctl enable usbipd`? Executa-la i comprova que funciona correctament observant la sortida de `systemctl status usbipd`

b) Crea l'arxiu `"/etc/systemd/system/usbip-bind@.service"` amb un contingut tal que tingui les següents característiques: que sempre es posi en marxa després d'haver arribat el sistema al target `"network-online"` i iniciat el servei `"usbipd.service"` (directiva `After=`), que si no està aquest target o servei iniciats, que els iniciï llavors (directiva `Requires=`), que sigui de tipus `"simple"` (directiva `Type=`), que executi la comanda `usbip bind --busid %i` en iniciar-se (directiva `ExecStart=`) i la comanda `usbip unbind --busid %i` en aturar-se (directiva `ExecStop=`), que es reiniciï automàticament si s'atura per algun problema (directiva `Restart=`) i que "enganyi" al sistema fent-li creure que el servei roman encès tot i que la comanda indicada a `ExecStart=` és instantània (directiva `RemainAfterExit=`). ¿Per a què servirà, llavors, la comanda `sudo systemctl start usbip-bind@1-1` ? Executa-la (si cal, indicant un altre valor diferent del marcat amb color lila) i comprova que funciona correctament "endollant" el dispositiu indicat a la màquina client sense problemes.

NOTA: Pots trobar ajuda de com fer els apartats anteriors aquí: <https://aur.archlinux.org/packages/usbip-systemd>

D-BUS

6.- Usa el D-Feet per descobrir quin és el servei, objecte, interfície i mètode/propietat/senyal que:

- *Permet saber els usuaris que tenen alguna sessió iniciada actualment a la màquina
- *Tancar la sessió actual de l'usuari indicat

PISTA: El servei D-Bus demanat és proporcionat pel dimoni `Systemd-logind`

A partir de saber això, escriu un script que faci servir la comanda `busctl` per tancar, mitjançant algun tipus de bucle, totes les sessions que hi puguin haver obertes actualment dels diferents usuaris del sistema (ja siguin gràfiques, SSH, etc).

El propi dimoni D-Bus (a través del binari integrat `"dconf-service"`) proporciona un servei D-Bus de sessió anomenat `"ca.desrt.dconf"` la funció del qual és permetre escriptures (per part de qualsevol programa del sistema que l'invoqui) en la base de dades Dconf (és a dir, en la configuració del Gnome). Concretament, a través del seu objecte `"/ca/desrt/dconf/Writer/user"`, interfície `"ca.desrt.dconf.Writer"`, mètode `"Change()"`, el paràmetre del qual, però, és de tipus binari i, per tant, és difícil d'implementar directament

NOTA: Les lectures a la base de dades Dconf no involucren ni aquest ni cap altre servei.

7.-a) Obre dos terminals (no cal que siguis `"root"`); en un manté executant-se la comanda `busctl --user monitor ca.desrt.dconf | grep -B 1 -A 71 "ca.desrt.dconf.Writer Member=Change"` i en l'altra executa primer la comanda `gsettings set org.gnome.desktop.interface color-scheme prefer-dark` i després `gsettings set org.gnome.desktop.interface color-scheme prefer-light` Explica amb les teves pròpies paraules la sortida que apareix a la primera comanda, i en especial, el significat del conjunt de valors `"BYTE"` mostrats.

b) Tot i que no és necessari, es pot invocar un servei D-Bus per llegir dades concretes emmagatzemades a la base de dades Dconf. Concretament, per exemple, la comanda `busctl --user call org.freedesktop.impl.portal.desktop.gnome /org/freedesktop/portal/desktop org.freedesktop.impl.portal.Settings Read ss "org.freedesktop.appearance" "color-scheme"` obté la configuració actual del mode `"clar-fosc"` de l'escriptori. Obté aquesta mateixa informació (fent servir el mateix servei->objecte->interfície->mètode) però des del programa D-Feet en comptes de amb la comanda `busctl`

NOTA: Un altre element interessant de la interfície `org.freedesktop.impl.portalSettings` és la senya `SettingsChanged`

ENTORNS GRÀFICS

8.-a) Crea un accés directe (i situa'l al teu "dash") que tingui una icona personalitzada (te'n pots descarregar una qualsevol d'Internet si no la vols fer a mà) i que serveixi per mostrar la comanda *top* dins un terminal cada cop que li facis doble clic.

b) Fes que l'accés directe anterior tingui, a més, una opció disponible en el menú contextual que serveixi per executar el client *ssh* (amb un usuari i IP a usar predeterminats)

9.- Modifica la teva sessió d'escriptori per tal de què tingui un tema d'icones, un tema de cursors, un tema GTK i un tema Gnome Shell diferent de l'estàndard (tant se val com els obtinguis, bé dels repositoris oficials, de pàgines d'Internet, etc)

10.-a) Digues les comandes *gsettings* adients per configurar els següents aspectes (pots usar el *dconf-editor* per anar buscant i trobant la clau i el valor en qüestió en cada cas).

- *Obligar al Gedit (instal·la'l si no hi és) a no mostrar els números de línia ni tampoc la barra d'estat
- *Establir a 10s el temps automàtic de pas entre diapositives en el visor EoG a pantalla completa
- *Deixar de mostrar la finestra-avís "dels 60 segons" en tancar sessió/aturar el sistema.

b) A partir de la informació mostrada a <https://help.gnome.org/admin/system-admin-guide/stable/logout-automatic.html.en> configura Gnome (via *dconf-editor* o *gsettings* pel teu usuari particular, o bé, com diu l'enllaç anterior, amb arxius "profile" per tots els usuaris...com vulguis) per a què apagui la màquina al cap de 10 segons de no haver-se usat (suposant que la màquina està endollada al corrent elèctric).

c) Utilitza les descripcions mostrades al *dconf-editor* (o qualsevol altra font d'informació) per esbrinar (i explicar) per a què serveixen les següents claus del Dconf:

<i>/org/gnome/settings-daemon/plugins/power</i>	<i>idle-dim</i>
<i>/org/gnome/settings-daemon/plugins/power</i>	<i>idle-brightness</i>
<i>/org/gnome/desktop/session</i>	<i>idle-delay</i>
<i>/org/gnome/desktop/screensaver</i>	<i>lock-delay</i>
<i>/org/gnome/desktop/lockdown</i>	<i>disable-command-line</i>
<i>/org/gnome/desktop/lockdown</i>	<i>disable-save-to-disk</i>
<i>/org/gnome/desktop/lockdown</i>	<i>disable-printing</i>
<i>/org/gnome/desktop/lockdown</i>	<i>disable-log-out</i>
<i>/org/gnome/desktop/lockdown</i>	<i>disable-user-switching</i>
<i>/org/gnome/online-accounts</i>	<i>whitelisted-providers</i>

11.-a) Fes tot el necessari (mitjançant els "locks" de Gnome) per evitar que cap usuari pugui activar cap extensió de Gnome.

PISTA: El darrer apartat del darrer exercici del PDF sobre Gnome pot ser d'ajuda

b) Fes tot el necessari (mitjançant els "locks" de Gnome) per establir a tots els usuaris, de forma obligatòria i sense que el puguin canviar, un fons de pantalla que sigui un gradient de colors horitzontal des del negre fins el groc

PISTA: L'apartat aII) de l'exercici 7 del darrer exercici del PDF sobre Gnome pot ser d'ajuda

12.-a) Fes el necessari per canviar el fons de pantalla del Gdm i fer que no aparegui la llista d'usuaris (sinó que hagi d'escriure'l explícitament en una caixa de text)

b) Fes el necessari per iniciar, al final de l'arranc d'un sistema, una sessió Weston amb un usuari determinat automàticament.

Un sistema web-kiosk és un terminal que proporciona accés a Internet (a través d'un navegador a pantalla completa) i res més. Normalment s'utilitzen en llocs públics com ara aeroports, recepcions d'hotels, biblioteques, panells informatius de museus...per tenir un ràpid accés a pàgines web (les quals també es poden limitar a un cert conjunt) i prou. La idea és que l'usuari no pugui moure, redimensionar, minimitzar o desmaximitzar el navegador ni que pugui "escapar" a cap terminal (fent per exemple ALT+F2 o CTRL+ALT+F2, entre moltes altres opcions possibles) i, en general, que no pugui realitzar cap interacció o activitat que no estigui explícitament prevista.

Per construir un web-kiosk es pot prendre l'estratègia de partir d'un sistema gràfic complet (com un Gnome) i anar restringint opcions des de la seva pròpia configuració o mitjançant altres tècniques, o bé a l'inversa, partir d'un sistema de text i anar-li incorporant els components mínims necessaris (bàsicament un compositor Wayland) per poder arribar a visualitzar el navegador.

NOTA: Els sistemes kiosk no cal que mostrin un navegador: també podrien mostrar, a pantalla completa, una presentació de diapositives, un vídeo o una aplicació "ad-hoc", però el concepte sempre és el mateix: oferir una experiència limitada a l'usuari (eventualment interactiva fins a cert punt)

NOTA: Existeixen distribucions Linux especialitzades en implementar sistemes kiosk", per si no volem fer-ho "a mà". Exemples són Porteus (<https://porteus-kiosk.org>) o WebConverger (<https://webconverger.com>) D'altra banda, també existeixen compositors especialitzats en funcionar en aquest mode (per bastir a partir d'ells la resta de la pila d'aplicacions), com ara Cage (<https://github.com/Hjdskes/cage>; <https://www.hjdskes.nl/blog/cage-01>) o Mutter variant "Gnome-Kiosk" (<https://gitlab.gnome.org/GNOME/gnome-kiosk>), el qual l'estudiarem a continuació

13.-En una màquina virtual Fedora Workstation (no pot ser una altra) que tingui la seva tarja de xarxa en mode "adaptador pont" i el servidor SSH habilitat però el tallafocs no (per tal de poder seguir administrant el sistema quan funcioni com a kiosk), instal·la els paquets "gnome-kiosk" i "gnome-kiosk-script-session". A partir d'aquí:

a) Observa el contingut de l'arxiu `"/usr/share/wayland-sessions/gnome-kiosk-script-wayland.desktop"` (instal·lat pel 2º paquet). ¿Per a què serveix? ¿Què fa exactament la seva línia `Exec=` (mira l'apartat següent per tenir una pista)?

b) Observa el contingut de l'arxiu `"/usr/share/gnome-session/sessions/gnome-kiosk-script.session"` (instal·lat pel 2º paquet). ¿Per a què serveix? ¿Què fa exactament la seva línia `RequiredComponents=` (mira l'apartat següent per tenir una pista)?

c) Observa el contingut dels arxius `"/usr/share/applications/org.gnome.Kiosk.desktop"` i `"/usr/share/applications/org.gnome.Kiosk.Script.desktop"` (instal·lats pel 1º i 2º paquets, respectivament). ¿Per a què serveixen? ¿Què fan exactament les seves línies `Exec=` respectives (mira l'apartat següent per tenir una pista)?

d) Observa el contingut de l'arxiu `"/usr/bin/gnome-kiosk-script"` ¿Què fa? ¿Per a què serveix l'arxiu `"~/local/bin/gnome-kiosk-script"` (mira l'apartat següent per tenir una pista)?

NOTA: Altres fitxers implicats en l'inici de sessió en mode "kiosk" (però que no haurem de modificar mai, en principi, perquè funcionen a més baix nivell) són `"/usr/lib/systemd/user/org.gnome.Kiosk.target"` i `"/usr/lib/systemd/user/org.gnome.Kiosk@wayland.service"` (proporcionats pel primer paquet) i `"/usr/lib/systemd/user/gnome-session@gnome-kiosk-script.target.d/session.conf"` i `"/usr/lib/systemd/user/org.gnome.Kiosk.Script.service"` (proporcionats pel segon paquet)

e) Tanca sessió i ara sel·lecciona, d'entre les opcions que apareixen a la rodeta de la cantonada inferior dreta del GDM, l'opció "Kiosk script session (Wayland Display Server)" i inicia sessió amb l'usuari de sempre. ¿Què passa i per què? Un cop provat, reinicia la màquina i torna a triar l'opció "Gnome" per poder entrar a l'escriptori estàndard un altre cop

f) Elimina l'arxiu "~/local/bin/gnome-kiosk-script" (o bé, si no vols ser tan agressiu, comenta totes les línies del seu contingut) i fes que ara el contingut de l'arxiu "/usr/bin/gnome-kiosk-script" sigui el mostrat a continuació. Un cop fet això, entra de nou en l'opció "Kiosk script session (Wayland Display Server)". ¿Què passa ara i per què?

```
#!/bin/sh
firefox --kiosk --private-window --new-instance https://elpuig.xeill.net
sleep 1.0
exec "$@" "$@"
```

NOTA: Com podreu observar, la finestra que apareix a pantalla completa del navegador en aquest "mode kiosk" no es pot minimitzar i no mostra cap tipus de menú (tampoc contextual), barra de botons o barra de direccions ni res que no sigui la pròpia pàgina web mostrada; la idea és, tal com hem dit, que l'usuari només pugui navegar a través dels enllaços presents a les pròpies pàgines web i no en pugui sortir; per més informació pots consultar

<https://support.mozilla.org/en-US/kb/firefox-enterprise-kiosk-mode> D'altra banda, també iniciem el navegador en mode de "navegació privada"; en aquest cas, la idea és que així no es recordi cap contrasenya, ni historial de web visitades, etc; per més informació pots consultar <http://mzl.la/UpPfAP>

NOTA1: Malauradament, no existeix una forma senzilla de desactivar les combinacions de tecles pròpies del Firefox (veieu https://bugzilla.mozilla.org/show_bug.cgi?id=1635774, https://bugzilla.mozilla.org/show_bug.cgi?id=972662 o https://bugzilla.mozilla.org/show_bug.cgi?id=1267145 o <https://connect.mozilla.org/t5/ideas/customizable-hotkeys/idi-p/4979> entre altres). Això vol dir que l'usuari podrà pulsar combinacions com ara CTRL+N (per obrir una nova finestra del navegador) o CTRL+T (per obrir una nova pestanya) o CTRL+J o CTRL+U...i en general, qualsevol altra combinació de tecles que el faci sortir de la pàgina web visualitzada. Això és un gran problema que ara mateix no té sol·lució (es podria provar, no obstant, amb algun altre navegador, per exemple amb el Chromium fent així: *chromium-browser --kiosk --incognito https://elpuig.xeill.net ...*)

NOTA2: Menció especial hem de fer a les combinacions del Firefox CTRL+Q (tancar navegador) o CTRL+W (tancar darrera pestanya), les quals sí que funcionen igualment però que podem "desactivar" gràcies a la línia *exec "\$@" "\$@"* de l'script "gnome-kiosk-script", la qual el que fa és justament rellançar-se a ell mateix (i per tant, tornar a executar el Firefox de nou). El mateix passa amb la combinació de tecles general ALT+F4 (que tanca l'aplicació actualment en focus)

NOTA3: Afortunadament, excepte ALT+F4, molts dels "shortcuts" generals del Gnome sí que estan desactivats en aquest mode "kiosk". Per exemple, les combinacions de tecles ALT+F2, ALT+SPACE o ALT+TAB (que ens permetrien tancar la finestra del navegador i/o executar algun altre programa del sistema) no funcionen. Tampoc funciona la combinació SUPER+L (que bloqueja la pantalla). Tot i això, caldria fer un repàs més exhaustiu

NOTA: Una carència que té actualment el mode kiosk del Gnome és que no té incorporat el mecanisme (que sí té una sessió "estàndard" de Gnome) per detectar el tems d'inactivitat de la sessió. I això és un problema perquè mentre en una sessió "estàndard" es pot fer que passat un temps el sistema es suspengui o es tanqui la sessió automàticament (com ja hem vist en un exercici anterior), en el mode kiosk no. I això seria molt interessant per, per exemple, automàticament reiniciar la sessió de l'usuari per tornar a presentar de nou una pàgina web nova i fresca al nou visitant: ara mateix, tal com deixi la pàgina un visitant se la trobarà el següent.

g) Després de copiar totes les imatges existents en "/usr/share/backgrounds/gnome" a la carpeta "~/Imatges", fes que ara el contingut de l'arxiu "/usr/bin/gnome-kiosk-script" sigui el mostrat a continuació. Un cop fet això, entra de nou en l'opció "Kiosk script session (Wayland Display Server)". ¿Què passa ara i per què?

```
#!/bin/sh
if [[ ! "$(pidof eog)" ]]
then
    eog -s /home/usuari/Imatges
fi
sleep 1.0
exec "$@" "$@"
```

NOTA: Es pot modificar amb més detall el comportament i l'aparença del visor de Gnome canviant les claus desitjades que són presents sota la categoria "org.gnome.eog" (amb la comanda *gsettings* o el programa *dconf-editor* o mitjançant arxius de perfil, com es vulgui)

h) Acaba de personalitzar el sistema per a què sigui menys "hackeable" per part d'un usuari "massa llest". Concretament, recorda, de PDFs anteriors, com aconseguir les funcionalitats següents (i fes-ho):

*Que en arrencar no es vegi cap menú de GRUB ni cap tema Plymouth

*Que no es pugui entrar en cap terminal virtual (amb CTRL+ALT+Fx, al VirtualBox CTRL+dret+Fx)

Pista: per aconseguir això recorda que només cal modificar una directiva de configuració del servei Systemd-Logind

*Que el GDM no preguntí cap usuari ni contrasenya interactivament sinó que entri directament en la sessió de l'usuari que engega la sessió Kiosk (en aquest sentit, aquest usuari ha d'estar configurat adientment en `"/var/lib/AccountsService/users/nomusuari"`)

NOTA: Aquest mateix exercici es podria haver fet també en un sistema Server, només que caldria haver fet inicialment el següent: instal·lar els paquets "gdm" (el qual arrossegarà tots els paquets gràfics necessaris com a dependències), "firefox" i "gnome-kiosk-script-session" (que instal·la "gnome-kiosk" com a dependència) i, tot seguit, establir com a "target" per defecte el mode gràfic, així: `sudo systemctl set-default graphical.target`. I ja està.