

FreeDesktop i Gnome (I)

Freedesktop (<http://www.freedesktop.org>) és un projecte que fomenta la interoperabilitat i una base de tecnologia comuna pels diferents entorns d'escriptoris lliures que existeixen (Gnome, KDE, LXDE, XFCE, etc), de manera que les diferències entre aquests no provoquin incompatibilitats a nivell d'usuari. Els següents exercicis profunditzaran en les configuracions Freedesktop. Com que precisament el que es pretén és que aquestes siguin independents de l'entorn d'escriptori utilitzat, els exercicis es poden realitzar en teoria en qualsevol escriptori compatible amb Freedesktop.

Variables "XDG"

El projecte Freedesktop defineix a l'estàndard <https://specifications.freedesktop.org/basedir-spec> un conjunt de carpetes amb un determinat tipus de contingut; per ser compatible amb aquest estàndard qualsevol escriptori, sigui quin sigui, les haurà d'implementar. La ruta concreta d'aquestes carpetes ve donada per unes variables d'entorn concretes (el valor de les quals, tot i que es podria, normalment no modificarem perquè el que ve per defecte ja està bé), que són:

NOTA: Com que inicialment Freedesktop va ser conegut com a "X Desktop Group", aquestes variables (i també moltes de les eines desenvolupades per aquest projecte), porten al seu nom el prefix "xdg".

\$XDG_DATA_HOME: Defineix la ruta del directori base a partir del qual s'emmagatzemen fitxers contenint dades específiques de l'usuari en qüestió. Si no existeix o està buida, s'usa per defecte el valor "**\$HOME/.local/share**". Es pot utilitzar per guardar plugins descarregats per l'usuari, marcadors, emails, bases de dades creades per programes d'usuari, etc

\$XDG_CONFIG_HOME: Defineix la ruta del directori base a partir del qual s'emmagatzemen fitxers de configuració específics de l'usuari en qüestió. Si no existeix o està buida, s'usa per defecte el valor "**\$HOME/.config**". Es pot utilitzar per guardar la configuració dels programes específica per l'usuari actual.

\$XDG_CACHE_HOME: Defineix la ruta del directori base a partir del qual s'emmagatzemen fitxers contenint dades específiques no essencials de l'usuari en qüestió. Si no existeix o està buida, s'usa per defecte el valor "**\$HOME/.cache**". Es pot utilitzar per guardar "thumbnails" generats pel gestor de fitxers, la catxé del navegador, etc. Els programes haurien de poder continuar funcionant sense problemes si s'esborra aquesta carpeta.

\$XDG_RUNTIME_DIR: Defineix la ruta del directori base a partir del qual s'emmagatzemen fitxers temporals, no essencials i específics de l'usuari en qüestió, normalment utilitzats per la comunicació IPC dins la seva sessió, com ara "sockets" (com el que estableix el canal D-Bus d'usuari) "named pipes", etc. Normalment està definida amb el valor de "**/run/user/UID**". Aquest directori només existeix mentre existeixi la sessió actual de l'usuari (es crea en fer el primer login i es destrueix en fer el darrer logout; si un usuari iniciés diverses sessions alhora veuria la mateixa carpeta).

\$XDG_DATA_DIRS: Defineix la preferència (per ordre, i separats per ":") de les rutes dels directoris base on buscar fitxers de dades a més d'on indiqui \$XDG_DATA_HOME. Si no existeix o està buida, s'usa per defecte el valor "**/usr/local/share:/usr/share**". Es pot utilitzar per guardar els plugins o temes usats per tots els usuaris.

\$XDG_CONFIG_DIRS: Defineix la preferència (per ordre, i separats per ":") de les rutes dels directoris base on buscar fitxers de configuració a més d'on indiqui \$XDG_CONFIG_HOME. Si no existeix o està buida, s'usa per defecte el valor "**/etc/xdg**". Es pot utilitzar per guardar configuració genèrica dels programes que podrà ser sobreescrita per la de cada usuari en particular.

D'altra banda, el mòdul "pam_systemd", utilitzat en tots els sistemes Systemd en el procés d'autenticar usuaris de cara a obtenir una sessió (normalment d'escriptori però no té perquè), treballant conjuntament amb el servei "systemd-logind", el qual registra i manté aquesta sessió ja autenticada, afegeixen unes quantes variables més relacionades amb les sessions d'usuari (veure *man pam_systemd*):

\$XDG_SESSION_DESKTOP : Indica el nom de l'escriptori iniciat per la sessió d'usuari actual. Un valor pot ser "gnome" o "kde". Poden existir altres variables que valguin el mateix, com ara **\$XDG_CURRENT_DESKTOP** , **\$DESKTOP_SESSION** o **\$GDMSESSION** (en el cas d'usar "gdm"), depenent de l'escriptori utilitzat.

\$XDG_SESSION_TYPE : Indica el servidor gràfic sobre el qual s'implementa la sessió d'usuari actual. Pot valer "wayland" (el més normal), "x11" (obsolet), "tty" (pels terminals virtuals o SSH, sense servidor gràfic) o "unspecified".

\$XDG_SESSION_CLASS: Indica el tipus de sessió actualment activa. Pot valer "user" (per indicar que s'està en un entorn d'escriptori), "greeter" (o en la pantalla del display manager), "lock-screen" (o en la pantalla del salvapantalles bloquejant) o "background" (o en la pantalla de salvapantalles no bloquejant).

1.-a) Executa `printenv | grep "^XDG"` a la màquina real i dedueix, a partir dels valors d'algunes de les variables d'entorn mostrades, en quin escriptori tens oberta la sessió i sobre quin servidor gràfic aquest escriptori està funcionant

b) ¿Què representa que hi ha d'haver sota la carpeta indicada a la variable `$XDG_DATA_HOME`? ¿I sota la carpeta indicada a la variable `$XDG_CONFIG_HOME`? Si aquestes variables no estiguessin definides, ¿quin valor per defecte es prendria per les rutes de les dues carpetes anteriors?

Arxius `.desktop`

Un arxiu `*.desktop` és un fitxer de text que funciona com un “accés directe” a una aplicació o carpeta. A continuació es mostra un exemple (les línies en negreta són les més comunes i importants; per més informació consulteu <https://specifications.freedesktop.org/desktop-entry-spec>).

NOTA: Cal respectar les majúscules i minúscules

NOTA: L'ordre de les directives és irrellevant

NOTA: Les línies començant per # són comentaris

[Desktop Entry]

#Nom que apareixerà sota la icona de l'accés directe.

#Es permeten els símbols `\s` -espai-, `\n` -salt de línia-, `\t` -tabulador-

Name=Hola

#Nom que es mostrarà si l'idioma del sistema no és l'idioma per defecte però sí és l'especificat entre [].

#L'idioma del sistema en cada moment ve especificat per les variables `LC_MESSAGES` ó `LANG`.

Name[ca]=Hola

Name[it]=Ciao

#Comentari que apareixerà dins un tooltip emergent en activar l'accés directe

Comment=lalala

Comment[it]=lelele

#Aquesta opció pot tenir dos valors: *Application* si l'accés directe és d'un programa o *Link* si és d'una carpeta/web

Type=Application

#L'opció següent indica el nom (sense extensió) de la imatge que serà la icona de l'accés directe.

#Aquestes imatges es troben dins d'alguna subcarpeta (ja veurem) dins de “`/usr/share/icons`” i han de ser PNG o SVG

#També es pot indicar la ruta completa d'una imatge que estigui guardada en un altre lloc.

Icon=nomimatge

#Només té sentit si `Type` és `Application`. Indica si el programa s'executa en un terminal (`true`) o no (`false`)

Terminal=false

#Només té sentit si `Type` és `Application`. Indica la carpeta de treball de l'aplicació

Path=/ruta/carpeta/de/treball

#Només té sentit si `Type` és `Application`. Indica la ruta de l'executable a posar en marxa.

#Si només s'escriu el nom (sense ruta) es buscarà al `PATH`

Exec=/ruta/absoluta/ejecutable

#Només té sentit si `Type` és `Application`. Indica (separats per ;) els tipus MIME que pot obrir l'aplicació

MimeType=tipus;mime;soportats;per;l'aplicacio;

#Només té sentit si `Type` és `Application`. Indica (separades per ;) les categories a les que pertany l'aplicació.

#Aquestes categories es troben definides a l'estàndar `Freedesktop` (<http://standards.freedesktop.org/menu-spec/latest/apa.html>).

#Exemples són: `Audio`, `Video`, `Development`, `Education`, `Game`, `Graphics`, `Network`, `Office`, `Settings`, `Utility`

Categories=Una;UnaAtra;UnaMés;

#Només té sentit si `Type` és `Link`. Indica la ruta local d'una carpeta o fitxer (`file:///ruta/carpetaOFitxer`) o una URL (`http://ruta/fitxer`)

URL=file:///ruta/fitxer

#Serveix (si val `true`) per deshabilitar l'accés directe (com si no existís l'arxiu `*.desktop`)

Hidden=true

#Serveix (si val `true`) per ocultar l'accés directe dels diferents menús, llistes, desplegable, etc

#però continua “habilitat” per totes les tasques (com l'associació a tipus MIME, etc).

NoDisplay=true

#Serveix per indicar els escriptoris on es mostrarà l'accés directe i cap més (també està `NotShownIn`)

OnlyShowIn=KDE; GNOME; XFCE; Unity;

#Podem inventar qualsevol altra directiva dins d'un arxiu `.desktop` si comença per “X-”.

X-DirectivaPròpia=lalala

La línia *Exec* dels arxius *.desktop admet una sèrie de paràmetres a tenir en compte:

%f : Representa un únic nom de fitxer; usat quan es deixa anar el fitxer sobre una icona
%F : Representa una llista de fitxers; usat per aplicacions que poden obrir diversos fitxers de cop
%u : Representa una única URL (Http/Ftp/ruta local) senzilla
%U : Representa una llista d'URL
%i : Valor de la icona que apareix a la línia <i>Icon=</i>
%c : Nom traduït de l'aplicació que apareix a la línia <i>Name=</i>

Tots els arxius *.desktop comuns del sistema han d'ubicar-se en una sola carpeta central per poder ser reconeguts correctament: "\$XDG_DATA_DIRS/applications" i tots els arxius *.desktop particulars d'un usuari han d'ubicar-se dins la carpeta "\$XDG_DATA_HOME/applications"

2.-a) Crea a la carpeta "~/local/share/applications" de la màquina real un arxiu anomenat "pepe.desktop" que hauràs d'editar "a mà" (amb el Gedit) de la forma adient per a què funcioni com a accés directe apuntant a l'executable del LibreOffice Writer i que tingui la mateixa icona que el firefox (la qual s'anomena "firefox"); el nom i el comentari poden ser els que vulguis.

NOTA: Si no saps quin és l'executable del Writer, pots executar la comanda `ps -ef | grep "writer"` mentre el tens en marxa o, de forma alternativa, pots fer `dpkg -L libreoffice-writer` (a Ubuntu) o `rpm -ql libreoffice-writer` (on per saber el nom del paquet a inspeccionar -en aquest cas, "libreoffice-writer"- podríem haver fet abans `apt/df search libreoffice`, per exemple).

NOTA: En realitat podríem haver aprofitat com a plantilla pel nostre arxiu ".desktop" l'arxiu ".desktop" que ja està instal·lat a "/usr/share/applications" corresponent al LibreOffice Writer, però fent-ho així perd la gràcia.

b) Comprova que l'hagis escrit correctament amb la comanda `desktop-file-validate /ruta/arxiu.desktop`

NOTA: Aquesta comanda mostra per pantalla els errors trobats (per tant, no mostrarà cap missatge si arxiu.desktop està ben escrit). Errors molt comuns són escriure espais en blanc darrera els valors de les directives o oblidar-se el ; final a les directives Categories i MimeType

Una de les conseqüències de tenir ubicats els arxius *.desktop en alguna de les dues carpetes anteriors en comptes de qualsevol altra és que automàticament apareixen al buscador d'aplicacions del Gnome, (a més d'aparèixer a la llista general d'aplicacions general, ordenat alfabèticament pel seu nom). La paraula de referència per realitzar les recerques és el nom de l'executable o bé el nom indicat a la directiva *Name=*. Veiem-ho al següent apartat:

c) Troba l'accés directe creat a l'exercici anterior al buscador d'aplicacions del Gnome3 (fent servir o bé el valor de la directiva *Name=* o bé el nom de l'executable indicat a *Exec=*) o a la llista general d'aplicacions. Executa'l.

NOTA: També és possible "executar" un arxiu "*.desktop" des del terminal amb la comanda `gtk-launch nomArxiu` (sense l'extensió ".desktop")

d) Afegeix l'accés directe trobat a l'apartat anterior al Dash (això es fa simplement arrossegant-lo i deixant-lo anar, o bé triant l'opció "Afegeix als preferits" del menú que apareix en clicar-hi a sobre amb el botó dret)

NOTA:El "dash" és la barra horitzontal que apareix al Gnome3 en la zona inferior de la pantalla mostrant les icones de les aplicacions favorites i de les que s'estan executant en aquest moment

Si cliques amb el botó dret sobre una icona qualsevol del Dash veuràs unes poques opcions (eliminar l'accés directe del Dash, obrir una instància nova del programa i poca cosa més). Es poden afegir més opcions personalitzades si es modifica convenientment l'arxiu *.desktop corresponent. Concretament, sota la secció [*Desktop Entry*] s'ha d'afegir la directiva *Actions*, el valor de la qual ha de ser una llista de paraules (finalitzades amb ;) que identificaran les diferents opcions a mostrar; i seguidament, s'ha d'afegir una secció anomenada [*Desktop Action paraulaIdentificadora*] per cada opció a mostrar, sota la qual s'haurà d'especificar la directiva *Name=* i la directiva *Exec=*. Veiem-ho al següent apartat:

3.-a) Fes que l'accés directe incorporat al Dash a l'exercici anterior ofereixi tres opcions extres visibles al menú contextual que permetin obrir el Calc, l'Impress i el Draw. Per fer això, afegeix al final de l'arxiu "pepe.desktop" creat a l'exercici anterior el següent contingut i reinicia la sessió d'usuari:

```
Actions=Calcul;Presentacio;Dibuix;
[Desktop Action Calcul]
Name=Full de càlcul
Exec=/usr/bin/oocalc
[Desktop Action Presentacio]
Name=Presentació
Exec=/usr/bin/ooimpress
[Desktop Action Dibuix]
Name=Dibuix vectorial
Exec=/usr/bin/oodraw
```

b) Observa el contingut de l'arxiu "/usr/share/applications/org.gnome.Terminal.desktop" i digues quin és el paràmetre (utilitzat a una de les "Actions" allà predefinides) que, afegit a l'executable *gnome-terminal*, obre directament les seves preferències. ¿I el que obre una finestra nova? ¿Quin seria el paràmetre (no utilitzat a l'arxiu *.desktop ...consulta *man gnome-terminal*) per obrir una pestanya nova? ¿I amb un títol determinat?

bII) Copia l'arxiu "org.gnome.Terminal.desktop" anterior dins de la carpeta "~/.local/share/applications" (de manera que tingui preferència la còpia sobre l'original) i edita la còpia de la següent manera. ¿Què fa la nova opció que apareixerà al menú contextual de l'accés directe al terminal?

- * Afegeix la cadena *Top*; al final de la línia *Actions=* existent
- * Afegeix al final del fitxer les línies següents:

```
[Desktop Action Top]
Name=Top
Exec=gnome-terminal --maximize -- top -d 2
```

c) Arrenca una màquina virtual que tingui ja instal·lat l'escriptori Gnome (Ubuntu o Fedora, és igual...si no tens a mà cap, llegeix la primera nota inferior) i crea-hi un nou arxiu *.desktop dins de "/usr/share/applications" (anomenat "lamp.desktop" per exemple) amb el següent contingut. Després d'incloure'l al Dash, ¿per a què serviria? (suposant que els paquets "apache2" -"httpd" a Fedora- i "mariadb-server" estiguin instal·lats ja). Comprova la seva utilitat observant la sortida de *systemctl status apache2/httpd* i *systemctl status mariadb* després de triar les diferents opcions que mostra l'accés directe

```
[Desktop Entry]
Name=LAMP
Comment=Open services of your LAMP server
Type=Application
Exec=bash -c "sudo systemctl start apache2 && sudo systemctl start mariadb" #A Fedora és "httpd"
Terminal=true #Veure https://itectec.com/ubuntu/ubuntu-unable-to-start-xampp-using-sudo-in-desktop-entry/
Icon=emblem-web
Actions=Stop;Restart;
[Desktop Action Stop]
Name=Stop
Exec=bash -c "sudo systemctl stop apache2 && sudo systemctl stop mariadb" #A Fedora és "httpd"
[Desktop Action Restart]
Name=Restart
Exec=bash -c "sudo systemctl restart apache2 && sudo systemctl restart mariadb" #A Fedora és "httpd"
```

NOTA: En el cas de tenir un sistema Server, s'hi pot instal·lar l'escriptori Gnome a sobre. Concretament, a Fedora es pot fer executant la comanda *sudo dnf group install "Fedora Workstation"* i a Ubuntu es pot fer el mateix executant la comanda *sudo apt install ubuntu-desktop*

NOTA: A la línia *Exec=* hem hagut d'utilitzar la sintaxi *bash -c "comandes"* perquè per si sola aquesta línia només admet l'execució d'una sola comanda, i aquí nosaltres estem executant dues concatenades pels símbols *&&*; per fer que això es pugui entendre hem d'executar un terminal Bash que sigui capaç d'interpretar-ho

cII) Recordant el que hem vist a l'assignatura de Seguretat, ¿què caldria fer per a què l'accés directe anterior no preguntés cap contrasenya en el moment d'executar el *sudo*?

Tipus MIME

Un tipus MIME identifica el format d'un fitxer. Els gestors de fitxers (entre altres programes, com els navegadors) fan servir el tipus MIME d'un fitxer per associar-li la icona corresponent i per decidir quines accions realitzar sobre ell quan l'usuari els sel·lecciona (és a dir, si detecten que és un fitxer de text obriran un editor de text, si detecten que és un fitxer de vídeo obriran un reproductor multimèdia, etc). Cal dir que aquests programes solen detectar el tipus MIME d'un fitxer no a partir de la seva extensió (que pot ser canviada molt fàcilment) sinó a partir de la seva capçalera binària.

Els tipus MIME són identificadors compostos d'un tipus de nivell superior seguit (i separat pel caràcter "/") d'un identificador de subtipus. Per exemple: el tipus MIME d'una imatge jpeg és "image/jpeg", el d'un fitxer de text pla és "text/plain", etc. El tipus superior és una categorització general del contingut del fitxer, mentre que el subtipus identifica específicament el format del fitxer. Els tipus de nivell superior són estandaritzats per l'organització internacional anomenada IANA i actualment són deu: *application, audio, font, example, image, message, model, multipart, text, video*. A més, existeixen molts altres subtipus per a cadascun dels tipus anteriors. Per més informació, consulta <http://www.iana.org/assignments/media-types>.

NOTA: Els subtipus MIME prefixats amb "x-" o també amb "vnd-" (de "vendor specific") són vàlids però no estandaritzats

Tal com hem dit, els tipus MIME serveixen, en un entorn d'escriptori, per associar automàticament cada fitxer (amb un tipus MIME determinat) a la llista d'aplicacions que el podran obrir (visibles a l'opció "Obrir amb..." del seu menú contextual) i, més específicament, a l'aplicació concreta dins d'aquestes que serà usada per defecte (fent-hi "doble clic"). Explicarem això en detall:

1.-Per decidir quines aplicacions seran capaces d'obrir fitxers d'un determinat tipus MIME (dels que estan reconeguts a la base de dades) i que, per tant, apareixeran a la seva opció "Obrir amb...") només cal afegir el tipus MIME desitjat a la línia *MimeType=* del fitxer *.desktop associat amb l'aplicació i seguidament executar la comanda *update-desktop-database*. (o, equivalentment, reiniciar la sessió d'usuari)

2.-Per establir quina d'entre les aplicacions anteriors serà la per defecte, al fitxer "**mimeapps.list**" (ubicat tant dins de "\$XDG_DATA_DIRS/applications" i/o "\$XDG_CONFIG_DIRS" com dins de "\$XDG_DATA_HOME/applications" i/o "\$XDG_CONFIG_HOME") s'ha d'editar la línia desitjada (totes tenen l'estructura *tipusMime=arxiu.desktop*, excepte la capçalera [*Added Associations*]). Concretament, s'ha d'escriure l'arxiu *.desktop associat al programa que volem executar per defecte en obrir tots els arxius del tipus MIME escollit (es poden especificar varis arxius *.desktop ordenats segons preferència i separats per ;). Un cop fet això, seguidament s'ha d'executar obligatòriament la comanda *update-desktop-database* (o reiniciar la sessió d'usuari).

NOTA: Pot ser que existeixi a la mateixa carpeta que "mimeapps.list" un altre arxiu semblant anomenat "defaults.list", però aquest és una alternativa obsoleta. També pot existir un fitxer anomenat "nomEscriptori-mimeapps.list", en aquest cas, tindria preferència sobre "mimeapps.list". En qualsevol cas, tot aquest conjunt de fitxers, incloent el format intern del seu contingut, estan definits a l'estàndard <https://specifications.freedesktop.org/mime-apps-spec>

NOTA: En executar la comanda *update-desktop-database* qualsevol dels canvis que apareguin en algun dels fitxers anteriors es reflectirà a l'arxiu "**mimeinfo.cache**" (ubicat normalment dins de "\$XDG_DATA_DIRS/applications" i/o "\$XDG_DATA_HOME/applications"), el qual és l'arxiu -únic- on realment, per qüestions de velocitat i eficiència, el sistema va a consultar les associacions tipus MIME<> programes cada cop que ho necessiti (però que nosaltres no l'haurem de modificar mai directament ja que serà "matxacat" cada cop que s'executi *update-desktop-database*, com hem dit).

No obstant, els passos anteriors es poden realitzar d'una forma molt més còmoda i senzilla sense haver de modificar cap fitxer a mà. En concret, la comanda *gio mime* (disponible al paquet "glib2", propi de l'ecosistema Gnome) és la que farem servir per conèixer les aplicacions associades als fitxers d'un determinat tipus MIME i també per establir-ne l'aplicació per defecte. Bàsicament:

gio mime tipus/mime serveix per saber quina és l'aplicació associada per defecte al tipus especificat (és a dir, consulta l'arxiu "mimeapps.list")

gio mime tipus/mime arxiu.desktop serveix per establir l'aplicació indicada (via arxiu **.desktop* ubicat a */usr/share/applications*) com a aplicació per defecte del tipus especificat (és a dir, modifica l'arxiu *"mimeapps.list"* i executa *update-desktop-database*).

D'altra banda, per saber quin tipus MIME té un determinat fitxer podem executar *gio info arxiu.qualsevol*, concretament veurem aquesta dada a l'opció *"standard::content-type"*. Una altra comanda que ens dona la mateixa informació però és independent de l'escriptori és *file -i [-b] arxiu.qualsevol*

NOTA: Tal com es pot veure, la comanda *gio info* a més ofereix moltíssima informació addicional sobre el fitxer, com ara els permisos, les dates (de creació, modificació, accés...), el tamany, el propietari, etc.

NOTA: La base de dades dels tipus MIME reconeguts pel sistema (que en realitat és un conjunt de fitxers XML) es troba dins de la carpeta *"\$XDG_DATA_DIRS/mime"* i se sol crear durant la instal·lació del sistema mitjançant la instal·lació del paquet *"shared-mime-info"* (tot i que qualsevol paquet d'una aplicació particular pot instal·lar els seus propis tipus MIME associats particulars; també hi podria haver una altra base de dades, complementària pels tipus MIME reconeguts per un usuari en particular, a *"\$XDG_DATA_HOME/mime"*; en cas d'existir aquesta base de dades de l'usuari tindria precedència sobre la del sistema global en cas de conflicte). En qualsevol cas, dins d'aquestes carpetes hi ha una subcarpeta per cada tipus de nivell superior, dins de les quals per cada subtipus hi ha el seu fitxer XML corresponent. Aquesta estructura de carpetes, així com les etiquetes concretes que són necessàries per definir els diferents tipus MIME en aquests arxius XML, estan definits als estàndards <https://specifications.freedesktop.org/shared-mime-info-spec>

NOTA: Una altra comanda similar a *gio mime* és *xdg-mime* (disponible al paquet *"xdg-utils"*). La diferència entre una i altra és que la primera és particular per l'escriptori Gnome i la segona és general per qualsevol tipus d'escriptori (i per tant, és més versàtil). Un exemples d'ús són els següents:

xdg-mime query filetype /ruta/arxiu : retorna el tipus MIME de l'arxiu indicat (similar a *gio info /ruta/arxiu*)
xdg-mime query default tipus/mime : similar a *gio mime tipus/mime*
xdg-mime default arxiu.desktop tipus/mime : similar a *gio mime tipus/mime arxiu.desktop*

4.-a) Fes que totes imatges PNG del sistema s'obrin per defecte per l'accés directe creat a l'apartat a) de l'exercici 2 i comprova-ho. Pista: Per aconseguir això primer hauràs de fer servir la comanda *gio info* per conèixer el tipus MIME d'aquestes imatges i seguidament la comanda *gio mime* per realitzar la vinculació d'aquest tipus amb el fitxer **.desktop* desitjat.

aII) Tant a Ubuntu com a Fedora, la comanda *gio mime* anterior afecta a l'arxiu *"~/config/mimeapps.list"*. Modifica manualment aquest arxiu per tornar a tenir com a aplicació per defecte dels arxius *"image/png"* el programa original: el visor de fotos de Gnome (l'accés del qual s'anomena *"org.gnome.eog.desktop"*). Comprova-ho.

b) Esbrina quin programa obre per defecte *"/usr/share/man/man1/gzip.1.gz"* (un altre cop amb les comandes *gio info* i *gio mime*). A partir d'aquí, utilitza la comanda *gio mime* per fer que els arxius *".gz"* s'obrin automàticament amb el reproductor de vídeo Totem (has de trobar abans el nom del seu arxiu *"*.desktop"* corresponent). Comprova-ho amb *gio mime* també i fent doble clic sobre algun arxiu d'aquest tipus

c) Raona el significat de la sortida de *grep "org.gnome.Nautilus" /usr/share/applications/mimeapps.list* (a Fedora) o *grep "org.gnome.Nautilus" /usr/share/applications/defaults.list* (a Ubuntu) i concretament, el de l'associació establerta amb el tipus MIME *"inode/directory"*. Pista: Nautilus és el gestor de fitxers de Gnome i File-Roller és el (des)compressor de fitxers de Gnome

El tipus MIME *"x-scheme-handler"* es refereix no a un tipus de fitxer sinó a un tipus d'enllaç (que començarà pel subtipus MIME corresponent: *http://*, *mailto://*, etc) i per tant, l'associació establerta amb un arxiu *"*.desktop"* indicarà el programa que serà executat no en clicar sobre un fitxer sinó sobre un enllaç del tipus en qüestió (sempre que ho fem fora d'un navegador, ja que aquests programes ja tenen la seva pròpia associació amb tipus d'enllaços independent de la del sistema).

d) Havent llegit el paràgraf blau anterior, raona per què els arxius *"*.desktop"* que apareixen a la sortida de *grep "x-scheme-handler" /usr/share/applications/mimeapps.list* (a Fedora) o *grep "x-scheme-handler" /usr/share/applications/defaults.list* (a Ubuntu) són els que són.

dII) Fes que tots els enllaços que comencin per *https://* s'obrin a partir d'ara pel Gedit. Comprova-ho creant un nou document *"odt"* (amb el Libreoffice Writer) que contingui una URL que comenci per *"https://"* i tot seguit clica-hi a sobre. ¿Què passa?

Carpetes d'escriptori especials

La carpeta "**autostart**" (ubicada a `$XDG_CONFIG_DIRS` per tots els usuaris o a `$XDG_CONFIG_HOME` per un usuari en concret) conté arxius `*.desktop` que apunten als programes que s'executaran automàticament cada cop que s'iniciï una sessió d'escriptori. Allà pots trobar per exemple l'arxiu corresponent al programa `xdg-user-dirs-update`, precisament (entre d'altres arxius molt interessants). Aquest comportament ve definit per l'estàndard <https://specifications.freedesktop.org/autostart-spec>

5.-a) Copia l'arxiu `*.desktop` corresponent al Firefox de `"/usr/share/applications"` a `"~/config/autostart"` (si aquesta darrera carpeta no existeix l'hauràs de crear primer) i reinicia sessió. ¿Què passa?

b) Llegeix la pàgina *man systemd-xdg-autostart-generator* i digues què fa aquest "generator" Systemd (ubicat a `"/usr/lib/systemd/user-generators"`) i què hi té a veure amb ell el target `"xdg-desktop-autostart.target"` (amb una eventual línia `Wants=` dins de la seva configuració).

L'especificació <https://specifications.freedesktop.org/trash-spec/trashspec-latest.html> descriu el funcionament de la "paperera" als diferents entorns d'escriptori. Bàsicament, explica que cada usuari manté una paperera pròpia dins la carpeta `"$XDG_DATA_HOME/Trash"`, dins de la qual han d'aparèixer com a mínim dues subcarpetes: "**files**" (que conté els fitxers esborrats) i "**info**" (que conté fitxers amb l'extensió `.trashinfo`, cadascun dels quals es correspon a un dels fitxers esborrats que es troben a "files"). Els fitxers `.trashinfo` contenen dues dades: la ruta original del fitxer esborrat (per poder-lo restaurar si s'escau) i la data de l'esborrament (per a què el Nautilus pugui fer ordenacions amb aquesta dada).

6.-a) Després de crear un fitxer qualsevol (fent, per exemple, `echo "Hola" > a.txt`), executa la comanda `gio trash a.txt` i tot seguit executa `ls ~/.local/share/Trash/files` ¿Què veus? ¿I si executes `gio list trash://`?

b) No existeix una comanda "gio" específica per restaurar fitxers de la paperera, però es pot fer com a "sucedani" això: `gio move trash:///a.txt $(grep "Path" ~/.local/share/Trash/info/a.txt.trashinfo | cut -f2 -d"=")`. Prova-ho. ¿Què fa exactament la comanda anterior?

c) Executa la comanda `gio trash --empty` ¿Què passa si tornes a executar llavors `gio list trash://`?

NOTA: Una alternativa per gestionar la paperera des del terminal d'una forma completa és utilitzar el paquet "trash-cli"

El concepte de "**portapapers**" també és propi dels entorns gràfics (de fet, és una proposta d'especificació de Freedesktop: <https://www.freedesktop.org/wiki/Specifications/ClipboardsWiki>) tot i que existeixen comandes que poden gestionar-lo des d'un terminal (sempre i quan aquest, això sí, s'executi dins d'una sessió gràfica -és a dir, que no sigui un terminal virtual-), com són per exemple `wl-copy` i `wl-paste`, pertanyents al paquet "wl-clipboard" (<https://github.com/bugaevc/wl-clipboard>)

7.- a) Instal·la el paquet "wl-clipboard" i, en un terminal funcionant sobre una sessió en el teu escriptori, executa les següents comandes i digues què fan:

```
wl-copy "Hola què tal?"    i tot seguit  wl-paste
ls /etc | wl-copy          i tot seguit  wl-paste > clipboard.txt
wl-copy < /ruta/foto.png   i tot seguit  wl-paste > /ruta/foto2.png && wl-copy -c
```

NOTA: El prefix "wl" de les comandes anteriors prové de Wayland, programa del qual en parlarem més endavant

Nautilus

Altres comandes de la família "gio", a més de les ja vistes en exercicis anteriors (com *gio mime*, *gio info* o *gio trash*), són *gio cat*, *gio list* (similar a *ls*), *gio mkdir*, *gio copy*, *gio move*, *gio monitor* (per detectar modificacions en els fitxers o carpetes indicats), *gio save* (per redireccionar la sortida d'una altra comanda), *gio set* (per establir els atributs mostrats per *gio info*), *gio mount*, *gio rename*, *gio tree* (similar a la comanda *tree*), *gio open* (obre el fitxer/url especificat amb el programa que estigui configurat per defecte segons el seu tipus MIME) o *gio remove* (similar a *rm*), entre altres.

8.-a) Executa les següents comandes (a la màquina real) i digues què aconseguim en cadascuna i al final (pots consultar *man gio* per més referència):

```
gio mkdir carpeta
echo "Hola" | gio save carpeta/a.txt
gio list carpeta (o gio tree carpeta)
gio cat carpeta/a.txt
gio open carpeta/a.txt
gio info carpeta/a.txt
gio info -a "time:.*" carpeta/a.txt
gio info -a "metadata:.*" carpeta/a.txt
gio set -t string carpeta/a.txt "metadata::nota" "Compra llet!"
gio info -a "metadata:.*" carpeta/a.txt
gio move carpeta/a.txt b.txt
gio info -a "metadata:.*" b.txt
gio copy b.txt c.txt
gio info -a "metadata:.*" c.txt
gio remove b.txt c.txt carpeta
```

La gràcia d'emprar les comandes "gio" en lloc de les seves "alternatives clàssiques" (*ls*, *cp*, *mv*, *rm*...) és que les primeres permeten realitzar la mateixa tasca tant sobre fitxers o carpetes locals (usant la URL *file:///ruta/fitxOCarp*) com sobre fitxers o carpetes remotes mitjançant diversos protocols possibles d'accés a emmagatzematge remot, com ara SSH (amb la URL *ssh://usuari@ipServidor/ruta/fitxOCarp*), WebDAVS (amb URL *davs://usuari@ipServidor/ruta/fitxOCarp*), SMB (amb URL *smb://usuari@ipServidor/nomShare*) o altres. Concretament, la llista de protocols de xarxa suportats es troba disponible a <https://wiki.gnome.org/Projects/gvfs/backends> i la de les corresponents URLs ("schemes") acceptades per cadascun es troba a <https://wiki.gnome.org/Projects/gvfs/schemes>

NOTA: Aquesta funcionalitat "d'entendre" URLs ("schemes") de xarxa per accedir a recursos (carpetes i/o fitxers) remots tal com si fossin locals és possible gràcies a l'ús per part de Gnome (i, en particular d'aplicacions que en formen part, com *gio* o el gestor de fitxers Nautilus) de la llibreria GIO (<https://docs.gtk.org/gio>). Aquesta llibreria presenta als desenvolupadors de les aplicacions que la vulguin fer servir una API que proporciona un sistema de fitxers virtual (anomenat "GVFS") la qual implementa una abstracció que fa que, per l'aplicació en qüestió sigui indistint accedir a un recurs local o remot perquè per ella s'accedirà de la mateixa manera. Els anomenats "backends" GVFS són els "drivers" específics que permeten accedir a cada tipus de recurs en concret, però dels seus detalls no s'ha de preocupar l'aplicació.

b) Obre una finestra del Nautilus (a la màquina real) i pulsa **CTRL+L** per a què la barra de botons visible a la zona superior de la finestra es converteixi en una barra de direccions on hi puguem escriure. A partir d'aquí:

*¿Què veus si escrius en aquesta barra de direccions la URL "*network:///*"? ¿I "*computer:///*"? ¿I "*trash:///*"?

*¿Què veus si escrius en aquesta barra de direccions la URL "*ssh://usuari@ip.dalgun.serv.SSH*" ?

NOTA: El mateix podries haver fet clicant a l'opció "Altres ubicacions" -> "Connecta al servidor" que apareix a la barra lateral esquerra del Nautilus. En qualsevol cas, fent-ho de qualsevol d'aquestes dues maneres, per tal de "desmuntar" la carpeta remota només caldrà clicar sobre el botonet d'"expulsar" que apareixerà a la barra lateral del Nautilus. Els punts de muntatge establerts estaran disponibles per totes les aplicacions que s'executin (a la vegada fins i tot, si el "backend" ho permet) dins del context de la sessió d'usuari actual.

NOTA: Si no vols que se't preguntï la contrasenya, es podria incloure a la URL d'aquesta manera: *ssh://usuari:contrasenya@ip.serv.SSH* Igualment, es pot indicar explícitament la ruta de la carpeta de destí on s'hi vol anar (en lloc de la carpeta personal de l'usuari utilitzat, que és el destí per defecte) d'aquesta manera: *ssh://usuari@ip.serv.SSH/ruta/carpeta*

9.-a) Obre una carpeta qualsevol amb el Nautilus a la màquina real i canvia la vista de graella (per defecte) a llista. ¿Què has de fer per a què a aquesta vista aparegui la columna mostrant el tipus de tots els fitxers i carpetes (obtingut a partir del seu tipus MIME)?

b) ¿Com pots afegir l'opció d'esborrar fitxers definitivament (sense utilitzar la papelera) al menú contextual? I l'opció de crear enllaços simbòlics? Pista: consulta les preferències del Nautilus.

NOTA: L'acció d'esborrar definitivament, en qualsevol cas, es pot realitzar sempre pulsant les tecles **CTRL+SUPR**

c) Evita que es generin thumbnails ("miniatures") dels fitxers. Pista: consulta preferències del Nautilus

d) Fes que a la vista de graella es mostri la mida de cada fitxer. Pista: consulta preferències del Nautilus

e) Canvia la icona d'un fitxer o carpeta en concret. Pista: consulta l'opció "Propietats" del fitxer

f) ¿Què hauries de fer dins de l'opció de "Propietats" d'un fitxer que sigui un script Bash per donar-li permisos d'execució?

10.-a) Ja sabem que per ocultar un arxiu o carpeta només cal precedir el seu nom d'un punt. Comprova-ho creant (amb *touch*, per exemple) dins de la teva carpeta personal (a la màquina real) un fitxer anomenat ".fitxerocult.txt" i tot seguit obrint aquesta carpeta amb el Nautilus: no l'hauràs de veure. ¿Què passa, però, si pulses CTRL+H? ¿I si ho tornes a pulsar?

b) Crea ara un nou arxiu anomenat ".hidden" a la teva carpeta personal que tingui el següent contingut ...

Música

Imatges

i tot seguit obre la teva carpeta personal amb el Nautilus, ¿què passa? ¿I si pulses **CTRL+H**? ¿Quin avantatge li veus a aquest mètode d'ocultar carpetes o fitxers en front del vist a l'apartat anterior?

NOTA: Aquest truc només funciona per ocultar els fitxers o carpetes ubicats dins de la mateixa carpeta on hi sigui l'arxiu ".hidden". Això vol dir que per cada carpeta on volguem aplicar-lo haurem de crear allà l'arxiu ".hidden" corresponent

NOTA: L'arxiu ".hidden" admet comentar línies (i per tant, "desocultar" l'element corresponent) precedint-les de "#"

NOTA: L'arxiu ".hidden", però, no funciona per comandes de terminal (com *ls*, per exemple)

c) ¿Quin és el contingut de l'arxiu "~/.config/gtk-3.0/bookmarks"? ¿Què passarà si, al final d'aquest arxiu, afegeixes, per exemple, la línia *file:///home/asix2/.config pepito*?

cII) Situa't, amb el Nautilus, dins de, per exemple, la carpeta "/home/asix2/.cache" i tot seguit pulsa **CTRL+D**. ¿Què passa? ¿Quin és ara el contingut de l'arxiu "~/.config/gtk-3.0/bookmarks"?

d) Escriu a la barra de direccions del Nautilus la URL "*recent://*". ¿Què veus? Tot seguit, esborra el contingut del fitxer "*~/local/share/recently-used.xbel*" ¿Què veus ara a la finestra anterior? Ara obre un arxiu de text qualsevol, grava alguna cosa en ell i observa ara quin és el contingut del fitxer "recently-used.xbel" i el de la finestra anterior. ¿Què veus ara?

NOTA: A l'estàndard <https://freedesktop.org/wiki/Specifications/desktop-bookmark-spec> ve definit el format d'aquest arxiu, anomenat XBEL, que no és més que una variant d'XML

11.-Crea al teu sistema real un fitxer "*.odt" que contingui un títol més un paràgraf i un altre fitxer "*.ods" que contingui alguns valor diferents dins d'algunes cel·les qualssevol, i guarda'ls tots dos (amb el nom que vulguis) dins de la carpeta "~/.Plantilles" (o si no hi és, dins de la carpeta indicada per la comanda *xdg-user-dir TEMPLATES*). Observa quina nova opció apareix sota l'opció "Document nou" al menú contextual mostrat en qualsevol zona buida d'una finestra oberta amb el Nautilus

NOTA: També podem organitzar les plantilles en subcarpetes (per temes, ús,...) si s'hi creen dins de la carpeta "~/.Plantilles"

12.-a) Guarda el següent codi dins d'un fitxer (anomenat com vulguis) ubicat a la carpeta "`~/local/share/nautilus/scripts`" del teu sistema real i dóna-li permisos d'execució (`chmod +x fitxer`):

```
#!/bin/bash
gnome-terminal --working-directory=$NAUTILUS_SCRIPT_SELECTED_FILE_PATHS
```

Observa que, si cliques amb el botó dret sobre algun fitxer o carpeta visible dins d'una finestra del Nautilus, apareix al menú contextual una nova opció anomenada "Scripts" on es llista l'script anterior. Prova'l ¿Per a què serveix? (llegeix nota següent per saber-ho).

NOTA: Els scripts del Nautilus poden ser scripts Bash completament estàndard, i per tant, poden realitzar qualsevol tasca que es podria fer en un terminal (encara que també poden ser scripts Python, o Perl, etc). S'ha de tenir en compte, però, que en aquests scripts podem utilitzar unes variables d'entorn que ens poden ser útils com ara:

NAUTILUS_SCRIPT_SELECTED_FILE_PATHS conté la ruta dels fitxers/carpetes actualment sel.leccionats, **NAUTILUS_SCRIPT_SELECTED_URI** conté la ruta començant per "`file://`" dels fitxers/carpetes sel.leccionats i **NAUTILUS_SCRIPT_CURRENT_URI** conté la ruta començant per "`file://`" de la carpeta on es troben els fitxers/carpetes actualment sel.leccionats.

NOTA: Existeixen "extensions" del Nautilus ja empaquetades i llestes per descarregar i utilitzar que afegeixen opcions concretes al seu menú contextual (a mode d'"scripts ja prefabricats" però en format binari; concretament, en forma de llibreria dinàmica). Algunes d'aquestes extensions es troben disponibles als repositoris oficials d'Ubuntu o Fedora (es poden buscar amb `apt/dnf search nautilus`); en tot cas, les que tinguem instal·lades es trobaran sota la carpeta "`/usr/lib64/nautilus/extensions-3.0`"

b) ¿Què faria el següent script (consulta *man oowriter* per saber per a què serveix el seu paràmetre `-p`)?

```
#!/bin/bash
oowriter -p $NAUTILUS_SCRIPT_SELECTED_FILE_PATHS
```

c) Arrenca una màquina virtual qualsevol que tingui escriptori Gnome i instal·la-hi el paquet "ImageMagick". Consulta la seva pàgina web (<https://legacy.imagemagick.org> i també <https://legacy.imagemagick.org/script/command-line-tools.php>) per descobrir per a què serveix aquesta suite de comandes i, en especial, la comanda *convert* que en forma part. Un cop fet això, guarda el següent codi dins d'un fitxer (anomenat com vulguis) ubicat a la carpeta "`~/local/share/nautilus/scripts`" d'aquesta màquina virtual i dóna-li permisos d'execució (`chmod +x fitxer`):

```
#!/bin/bash
echo "$NAUTILUS_SCRIPT_SELECTED_FILE_PATHS" | xargs -I{} convert "{}" "{}.jpg"
```

NOTA: En aquest exemple la comanda *xargs* és necessària perquè la comanda *convert* no admet més d'una ruta com a fitxer d'entrada ni tampoc més d'una ruta com a fitxer de sortida, així que cal executar-la cada vegada per cada fotografia sel.leccionada. Precisament és això el que fa la comanda *xargs*: recollir cada línia (separada de la resta de línies per un salt de línia) rebuda de la canonada i assumir que el contingut d'aquesta línia (que sí que podrà tenir espais en blanc) equival al símbol "{}"; a partir d'aquí, executarà la comanda *convert* per cadascun dels valors "{}" recollits de la canonada

cII) Copia (sempre en aquesta màquina virtual) totes les fotos que hi ha ubicades sota la carpeta "`/usr/share/backgrounds/gnome`" a la teva carpeta personal (per tal de què hi tinguis permisos d'escriptura) i tot seguit sel.lecciona totes aquestes còpies i executa l'script anterior des del menú contextual del Nautilus. ¿Què passa?

13.-a) Obre una carpeta qualsevol amb el Nautilus (a la teva màquina real) i observa la xuleta de dreceres de teclat que tens disponible a l'opció "Dreceres del teclat" del menú general (el que mostra les tres línies horitzontals, a la cantonada superior dreta de la finestra) i digues, a partir d'ella, quina és la combinació de tecles per obrir una pestanya nova del Nautilus. ¿I una finestra nova?

b) Amb una carpeta qualsevol oberta, ¿què fa i com funciona la combinació CTRL+S? ¿I F9? ¿I F2 (amb un fitxer o carpeta seleccionada)?

c) Relaciona la llista d'accions següents amb la combinació de tecles utilitzada per realitzar-les

NOTA: Podeu conèixer (i canviar!) més combinacions si aneu al panell de control de Gnome, subpanell Teclat->Dreceres de teclat. En tot cas, moltes d'elles estan documentades a <https://help.gnome.org/users/gnome-help/stable/shell-keyboard-shortcuts.html.en> i també a <https://help.gnome.org/users/gnome-help/stable/keyboard-nav.html.en>

PRINT
SHIFT + PRINT
ALT + PRINT
ALT + TAB (i seguidament soltant ALT)
ALT + TAB + cursors (i seguidament soltant ALT)
ALT + ESC (i seguidament soltant ALT)
ALT + \ (i seguidament soltant ALT)
ALT + F4
ALT + F2
SUPER + L
ALT + SPACE
ALT + F7
ALT + F8
SUPER + A
SUPER (o movent el el ratolí a la cantonada superior esquerra de la pantalla)
CTRL + ALT + Cursors Left/Right
CTRL + SHIFT + ALT + R
SUPER + M
CTRL + clic a una icona del Dash ja en execució (o clic amb el botó del mig)
SUPER + Cursors Left/Right
SUPER + Cursors Up/Down

- *Capturar tota la pantalla
- *Capturar només la finestra activa
- *Sel.leccionar l'àrea a capturar
- *Passar a un altre programa que estigui obert en aquest moment, de forma seqüencial i cíclica.
- *Fer el mateix que el punt anterior però usant els cursors per triar el programa en qüestió
- *Fer el mateix que el punt anterior però directament, sense triar el programa sinó anant-hi
- *Passar a una altra finestra del mateix programa
- *Tancar la finestra actual
- *Executar una comanda
- *Bloquejar l'escriptori
- *Mostrar el menú principal de la finestra activa
- *Moure la finestra activa del Nautilus
- *Redimensionar la finestra activa del Nautilus
- *Mostrar la vista d'aplicacions
- *Mostrar/Ocultar la pantalla "Overview"
- *Anar, dins de la pantalla "Overview", a un altre "workspace"
- *Iniciar/Aturar una captura en vídeo (en guardarà en format "webm" a la carpeta personal)
- *Mostrar/Ocultar àrea de notifikacions
- *Obrir una nova instància ("finestra") del programa en comptes d'activar la que ja estigui executant-se
- *Mostrar la finestra activa només en una meitat (vertical) de la pantalla
- *Maximitzar/restaurar la finestra activa