

Systemd (III)

Gestió dels noms locals de la màquina

El nom "static" és el nom de la màquina tradicional, escollit per l'usuari a la instal·lació del sistema i guardat al fitxer **"/etc/hostname"**; només s'utilitza per activitats locals a la màquina. El nom "transient" és un nom dinàmic que pot rebre's a través de la xarxa (via DHCP, mDNS,...) durant l'execució del sistema; si no s'indica explícitament, s'assimila al nom "static". El nom "pretty" és una petita descripció de la màquina per presentar-la a l'usuari local.

NOTA: Un hostname "transient" pot canviar segons sigui necessari per evitar col·lisions de noms. Per exemple, si anomenau el vostre equip "bob" però ja hi ha un altre sistema amb aquest nom a la xarxa, el vostre hostname de la xarxa es pot convertir en "bob-1"

A més de l'arxiu **"/etc/hostname"** hi ha altres arxius importants, com ara **"/etc/os-release"** (el qual emmagatzema el nom de la distribució, la seva versió, la seva pàgina web, etc), **"/etc/machine-id"** (el qual conté un identificador únic per cada màquina generat durant el procés d'instal·lació del sistema - concretament mitjançant la invocació al binari *systemd-machine-id-setup*- i que és utilitzat per vèries eines, com ara D-Bus, Journald i altres) i **"/etc/machine-info"** (el qual conté informació extra sobre la màquina com ara el seus noms "transient" i "pretty", etc).

Les comandes més importants relacionades amb aquests fitxers són:

hostnamectl [status] : Mostra els tres tipus de nom de màquina a més d'altra informació emmagatzemada als arxius **"/etc/os-release"**, **"/etc/machine-id"** i **"/etc/machine-info"** (i **"/proc/cmdline"**)

(sudo) hostnamectl set-hostname nom [--transient | --pretty | --static] : Canvia el nom de la màquina. Si no s'indica cap paràmetre, s'aplica el nom indicat a tots tres tipus: "transient", "pretty" i "static"; si s'indica algun en concret, només es canvia aquest.

NOTA: Una manera alternativa de canviar el nom "transient" és mitjançant *sudo systemctl kernel.hostname=nom*

La comanda *hostnamectl* en realitat no és més que un client D-Bus que realitza les peticions mitjançant aquesta via (ja siguin de consulta o d'actualització de dades) al servei D-Bus ofert pel dimoni *systemd-hostnamed*, el qual es posa en marxa només quan és rep la petició (no està tota l'estona funcionant). Això significa que la mateixa funció que podem fer amb la comanda *hostnamectl* es pot realitzar mitjançant qualsevol eina que permeti fer crides al canal D-Bus, com ara *dbus-send*, *gdbus* o la pròpia de Systemd, *busctl*. Concretament, recordem que el funcionament bàsic d'aquesta darrera comanda és aquest:

```
busctl [--system | --user ] call nom.Servei /nom/Objecte nom.Interfície mètode [signatures paràmetres]
busctl [--system | --user ] get-property nom.Servei /nom/Objecte nom.Interfície propietat
busctl [--system | --user ] set-property nom.Servei /nom/Objecte nom.Interfície propietat signatura valor
busctl [--system | --user ] emit /nom/objecte nom.Interfície senyal signatura paràmetres
busctl [--system | --user ] tree [nom.Servei]
busctl [--system | --user ] introspect nom.Servei /nom/Objecte [nom.Interfície]
```

on "signatura" vol dir el tipus de paràmetre indicats, que pot indicar-se amb una "i" si és de tipus sencer, una "s" si és de tipus "string", "b" si és booleà, "ai" si és un array de sencers, "as" si ho és de cadenes, etc.

EXERCICIS (tots es realitzaran en una màquina virtual Ubuntu o Fedora, Destkop o Server, és igual):

1.-a) Digues quina línia d'informació de les mostrades per la comanda *hostnamectl* és igual a la que sortida que mostra cadascuna de les comandes de Linux presentades a continuació:

```
uname -m
uname -r
grep CPE_NAME /etc/os-release
grep PRETTY_NAME /etc/os-release
cat /etc/machine-id
cat /etc/hostname
journalctl --list-boots | tail -n 1
systemd-detect-virt
```

b) Canvia el nom de la màquina amb la comanda *hostnamectl ...* i confirma que ho hagi fet bé observant tres coses: el (nou) contingut de l'arxiu *"/etc/hostname"*, la (nova) sortida de la comanda *hostnamectl* i el (nou) prompt del terminal després de tancar-lo i tornar-lo a obrir.

c) Fes servir la comanda *busctl introspect* per esbrinar les propietats i mètodes existents a les interfícies oferides per l'objecte *"/org/freedesktop/hostname1"* pertanyent al servei D-Bus *"org.freedesktop.hostname1"* (oferit pel dimoni *systemd-hostnamed*). A partir de la informació obtinguda, realitza llavors una crida amb *busctl call* al mètode adient (el qual conté un paràmetre de tipus booleà al que podràs assignar-li el valor que vulguis) per tornar a canviar el nom de la màquina (concretament, el de tipus "static"). Comprova que ho hagi fet bé observant les mateixes tres coses que a l'apartat anterior

Gestió de les "locales"

Una "locale" és una configuració regional definida pel sistema que inclou principalment el llenguatge amb què es mostraran els missatges a pantalla i el mapejat del teclat però també el format dels números, hores, dates, telèfons, monedes, unitats de mesura, etc. Aquesta configuració es global per tots els programes/serveis/usuaris el sistema, encara que cada programa/servei/usuari en particular podria tenir la seva pròpia configuració regional definida a les seves preferències; en aquest cas la configuració particular sobreescriria la global.

***Per esbrinar la "locale" actualment definida es pot executar la comanda *localectl [status]* , la qual simplement mostra el contingut dels arxius *"/etc/locale.conf"* (a Ubuntu és *"/etc/default/locale"*) i *"/etc/vconsole.conf"*. Allà dins hi poden haver establertes diferents variables d'entorn (amb el nom *LC_XXX*) que defineixen cadascuna un determinat aspecte de la configuració regional (el que hem comentat: dates, nombres, etc) i la variable *LANG*, la qual representa la configuració genèrica.**

NOTA: Per conèixer el significat concret de cada variable cal consultar la pàgina *man 7 locale* Per conèixer els valors i formats concrets que admet una determinada variable es pot consultar la pàgina del manual *man 5 locale* o bé executar *locale -ck LC_NOMVARIABLE* Per conèixer els estàndards internacionals ISO emprats per establir valors vàlids de locales, es pot consultar la pàgina del manual *man 3 setlocale* (allà es pot veure, per exemple, que els codis d'idioma estan definits a l'estàndard ISO 639 i els territoris a l'ISO 3166; en tot cas, es pot obtenir una llista de totes les locales existents executant la comanda *locale -a*)

***Per veure totes les locales instal·lades al sistema cal fer el següent: *localectl list-locales*.**

NOTA: A Ubuntu l'arxiu *"/etc/locale.gen"* conté la llista de "locales" que, un cop descomentades allà i executada la comanda *sudo locale-gen*, apareixeran com a instal·lades. Això no serveix de res, però, mentre no s'instal·lin els paquets d'idioma associats (veure següent punt), procés en el qual es realitza automàticament aquestes passes

***Per establir una "locale" temporalment (mentre el terminal actual estigui obert) cal fer el següent:**
export LANG=fr_FR.utf8

Per establir una "locale" permanentment (gravant-se a l'arxiu *"/etc/locale.conf"*) cal executar: (*sudo*) *localectl set-locale LANG=en_GB.utf8* (o *LC_*=...*). **El canvi es veurà al següent inici de sessió.*

NOTA: També es pot modificar a mà l'arxiu `~/config/locale.conf` per sobreescrivre la configuració general de la "locale" del sistema per una particular de l'usuari en qüestió.

NOTA: També és possible indicar una eventual "locale" particular en un determinat arranc del sistema mitjançant paràmetres del kernel. Concretament amb el paràmetre `locale.LANG=` i/o els paràmetres més específics (`locale.LC_NUMERIC=`, `locale.LC_TIME=`, `locale.LC_MONETARY=`, `locale.LC_MESSAGES=`, `locale.LC_PAPER=`, `locale.LC_ADDRESS=`, `locale.LC_TELEPHONE=`, `locale.LC_MEASUREMENT=`, etc, etc)

*Per instal·lar una nova "locale" (per exemple, el francès, "fr_FR"), a Ubuntu es pot executar la comanda `check-language-support -l fr_FR`; aquesta comanda mostra tots els paquets que caldrien instal·lar per disposar d'aquesta "locale" plenament incorporada al sistema. A Fedora només cal instal·lar el metapaquet adient a la "locale" desitjada, el qual té el nom de "`langpacks-XX`" (on "XX" representa la "locale" escollida)...això fa que s'executi automàticament la comanda `localedef` (l'equivalent a la comanda `locale-gen` d'Ubuntu)

*Per veure tots els mapejats de teclat disponibles es pot fer: `localectl list-keymaps`

*Per establir temporalment un mapejat de teclat es pot fer `sudo loadkeys fr`. No obstant, cal aclarir que la comanda anterior **només funciona en terminals virtuals** degut a què en entorns gràfics la configuració del teclat ve determinada per la indicada al panell de control de l'escriptori.

*Per establir permanentment un mapejat de teclat (gravant-se a l'arxiu `/etc/vconsole.conf`) es pot fer: `sudo localectl set-keymap fr`. No obstant, cal aclarir que la comanda anterior **només funciona en terminals virtuals** degut a què en entorns gràfics la configuració del teclat ve determinada per la indicada al panell de control de l'escriptori.

NOTA: Existeix el paràmetre del kernel `vconsole.keymap=...` que permet sobreescrivre durant l'arranc del sistema la configuració escrita a `/etc/vconsole.conf`

La comanda `localectl` en realitat no és més que un client D-Bus que realitza les peticions mitjançant aquesta via (ja siguin de consulta o d'actualització de dades) al servei D-Bus ofert pel dimoni `systemd-locale`, el qual es posa en marxa només quan és rep la petició (no està tota l'estona funcionant). Això significa que la mateixa funció que podem fer amb la comanda `localectl` es pot realitzar mitjançant qualsevol eina que permeti fer crides al canal D-Bus, com ara `dbus-send`, `gdbus` o la pròpia de Systemd, `busctl`

NOTA: Existeix també el servei "systemd-vconsole-setup", que és invocat automàticament per "systemd-locale" cada cop que es canvia la configuració del teclat dels terminals virtuals (i sempre per "systemd-udev" en el 1r arranc del sistema). Internament aquest servei utilitza les comandes de baix nivell `setfont` i `loadkeys` (aquesta última és qui s'encarrega, de fet, i tal com hem vist, de trobar i carregar de forma efectiva els mapeigs concrets del teclat, els quals venen proporcionats en forma de fitxers ".map" -sota la carpeta `/usr/lib/kbd/keymaps-` gràcies al paquet "kbd", dependència de Systemd). Per obtenir més informació, veieu https://wiki.archlinux.org/index.php/Linux_console/Keyboard_configuration i *man keymaps*

EXERCICIS:

1.-a) Instal·la la "locale" del francès (de França) i tot seguit activa aquesta "locale" de forma temporal en un terminal. ¿Què passa si executes tot seguit, i en aquell mateix terminal, la comanda `date`? ¿I la comanda `cp` sense paràmetres? ¿Quina continua sent, però, la sortida de `localectl` (i el contingut de `/etc/locale.conf`)?

b) Ara estableix la "locale" del francès com la "locale" permanent del sistema. ¿Què passa ara si, un cop reiniciada la sessió d'usuari, executes la comanda `date`? ¿I la comanda `cp` sense paràmetres? ¿Quina és ara, efectivament, la sortida de la comanda `localectl` (i, per tant, el contingut de l'arxiu `/etc/locale.conf`)?

c) Canvia el mapejat del teclat de forma permanent per a què sigui l'anglès. Vés a un terminal virtual i comprova que allà sí ha canviat. ¿Quin és, efectivament, el contingut de l'arxiu `/etc/vconsole.conf`?

d) Sabent que el servei D-Bus "org.freedesktop.locale1" ofereix l'objecte `/org/freedesktop/locale1`, realitza una crida amb `busctl call` al mètode adient per canviar la "locale" del sistema a `es_ES`. Comprova-ho.

NOTA: El mètode en qüestió té dos paràmetres, el primer dels quals és un array de cadenes. Recorda que per indicar els elements d'un array qualsevol amb `busctl` cal fer-ho amb la sintaxis `nºposicio valor`. És a dir, si volguéssim indicar, per exemple, dos elements d'un array de cadenes, caldria escriure això: `1 "UnElement" 2 "UnAltreElement"`

Gestió del temps

Els sistemes operatius distingeixen dos tipus de rellotges diferents:

*Un rellotge de temps real (RTC), sovint referit com a "hardware clock" perquè sol ser un circuit integrat a la placa base, el qual és completament independent de l'estat del sistema operatiu i funciona fins i tot quan la computadora està apagada gràcies a l'alimentació elèctrica proporcionada per la pila botó.

*Un rellotge de sistema, sovint referit com a "software clock", el qual és mantingut pel kernel. El seu valor inicial es basa en l'obtingut del RTC però una vegada el sistema ha arrencat, aquest pot ser manipulat independentment, ja sigui de forma manual o bé a través del protocol NTP. No obstant, degut a què a cada reinici es torna a "sincronitzar" amb el RTC, el "software clock" només és autònom durant el funcionament del sistema.

Al sistemes Linux moderns, en comptes de la comanda *date*, per conèixer l'hora i data del sistema és més complet (ja que ens mostra tant l'hora i data local com la UTC -de "Universal Time", la del "meridià de Greenwich"- i la RTC -que ha de ser sempre igual a la UTC-, a més de mostrar també la zona horària) fer servir la comanda *timedatectl [status]*

Per canviar ja sigui la data, l'hora o les dues coses, tant a nivell de "software clock" com de "hardware clock" a la vegada, la comanda a executar és (*sudo*) *timedatectl set-time "nºany-nºmes-nºdia nºhora:nºmin:nºseg"* Compte, però, perquè la comanda anterior no funcionarà si hi ha algun servei NTP habilitat a la màquina (veure l'explicació de més endavant). D'altra banda, cal tenir en compte que l'hora que s'ha d'indicar ha de ser sempre la LOCAL, ja que *timedatectl* ja en deduirà la corresponent hora UTC convenientment per tal d'aplicar-la al sistema RTC. En aquest sentit, cal tenir la precaució de comprovar (amb *timedatectl status*) que l'hora RTC estigui configurada per establir-se efectivament amb l'hora UTC (com hauria de ser) i no pas amb l'hora local (circumstància aquesta gens desitjable perquè provoca moltes inconsistències). Si no fos el cas, caldrà executar la comanda *timedatectl set-local-rtc no* per tal d'"arreglar" aquest aspecte.

NOTA: La comanda *date* també es podria fer servir per modificar la data i hora (així: *date MMDDhhmmYY.ss*) però només ho fa a nivell de "software clock". Alternativament, tenim la comanda *hwclock*, la qual pot modificar la data i hora del RTC (així *hwclock --set --date="YYYY-MM-DD hh:mm:ss"*). Però no farem servir cap d'aquestes dues ja que amb *timedatectl* ja en tenim prou i és una eina més integrada amb Systemd i D-Bus.

La comanda *timedatectl* en realitat no és més que un client D-Bus que realitza les peticions mitjançant aquesta via (ja siguin de consulta o d'actualització de dades) al servei D-Bus ofert pel dimoni *systemd-timedated*, el qual es posa en marxa només quan és rep la petició (no està tota l'estona funcionant). Això significa que la mateixa funció que podem fer amb la comanda *timedatectl* es pot realitzar mitjançant qualsevol eina que permeti fer crides al canal D-Bus, com ara *dbus-send*, *gdbus* o la pròpia de Systemd, *busctl*

La zona horària del sistema està indicada a un arxiu binari anomenat **"/etc/localtime"**, que no és més que un accés directe a **"/usr/share/zoneinfo/Europe/Madrid"** (o a l'arxiu associat a la zona indicada durant la instal·lació del sistema; també podria ser **"/usr/share/zoneinfo/UTC"**). Per saber totes les zones horàries possibles no cal, però, fer cap llistat manual del contingut de la carpeta **"/usr/share/zoneinfo"** perquè la següent comanda ja ens ho fa automàticament: *timedatectl list-timezones*

Canviar de zona horària, doncs, seria tan senzill com fer que **"/etc/localtime"** apuntes a un altre arxiu que estigui dins de **"/usr/share/zoneinfo"** (aquests arxius s'instal·len amb el paquet "tzdata", mantingut per la IANA). Això es pot fer "manualment" amb la comanda *ln -s* o bé, més fàcil, amb la comanda: *timedatectl set-timezone "Europe/Madrid"* (no cal ser "root").

La comanda *timedatectl* també és capaç de sincronitzar automàticament el nostre rellotge (tant software com hardware, recordem) amb servidors NTPs públics d'Internet (com per exemple els llistats a <https://www.ntppool.org>) si hem executat en algun moment la comanda (*sudo*) *timedatectl set-ntp on*. Aquesta comanda el que fa és iniciar automàticament (i habilitar pels propers reinicis) el client NTP per defecte del sistema, el qual pot ser, en sistemes moderns, o bé el dimoni *chronyd* (un paquet de tercers però que sol venir ja instal·lat i configurat a la majoria de distribucions grans) o bé, si aquest no estigués instal·lat, el dimoni *systemd-timesyncd* (propi de Systemd).

NOTA: En el cas de què hi hagi instal·lat *chronyd*, és aquest el dimoni que la comanda *timedatectl set-ntp on* activa/inicia: només en el cas de què sols hi hagi instal·lat *systemd-timesyncd*, llavors serà aquest el dimoni activat/iniciat. Igualment, si s'executés *timedatectl set-ntp off*, s'aturaria/deshabilitaria el dimoni *chronyd* i si aquest no hi fos, el dimoni *systemd-timesyncd*.

Als exercicis s'estudiarà com configurar el dimoni *chronyd*. En el cas de fer servir *systemd-timesyncd*, però, els servidors NTP concrets a contactar es determinen de forma molt senzilla a partir del següent mecanisme:

*Els servidors llistats a la directiva *NTP=* de l'arxiu de configuració global ***"/etc/systemd/timesyncd.conf"*** . Es pot indicar tant noms DNS com adreces IP.

*Els servidors llistats estàticament a la directiva *NTP=* de l'arxiu ".network" corresponent a una tarja de xarxa concreta o bé als obtinguts de la configuració dinàmica rebuda mitjançant DHCP. En qualsevol cas, aquesta configuració tindria preferència sobre la global

NOTA: Si no s'adquireix cap informació sobre quin servidor NTP usar després de completar aquests passos, s'utilitzaran els servidors NTP definits a la directiva *FallbackNTP* = del fitxer "timesyncd.conf".

En resum: si hi ha connexió a Internet, el servei *systemd-timesyncd/chronyd* sincronitzarà automàticament el rellotge del sistema local amb un servidor NTP remot i el servei *systemd-timesyncd* s'assegurarà llavors que el nou rellotge del sistema estigui sincronitzat immediatament amb el rellotge hardware (RTC).

EXERCICIS:

1.-a) Intenta canviar l'hora manualment amb la comanda *sudo timedatectl set-time ...* tenint la sincronització NTP activada (per defecte a Ubuntu i Fedora això ja està així) en una màquina virtual qualsevol que tinguis a mà ¿Pots?

aII) Ara desactiva aquesta sincronització executant la comanda *sudo timedatectl set-ntp off* . ¿En quin estat es troba tot seguit el servei *systemd-timesyncd* (o *chronyd* si tens aquest programa instal·lat, com passa a Fedora)? Torna a intentar canviar l'hora manualment. ¿Pots ara?

aIII) Torna a activar la sincronització NTP ¿En quin estat es troba tot seguit el servei *systemd-timesyncd* (o *chronyd*)? ¿Totes les hores (local, UTC i RTC) s'han actualitzat?

aIV) Espera't 11 minuts (ves fent altres apartats i exercicis) i torna a comprovar llavors les hores de nou. ¿Totes les hores (local, UTC i RTC) estan actualitzades ja?

b) Sabent que el servei D-Bus "org.freedesktop.timedate1" ofereix l'objecte "/org/freedesktop/timedate1", el qual conté la interfície "org.freedesktop.timedate1" que ofereix la propietat "Timezone", executa la comanda *busctl get-property* adient per conèixer el valor d'aquesta propietat.

c) Canvia la zona horària del sistema a la de Los Angeles. ¿Què passa amb les hores locals, UTC i RTC? Un cop comprovat, torna a la zona horària original del sistema.

2.-a) Arrenca una segona màquina virtual (amb la seva tarja de xarxa en mode "adaptador pont") i instal·la-hi, si no ho està ja, el paquet "chrony". Aquest programa pot actuar tant com client NTP (igual que el dimoni *systemd-timesyncd*, com ja sabem) però també, si així es configura, com servidor NTP. Per això darrer, fes el següent: afegeix a l'arxiu `"/etc/chrony.conf"` (a Fedora) o `"/etc/chrony/chrony.conf"` (a Ubuntu) d'aquesta segona màquina les següents línies: `"local stratum 8"` i `"allow X.X.X.X/Y"` (on "X.X.X.X/Y" representa la IP i màscara de la xarxa local -per exemple, 192.168.12.0/24- a la qual vols donar servei), i manté com estan les altres línies que hi puguin haver. Un cop fet aquest canvi, posa en marxa el servei (`sudo systemctl enable chronyd && sudo systemctl start chronyd`).

NOTA: La directiva **"local"** indica, als clients NTP que puguin consultar el servidor *chronyd*, el grau de "confiança" en les respostes que aquest servidor pugui oferir-los. A la pràctica, fa que davant d'aquests aparegui sincronitzat, fins i tot quan mai ho hagi estat (o quan l'última actualització del rellotge hagi sigut fa molt temps). Aquesta directiva s'utilitza normalment en una xarxa aïllada, on es requereix que els ordinadors se sincronitzin entre si, però no necessàriament en temps real (en aquestes circumstàncies el servidor es podria mantenir vagament sincronitzat en temps real mitjançant l'entrada manual, per exemple, o no estar sincronitzat per res). Aquesta directiva pot tenir diverses opcions, entre les quals hi ha **"stratum"**; aquesta opció estableix l'estrat (la "importància", la "rellevància") del servidor: el valor especificat pot estar entre 1 (més rellevant) i 15 (menys rellevant); el valor per defecte és 10, però en tot cas hauria de ser superior a l'estrat màxim esperat a la xarxa quan els servidors NTP externs siguin accessibles. L'estrat 1 indica que un ordinador té una referència de temps real directament connectada a ell (per exemple, GPS, rellotge atòmic, etc.). Els ordinadors d'estrat 2 són els que tenen com a servidors NTP de referència els d'"stratum 1"; els ordinadors d'estrat 3 són els que tenen com a servidors NTP de referència els d'"stratum 2"; etc. Un valor de 10 indica que el rellotge està a tants "salts" d'un rellotge de referència que el seu temps és bastant poc fiable. Per més informació, consultar *man chrony.conf*

NOTA: L'adreça de la directiva **"allow"** representa l'adreça IP de xarxa/subxarxa/host des de la qual es poden realitzar peticions NTP (el valor per defecte és no permetre l'accés a cap client). Si s'utilitza aquesta directiva, *chronyd* funcionarà tant com un client dels seus servidors remots (especificats per les directives "pool" o "server", veure nota següent) com, efectivament, un servidor per als clients permesos.

NOTA: Per indicar a *chronyd* amb quins altres servidors NTP remots amb "stratum" inferior ha de posar-se en contacte per obtenir l'hora actual més precisa, podem fer servir dues directives complementàries: **"server"** o **"pool"**. La sintaxi i significat de la segona és similar a la de la primera excepte que la segona serveix per especificar una agrupació de servidors NTP en lloc d'un servidor NTP únic (per tant, s'espera que el nom de l'agrupació es resolgui cada cop a un d'entre diverses adreces IPs diferents...això es pot comprovar per exemple amb qualsevol comanda DNS com *dig*, *host* o *resolvectl query* contra el valor d'alguna directiva "pool"). Les dues directives accepten les mateixes opcions; per exemple, l'opció **"iburst"** s'utilitza per accelerar la sincronització inicial i l'opció **"prefer"** s'utilitza per especificar el servidor/pool preferit si hi hagués més d'un llistat al fitxer de configuració. L'única opció diferent entre aquestes dues directives és **"maxsources"** (només disponible a "pool"), que fa referència al nombre màxim de fonts NTP que es poden utilitzar des del pool

NOTA: La directiva **"manual"**, si aparegués indicada a l'arxiu de configuració de *chronyd*, permetria modificar l'hora del servidor NTP manualment (mentre aquest s'està executant!) mitjançant la comanda *chronyc settime "hh: mm: ss"*. Això podria ser útil si el servidor NTP no tingués definides (o no funcionessin) les directives "pool"/"server". En aquest sentit, fer notar que el paquet "chrony" incorpora una comanda de terminal anomenada "chronyc" que permet fer tasques de control i monitorització del dimoni *chronyd*. Alguns altres exemples d'ús són:

chronyc tracking : Comprova si *chronyd* està sincronitzat

chronyc sources -v : Verifica els servidors NTP remots que *chronyd* fa servir com a font de temps

chronyc activity : Verifica l'estat de les fonts de temps

chronyc sourcestats : Mostra estadístiques de cada font de temps, tals com el "drift rate" o estimació de l'"offset"

chronyc offline : Notifica a *chronyd* que el sistema no està connectat a Internet (per evitar així malentesos); per retornar a sincronitzar, *chronyc online*

NOTA: Altres directives interessant que es poden utilitzar per implementar *chronyd* com a servidor NTP són, per exemple **"binddevice"** (o **"bindaddress"**), per vincular l'escolta a una tarja de xarxa concreta (o una IP, respectivament), entre moltes altres com ara **"bindcmdaddress"** i **"cmdport"**, que es poden consultar a *man chrony.conf*

b) A la primera màquina virtual (on vas fer el primer exercici), desinstal·la el paquet "chrony" (si estigués instal·lat d'abans) per a què només tinguis com a client NTP disponible el dimoni *systemd-timesyncd*, el qual t'hauràs d'assegurar que estigui encès. Configura tot seguit la tarja de xarxa d'aquesta màquina per a què faci servir com a servidor NTP el que acabes de posar en marxa a l'apartat anterior. Pista: hauràs d'activar el servei *systemd-networkd* amb una configuració similar a aquesta (on "x.x.x.x" és la IP del servidor NTP):

```
[Match]
```

```
Name=enp0s3
```

```
[Network]
```

```
DHCP=yes
```

```
NTP=x.x.x.x
```

c) Executa `timedatectl` ¿Veus l'hora correcta? ¿Què mostra la comanda `timedatectl timesync-status`?

NOTA: Podeu comprovar amb `networkctl status` si el servidor DHCP ens ha assignat algun un altre servidor NTP més enllà del nostre servidor `chronyd`. Si fos el cas, si aquest altre servidor NTP té un "stratum" menor que el nostre servidor `chronyd`, serà triat en comptes d'aquest. El "stratum" es pot veure a la sortida de la comanda `timedatectl timesync-status`. Per evitar acceptar els servidors NTP proporcionats pel servidor DHCP, es pot afegir la directiva `UseNTP=no` sota la secció `[Network]` de l'arxiu `"*.network"` anterior.

NOTA: Fixa't que estem fent servir el client `systemd-timesyncd` contra un servidor Chrony. Tot i que no ho farem, també podríem fer servir un altre client NTP, com el propi Chrony funcionant en aquest mode (y deshabilitar `systemd-timesyncd`). Per fer això, només cal assegurar-se de comentar a l'arxiu `"/etc/chrony.conf"` (a Fedora) o `"/etc/chrony/chrony.conf"` (a Ubuntu) les línies que comencin per `"pool ..."` i, en canvi, afegir la línia `"server x.x.x.x iburst prefer"` (on "x.x.x.x" és la IP del servidor NTP escollit -també pot ser un nom DNS-), mantenint com estant les altres línies que hi puguin haver; per més informació, consultar `man chrony.conf`). Un cop fet aquest canvi, caldrà posar en marxa el servei (`sudo systemctl enable chronyd && sudo systemctl start chronyd`).

d) Visita <https://www.ntppool.org> i busca quants servidors NTP públics existeixen a Espanya. Busca també en algun apartat d'aquesta web informació sobre quins són els requisits que hauria de tenir la nostra màquina per poder afegir-la com un nou servidor NTP dins d'aquest projecte

Units "mount" i generadors

L'arxiu `"/etc/fstab"` és prescindible a sistemes Systemd ja que cada punt de muntatge d'algun dispositiu es pot definir en una unit de tipus "mount", la qual podrà estar habilitada (`enable`) -és a dir, "muntada"- a partir d'un determinat "target" d'arranc (concretament, l'indicat a la seva directiva `WantedBy=`). De totes maneres, per mantenir la compatibilitat, existeix un programa anomenat `"/usr/lib/systemd/system-generators/systemd-fstab-generator"`, executat automàticament per Systemd durant el seu inici, que s'encarrega de llegir les línies que estiguin presents a `"/etc/fstab"` i transformar-les en respectives unitats de tipus "mount" (les quals, en aquest cas, s'ubicaran dins de la carpeta `"/run/systemd/generator"`).

NOTA: Existeix un altre "generator" anomenat `"/usr/lib/systemd/system-generators/systemd-gpt-auto-generator"` que automàticament descobreix, crea els arxius ".mount" pertinents i munta (només a partir del valor del seu GUID, sense cap configuració explícita) les particions GPT corresponents a la carpeta arrel, la carpeta `/home` i la carpeta `/srv` del sistema instal·lat al mateix disc on es trobi la partició ESP (que serà (auto)muntada al seu torn a `/boot` o, si existeix, a `/efi`). En el cas de què alguns d'aquests punts de muntatge estiguin indicats a l'arxiu `/etc/fstab`, llavors té preferència aquest darrer.

Els "generadors" són petits executables que Systemd posa en marxa a l'inici de la seva posada en marxa per tal de realitzar diferents tasques bàsiques variades. En general, transformen certes configuracions que no són "natives" de Systemd a unitats que sí que ho són. És a dir, són generadors d'unitats dinàmiques en cada arranc. Aquests petits programes es poden ubicar en les següents carpetes (ordenades per ordre d'execució i, per tant, de sobrescriviment): `"/usr/lib/systemd/system-generators"`, `"/etc/systemd/system-generators"` i `"/run/systemd/system-generators"` (això vol dir que un enllaç ubicat en `"/etc/systemd/system-generators a /dev/null` podria ser utilitzat per enmascarar un generador concret i, per tant, evitar la seva execució)

El fet de tractar els punts de muntatge de dispositius com unitats permet que el fet de muntar-los i desmuntar-los es transformi simplement en un inici o aturada de la seva unitat corresponent.

Els arxius de les unitats "mount" s'han d'anomenar obligatòriament com la ruta del punt de muntatge que representen però eliminant la primera barra ("`/`") corresponent a la carpeta arrel i substituïnt la resta d'eventuals barres que apareguin a la ruta per guions ("`-`"). Per exemple, si tenim una unitat que representa el punt de muntatge `"/mnt/pepe"`, el seu arxiu de configuració s'haurà d'anomenar `"mnt-pepe.mount"`.

NOTA: L'arxiu de la unitat "mount" corresponent al punt de muntatge de l'arrel del sistema (la carpeta `"/`) es diu `"-.mount"`

A més de les seccions "[Unit]" i "[Install]" comunes a la resta d'unitats, les de tipus "mount" disposen d'una secció **[Mount]** que pot incloure, entre d'altres, les següents directives:

What=/dev/disp1 : Dispositiu a muntar. El valor d'aquesta directiva també pot ser *UUID=unUuid* o *LABEL=unaEtiqueta*, entre altres

Where=/ruta/punt/muntatge : Ha de coincidir amb el nom de la unitat (canviant les barres per guions). No cal que existeixi prèviament

Type= { ext4 | vfat | xfs | ... } Sistema de fitxers del dispositiu a muntar.

Options = { defaults | ... } : Qualsevol opció de muntatge vàlida a "/etc/fstab" es pot indicar aquí. Si n'hi ha més d'una, s'escriuran totes seguides separades per comes.

TimeoutSec=nº : Indica la quantitat de temps (per defecte en segons) que el sistema esperarà abans de considerar que el muntatge ha fallat.

NOTA: Encara que no estiguin escrites explícitament al seu fitxer ".mount", totes les unitats de tipus "mount" adquireixen automàticament una dependència *Before=* i també *Conflicts=* amb "umount.target" per tal de desmuntar-se durant l'apagada del sistema. D'altra banda, unitats "mount" corresponents a sistemes de fitxers locals adquireixen automàticament una dependència *After=* amb "local-fs-pre.target" i, en canvi, unitats "mount" corresponents a sistemes de fitxers en xarxa (com ara NFS o Samba) adquireixen automàticament una dependència *After=* amb "remote-fs-pre.target", "network.target" i "network-online.target" (i respecte aquest últim target també s'afegeix una dependència *Wants=*) a més d'una dependència *Before=* amb "remote-fs.target", per assegurar-se de què la xarxa estigui habilitada abans d'intentar realitzar el muntatge. Per més informació, consultar l'apartat "Automatic dependencies" de *man systemd.mount*

Unitats "automount"

Les unitats de tipus "automount" han de venir acompanyades sempre d'una unitat de tipus "mount" homònima. És a dir, no poden existir unitats "automount" sense la seva parella "mount". Les unitats "automount" serveixen, si estan iniciades, per no haver de tenir iniciada la corresponent unitat "mount" mentre aquesta no s'utilitzi. D'aquesta manera, no es té muntat el dispositiu en qüestió inútilment. Només quan l'usuari hi accedeix, en aquell moment, la unitat "mount" s'iniciarà sola i l'usuari podrà entrar al punt de muntatge sense notar que fins llavors el seu contingut no estava disponible. En aquest sentit, les unitats "automount" serien a les "mount" com les "socket" són a les "service": només cal tenir habilitades les unitats "automount" mentre que les unitats "mount" associades podran romandre apagades per defecte.

Les unitats "automount" disposen, a més de les seccions [Unit] i [Install] conegudes, de la secció **[Automount]** la qual només sol estar composta d'un parell de directives només:

Where=/ruta/punt/muntatge : Ha de coincidir amb el nom de la unitat (canviant les barres per guions). No cal que existeixi prèviament. Representa el punt de muntatge que s'estarà "monitoritzant" per detectar si s'hi accedeix i, per tant, si s'ha d'iniciar la unitat "mount" associada

TimeoutIdleSec=nº : Indica la quantitat de temps (per defecte segons) d'inactivitat en el punt de muntatge en qüestió que Systemd s'esperarà per desmuntar-lo. Per defecte aquest temps és infinit.

EXERCICIS:

1.- a) ¿Què fa, en general, la comanda `systemctl list-units -t mount` ? ¿I `systemctl list-units --all -t mount` ?

b) La comanda `systemctl status ---.mount` permet saber si aquesta unit ha sigut "generated" ¿On ho mostra i què vol dir això? En aquest sentit, executa la comanda `systemctl cat ---.mount` i raona el significat del valor que veus a la directiva `SourcePath=`

NOTA: El doble guió "--" és necessari per a què la comanda `systemctl` no doni error de sintaxi

2.-a) Descarrega't un arxiu "iso" si no en tens cap a mà (un de tamany petit -per descarregar-ho ràpid- és, per exemple, <http://mirror.slitaz.org/iso/rolling/slitaz-rolling.iso>). Crea tot seguit l'arxiu `"/etc/systemd/system/opt-iso.mount"` amb el següent contingut...:

```
[Unit]
Description=Muntar iso
[Mount]
What=/ruta/dalgun/arxiu.iso
Where=/opt/iso
Type=iso9660
Options=loop
```

...i inicia la unit. ¿Què apareix a la sortida de la comanda `findmnt`? ¿Què veus quan fas `ls /opt/iso`?

b) ¿Què significa que la unit sigui "static" (ho pots veure en fer `systemctl status opt-iso.mount`) ? PISTA: consulta els apunts de "Systemd (I)" i/o prova d'executar `systemctl enable opt-iso.mount` a veure què passa. A partir de la resposta anterior, ¿què creus que passaria llavors si hi afegíssim a l'arxiu anterior la secció `[Install]` amb la línia `WantedBy=multi-user.target` ? Fes-ho i executa a continuació `systemctl enable opt-iso.mount` de nou. ¿Què passa ara?

c) Mou de lloc l'arxiu iso i intenta iniciar la unit "mount". ¿Què passa?

3.-Associa a la unit "mount" creada a l'exercici anterior una unit "automount" homònima que tingui un timeout de 20 segons. Atura la unit "mount" i inicia la unit "automount". Executa la comanda `lsblk` per confirmar que l'arxiu iso no està muntat. Fes ara un simple `ls` al punt de muntatge de l'arxiu iso. ¿Què veus? Comprova seguidament l'estat de la unit "mount" amb `systemctl status` o `systemctl list-units` i també amb `lsblk`. ¿Quin és? Espera't 20 segons i torna'l a comprovar. ¿Quin és ara?

4.-a) Els muntatges de tipus "bind" serveixen per fer que una carpeta estigui "muntada" no sobre cap partició ni dispositiu de bloc sinó sobre una altra carpeta. Crea l'arxiu `"/etc/systemd/system/home-usuari-lerere.mount"` amb el següent contingut...:

```
[Unit]
Description=Prova bind
[Mount]
What=/etc
Where=/home/usuari/lerere
Options=bind
```

...i inicia la unit. ¿Què apareix a la sortida de la comanda `findmnt`? ¿Per què creus que "no surt res", en canvi, a la sortida de la comanda `lsblk`? ¿Què passa quan accedeixes a la carpeta `"/home/usuari/lerere"`?

b) Si et trobessis amb un fitxer anomenat `"mnt-usr.mount"` amb el següent contingut, ¿quin tipus de recurs es muntaria quan s'iniciés, i on? ¿Per què no hi ha cap línia `After/Wants=network-online.target` (la resposta està a la darrera NOTA de teoria d'aquest apartat)? ¿Què passaria si s'executés `systemctl enable mnt-usr.mount`?

```
[Unit]
Description=Mount NFS share
[Mount]
What=192.168.50.151:/usr
Where=/mnt/usr
Type=nfs
Options=defaults
TimeoutSec=60
[Install]
WantedBy=remote-fs.target
```

Units "timer"

Les unitats "timer" s'usen per programar tasques:

- a) En un/s moment/s concret/s
- b) Després d'un (o cada) cert temps.

Una altra unitat (generalment de tipus "service") homònima amb el "timer" ha d'existir per tal de ser executada quan el moment definit al timer arribi. Convé aclarir que aquesta unitat "service" no cal que tingui cap secció `[Install]` perquè no cal habilitar-la explícitament: només caldria habilitar, si es volgués automatitzar la seva execució, el seu "timer" associat, el qual sí tindrà una secció `[Install]` (amb una línia `WantedBy=` generalment apuntant al target predefinit "timers.target").

NOTA: Si es donés el cas que una tasca s'executa durant tant de temps que arribés el moment, en el cas d'un timer periòdic, d'activar-se de nou, cap nova activació no es realitzarà perquè Systemd entén que la tasca ja està en marxa i la deixarà acabar sense executar-ne cap tasca addicional més (i això és així independentment del tipus de servei executat: *oneshot* -que passen de l'estat "activating" però bloquejant quan comencen a "inactive" quan acaben (a no ser que s'usi "RemainAfterExit=yes", on llavors acabaran en "active")-, *simple* -que passen d'"active" quan comencen -i per tant, un timer no hi farà res, a "inactive" quan acaben-, etc)

A més de les seccions `[Unit]` i `[Install]`, el fitxer de configuració del "timer" té una secció **[Timer]** que pot contenir alguna de les següents directives:

*Per un moment/s concret/s:

OnCalendar=... : Activa la unitat associada al moment exacte indicat. El format general que pot tenir el valor d'aquesta directiva és: `[nomDiaSetmana] any-mes-dia hora:minuts:segons on`:

- Els dies de la setmana s'han d'escriure en anglès i en forma abreviada: *Mon, Tue, Wed, ...*
- Es pot escriure el comodí * a qualsevol unitat (any, mes, dia, hora, minuts, segons)
- Es poden escriure varis valors en una unitat si se separen per comes
- Es pot indicar un rang de valors en una unitat escrivint ".." entre el primer i últim valor
- Es pot indicar un interval en una unitat escrivint "/n^o". Per exemple, `*:*:20/15` vol dir a les `*:*:20`, `*:*:35`, `*:*:50` i, com que s'acaba el minut, tornarà a començar un altre cop a `*:*:20`

Aquesta directiva també admet determinades paraules clau com a valors, les quals representen determinades combinacions de dia i hora. Concretament:

```
minutely → *_-* *:*:00
hourly → *_-* *:*:00:00
daily → *_-* 00:00:00
monthly → *_-*01 00:00:00
weekly → Mon *_-* 00:00:00
yearly → *-01-01 00:00:00
quarterly → *-01,04,07,10-01 00:00:00
semiannually → *-01,07-01 00:00:00
```

NOTA: Expressions com "daily" i "weekly" fan referència a hores d'inici específiques i, per tant, qualsevol "timer" que comparteixi aquests esdeveniments del calendari començarà simultàniament. Això pot causar un rendiment baix del sistema si els serveis associats a aquests "timers" competeixen pels recursos del sistema. L'opció **RandomizedDelaySec=** de la secció [Timer] evita aquest problema escalonant aleatòriament l'hora d'inici de cada timer.

*Després (i/o cada cert) temps:

OnBootSec=n° : Indica la quantitat de temps que Systemd s'esperarà a realitzar (una vegada) la tasca, començant a contar a partir de l'arranc del sistema. Una altra directiva similar a l'anterior és **OnStartupSec=n°** , la qual conta l'espera, però, en relació a quan el propi Systemd ha arrencat. Ambdues directives són útils quan el "timer" en qüestió està *enabled*

OnActiveSec=n°: Indica la quantitat de temps que Systemd s'esperarà a realitzar (una vegada) la tasca, començant a contar a partir de l'activació del "timer" (útil quan el "timer" no està *enabled* sinó que es fa un *systemctl start*)

OnUnitActiveSec=n° : Indica la quantitat de temps que Systemd s'esperarà a realitzar (de forma periòdica) la tasca, començant a contar des de la primera vegada que es va realitzar (generalment mitjançant *OnBootSec=/OnStartupSec=*). Aquesta directiva (en combinació amb alguna de les anteriors) és molt utilitzada per realitzar, per exemple de forma diària (si s'indica "1d") una determinada tasca (normalment de tipus "oneshot" tot i que no té per què). També existeix la directiva *OnUnitInactiveSec=*

Les quantitats de temps indicades a les directives anteriors per defecte són segons però afegint algun dels següents sufixes es pot transformar en altres unitats de temps: *us* (microsegons), *ms* (milisegons), *s* (segons), *m* (minuts), *h* (hores), *d* (dies), *w* (setmanes), *M* (mesos) o *y* (anys). Fins i tot es poden combinar, com ara així, per exemple: "5m 20s" (5 minuts i 20 segons).

NOTA: Systemd també accepta especificacions de temps relatiu. Concretament, un interval de temps que té el prefix + s'avalua a l'hora actual més l'espai de temps indicat. Igualment, un interval de temps que té el prefix - s'avalua a l'hora actual menys el període de temps indicat (o bé, en lloc d'usar el prefix -, de forma equivalent també es pot sufixar amb un espai i la paraula "ago")

NOTA:La directiva *OnCalendar=* no és exclouent amb les altres: es poden escriure a la vegada si és necessari.

Altres directives interessants són:

Persistent=yes : Si aquesta directiva val "yes", el "timer" en qüestió executarà la tasca associada immediatament després d'arrencar el sistema si li hagués tocat executar-se abans mentre el sistema estava apagat. Se sol fer servir juntament amb la directiva *OnCalendar=*.

NOTA: Això s'aconsegueix gràcies a què es guarda al disc dur la darrera vegada que es va executar la tasca

WakeSystem=yes : Si aquesta directiva val "yes" permet treure el sistema de l'estat de suspensió si li toca executar una tasca programada durant aquest estat. Per defecte val *no (false)*

Unit=nomUnit.service : Indica la tasca a realitzar quan arribi el moment indicat pel "timer". Només cal escriure-la si el nom del fitxer de configuració de la tasca (l'arxiu "service") no és igual al del fitxer de configuració del "timer".

NOTA: És interessant tenir en compte que la relació entre units "timer" i units "service" no només pot ser aquesta. Per exemple, un unit "timer" pot tenir dins la seva secció [Unit] una línia *Requires=* que apunti a un unit "service" (per tal de què en posar-se en marxa el timer aquest obligui a activar un determinat servei); o un unit "service" pot tenir dins la seva secció [Unit] una línia *Requires=* que apunti a un unit "timer" (per tal de què en posar-se en marxa el service aquest obligui a activar un determinat timer), etc

AccuracySec=n° : Indica la precisió del "timer" (per defecte és 1 minut). Per tasques que es vulguin realitzar cada cert segons (o a un segon determinat) cal indicar explícitament *AccuracySec=1s* Es pot arribar fins els microsegons de precisió

Com qualsevol altra unit, es poden veure la llista de timers amb la comanda `systemctl list-units -t timer` però a més disposem de la comanda específica **systemctl list-timers**, la qual informa de quan va ser la darrera vegada que es va executar la tasca, quant li queda per tornar-la a executar, etc

A l'igual que passava amb les unitats "service", les unitats "timer" també es poden executar en el context d'un usuari particular (és a dir, no cal que estiguin activats per tot el sistema en general). Per aconseguir això només cal crear l'arxiu *.timer (i l'arxiu *.service associat) dins d'alguna de les carpetes reservades per aquest context (és a dir, per ordre de sobreescritura: "/usr/lib/systemd/user", "~/.local/share/systemd/user", "/etc/systemd/user" o "~/.config/systemd/user") i executar les comandes adients (`systemctl daemon-reload`, `systemctl start`, `systemctl enable`, `systemctl list-timers...`) amb el paràmetre `--user`

EXERCICIS:

1.-a) ¿Quan s'executaria una tasca que tingués associada un "timer" amb la següent directiva:
`OnCalendar=Mon..Fri *-*- * 11:30:00` ?

aI) ¿I amb aquesta: `OnCalendar=Mon,Wed *-*-1..7 00:00:00` ?

aII) ¿I amb aquesta: `OnCalendar=2020-*-1,5 11/3:12:14` ?

aIII) ¿I amb aquesta: `OnCalendar=*:0/1` ?

b) ¿I si aquest "timer" tingués aquestes dues directives?

`OnBootSec=15min`

`OnUnitActiveSec=12h`

bI) I aquestes?

`OnBootSec=5min`

`OnCalendar=*:0/1:0`

c) Executa la comanda `systemctl cat systemd-tmpfiles-clean.timer` i digues per a què serveix aquest "timer" proporcionat per Systemd. ¿Quina és la comanda que invoca?

NOTA: Aquest "timer" té sentit en màquines de tipus servidor que no s'apaguen sovint (i que, per tant, poden mantenir en memòria fitxers "temporals" durant molt de temps).

NOTA: La comanda emprada disposa dels arxius de configuració "*.conf" ubicats principalment a "/usr/lib/tmpfiles.d" i "/etc/tmpfiles.d" que l'indiquen quins fitxers (o carpetes) concrets són els que haurà d'esborrar. Més informació a *man tmpfiles.d*

2.-a) Fes un "timer" que executi la tasca d'escriure la paraula "Guapo" al Journalctl en el moment d'activar-se ell mateix (amb `systemctl start`) i, a partir de llavors, cada 10 segons. Comprova-ho.

NOTA: Recorda que hauràs d'executar també `systemctl daemon-reload`

NOTA: Tingues en compte que per realitzar tasques amb precisió menor d'un minut (per exemple, d'un segon) cal obligatòriament afegir sempre la directiva `AccuracySec=1s`

b) Modifica el "timer" anterior per a què realitzi la mateixa tasca però ara exactament només als segons 15 i 45 de tots els minuts de totes les hores i fes també que sigui de tipus Persistent. Activa'l de nou i observa si ara, efectivament, apareixen els missatges només en els segons 15 i 45 de cada minut.

bII) Habilita el "timer" anterior (amb `systemctl enable`). Apaga la màquina i deixa-la apagada més d'un minut. Encén-la de nou i comprova al Journal que s'hagi executat la tasca, a més de quan li toca, també en iniciar el sistema (degut a què és Persistent i la màquina estava apagada) encara que no sigui exactament als segons indicats.

c) Canvia la tasca associada al "timer" anterior per a què ara, en comptes d'enviar el missatge "Guapo" al journal, faci una cosa més útil: per a què generi un arxiu tar.gz amb el contingut de la carpeta Escriptori i el guardi a la carpeta Baixades. Per a no sobreesciure el fitxer a cada execució, fes que l'arxiu tar.gz inclogui al seu nom la data de la seva creació, d'una forma similar a aquesta: *nom-\$(date +%H:%M:%S).tar.gz*

NOTA: La directiva *ExecStart=* no sap interpretar el símbol *\$(...)*. Si volem que s'interpreti, ho hem de fer a través de Bash. Per tant, el valor de la directiva *ExecStart=* hauria de ser semblant a *ExecStart=/bin/bash -c "/bin/tar -czf /ruta/nom-\$(date +%H:%M:%S).tar.gz /ruta/carpeta" (veieu, a més, que cal escriure dos % per escapar-lo, ja que el símbol % dins d'un arxiu ".service" té un significat especial, recordeu el tema de les plantilles)*

3.-¿Què fa aquesta parella de "timer" i "service" ? Prova-ho activant el "timer" amb *systemctl enable*:

```
# /etc/systemd/system/hola.timer
[Unit]
Description=Spoiler no
[Timer]
AccuracySec=1s
OnBootSec=1
OnUnitActiveSec=10
[Install]
WantedBy=timers.target

# /etc/systemd/system/hola.service
[Unit]
Description=Spoiler tampoc
ConditionPathExists=/opt/hola #La clau està aquí
[Service]
Type=oneshot
ExecStart=printf "MEEE\n"
```

4.-a) ¿Què fa aquesta parell de "services"? Prova'ls amb *systemctl start startpepe* (i *systemctl status startpepe*) i *systemctl start stoppepe* (i *systemctl status stoppepe*). Un cop provats, assegura't de què tots dos romanguin aturats.

```
#startpepe.service
[Unit]
Description=Misteri
[Service]
Type=simple
ExecStart=nc -l -p 8000

#stoppepe.service
[Unit]
Description=Misteri 2
Conflicts=startpepe.service
[Service]
Type=oneshot
ExecStart=printf "Servidor Netcat apagat"
```

b) ¿Què farien aquest parell de "timers"? Pots provar d'activar els dos "timer" (amb *systemctl start*) adequant l'hora indicada a l'actual per comprovar-ho.

```
#startpepe.timer
[Unit]
Description= Misteri 3
[Timer]
OnCalendar= *-*-* 17:00:00

#stoppepe.timer
[Unit]
Description= Misteri 4
[Timer]
OnCalendar= *-*-* 17:05:00
```

c) Atura els dos "timers" anteriors, afegeix ara a la secció *[Unit]* de "startpepe.service" la línia *Wants=stoppepe.timer* i inicia només el servei "startpepe.service" ¿Què passa d'aquí cinc minuts? ¿Per què?

5.-Fes que un "service", definit sota "*~/config/systemd/user*" i invocador de l'script següent, s'executi cada minut mitjançant un "timer", també definit sota "*~/config/systemd/user*". Posa en marxa el "timer" en el context del teu usuari (*--user*). ¿Què passa?

```
#!/bin/bash
selection=$(find $HOME -type f -name "*.jpg" -o -name "*.png" | shuf -n1)
gsettings set org.gnome.desktop.background picture-uri "file://$selection"
```

NOTA: La comanda `gsettings` és l'equivalent en terminal al panell de control de Gnome. Però per a què funcioni cal que s'executi en el mateix entorn en què la sessió de Gnome actual està funcionant (o dit d'una altra manera: cal que faci servir el mateix canal D-Bus de sessió). És per això que el "service" que invocarà aquest script ha de ser gestionat pel Systemd de "sessió" i no pel de "sistema".

NOTA: És interessant fixar-se també en l'ús en l'script de la comanda `shuf`

NOTA: Hi ha una manera més "elegant" d'aconseguir el mateix objectiu fent servir les eines que proporciona el propi Gnome, però la veurem més endavant

6.- Si volem executar serveis o timers en el context d'usuari (com l'exercici anterior) però per tots els usuaris, els arxius "service" i "timer" corresponents caldrà crear-los sota la carpeta `/usr/lib/systemd/user` (o `/etc/systemd/user`). Sabent això, crea en alguna d'aquestes carpetes els següents "service" i "timer" i tot seguit executa `systemctl --user enable hola.timer && systemctl --user start hola.timer`. ¿Què passarà?

```
#hola.timer                                #hola.service
[Unit]                                       [Unit]
Description=Un altre misteri 2             Description=Un altre misteri
[Timer]                                     [Service]
OnCalendar=14:50                           Type=oneshot
OnCalendar=14:54                           ExecStart=notify-send -u critical "Atencio" "Perill imminent"
[Install]
WantedBy=timers.target
```

b) ¿Què ensenya la comanda `systemctl --user list-timers`? ¿I la comanda `systemd-analyze calendar 14:50`?

Existeix una manera d'establir tasques programades sense haver de crear un arxiu `*.timer` gràcies a la comanda `systemd-run`, la qual ens permet crear "timers" (i, de fet altres tipus d'units) de tipus "transient" (és a dir, "fugaços"). Per exemple, per crear un fitxer d'aquí a 30 segons es pot fer el següent:

```
systemd-run --on-active=30 /bin/touch /tmp/foo
```

També es pot fer servir per executar un arxiu "service" preexistent que no tingui cap "timer" associat:

```
systemd-run --on-calendar="*-*-* 12:13:14" -u unServei.service
```

8.-a) Prova les dues comandes `systemd-run` mencionades al paràgraf anterior (en el segon cas, especifica una hora propera i un servei -com ara l'Apache- que no estigui funcionant en aquests moments i comprova que, efectivament, s'executin quan toca.

b) Suposant que tens el servei `Nftables` iniciat i gestionat per una unit `*.service` convencional, ¿què faria aquest segon servei addicional? Prova'l amb `systemctl start`

```
[Unit]
Description=Mystery
BindsTo=nftables.service
After=nftables.service
[Service]
Type=oneshot
ExecStart=systemd-run --on-active=180 --timer-property=AccuracySec=1s systemctl stop nftables
```