

ADSO Training 2

Instal·lació d'aplicacions

Índex

1. Introducció	2
2. Com començar	2
2.1. Contesta les següents preguntes abans de començar:	2
2.2. Configuració del teclat i de l'idioma	2
2.3. Actualització del kernel	3
3. Introducció a la Instal·lació d'aplicacions	4
4. Instal·lació de paquets binaris	5
4.1. Instal·lació manual	5
Consulteu l'ajuda (man del sistema o pàgina de manual) de la comanda <code>dpkg</code> i completeu la següent taula	5
Baixeu el paquet corresponent per instal·lar <i>make</i>	6
Quina diferència hi ha entre desinstal·lar un paquet i purgar un paquet?	7
Ara volem instal·lar els programes <i>lynx</i> (un navegador web per mode text) i <i>lftp</i> (un client avançat de ftp). Baixeu els paquets corresponents i instal·leu-los amb <i>dpkg</i> .	7
Executeu les comandes <i>lynx</i> i <i>lftp</i> per assegurar-vos que funcionen correctament.	7
4.2. Instal·lació amb un gestor de paquets	7
Explica les diferències entre <i>apt-get</i> , <i>apt</i> i <i>aptitude</i>	8
4.2.1 Configuració dels repositoris de software	8
4.2.2 Llista de paquets disponibles als repositoris	8
Amb quina comanda actualitzem els paquets instal·lats a l'última versió disponible?	9
4.2.3 Instal·lació del sistema X-Window	9
4.2.4 gestor de finestres i ambient d'escriptori	10
Escolliu un gestor de finestres i un ambient d'escriptori i instal·leu-los	10
4.2.5 Instal·lació de paquets	10
4.3. Instal·lació de binaris pre-compilats	11
5. Instal·lació a partir del codi font	12
6. Bibliografia	15

1. Introducció

L'objectiu és instal·lar correctament software preparat específicament per a un sistema

2. Com començar

2.1. Contesta les següents preguntes abans de començar:

- **Quina comanda es fa servir per connectar-se a un servidor d'sftp?**
`sftp usuari@direccio_del_servidor`, on “usuari” serà el nostre nom d'usuari, i “direccio_del_servidor” sera la direcció IP o el nom del domini del servidor al que ens volem connectar. Després d'aquesta comanda, ens demanaran la contrasenya de l'usuari al servidor SFTP per completar la connexió.
- **Amb quina comanda sftp es llisten els continguts d'un directori al servidor?**
 Amb la comanda `ls`. Si volem saber el contingut d'un directori en concret, seria amb:
`ls /ruta/al/directori/concret`
- **Amb quina comanda sftp podem agafar un fitxer del servidor?**
 Amb la comanda `get nom_del_arxiu` .
 Si no estem en el directori actual de l'arxiu, també podem fer:
`get nom_del_arxiu /ruta/local/del/arxiu` .
- **Amb quina comanda sftp podem agafar més d'un fitxer del servidor?**
`get *` : agafaria tots els fitxers de la carpeta actual.
`get *.txt` : agafaria tots els fitxers amb l'extensió .txt de la carpeta actual.
`get arxiu*.txt` : agafaria tots els fitxers els quals el seu nom comenci per “arxiu” i tinguin la extensió .txt .
- **Com es llista el contingut d'un fitxer tar?**
`tar -tf nom_del_arxiu.tar`
- **I si a més es troba comprimit amb gzip?**
`tar -tzf nom_del_arxiu.tar.gz` (en aquest cas, la z serveix per descomprimir l'arxiu amb gzip abans de mostrar el seu contingut)
- **Com es descomprimeix un fitxer tar?**
`tar -xf nom_del_arxiu.tar`
- **I un tar.gz? I un tar.bz2?**
`tar -xzf nom_del_arxiu.tar.gz`
`tar -xjf nom_del_arxiu.tar.bz2`
- **Com es feia un link a un fitxer?**
`ln nom_del_arxiu hard_link`
- **I un softlink?**
`ln -s ruta_al_arxiu enllac_simbolic`

- **Per a què serveix la variable d'entorn PATH?**

La variable PATH proporciona una llista de directoris que el sistema operatiu busca de forma seqüencial per trobar l'executable de qualsevol comanda ingressada.

És a dir, que la seva funció principal és especificar les rutes dels directoris on el sistema operatiu buscara els programes executables que volguem usar mitjançant una comanda al shell.

2.2. Configuració del teclat i de l'idioma

Explica per a que serveixen els següents fitxers: **keyboard** i **locale.conf**

keyboard és un fitxer per definir la configuració del teclat, per establir el disseny, el idioma d'aquest, el model i altres configuracions relacionades amb el hardware d'aquest.

locale.conf és un fitxer que s'utilitza per definir una configuració global de la localització i del idioma del sistema Operatiu, importa CLEAR

nt per poder definir l'idioma del sistema, el format de data i hora, el tipus de moneda i altres configuracions relacionades amb la localització.

Que fan les comandes **dpkg-reconfigure** i **localectl**?

La comanda **dpkg-reconfigure** es una comanda del sistema Debian que permet configurar paquets de software ja instal·lats i poder tenir accés per reconfigurar-los i executar-los correctament. Executat amb el paquet **keyboard-dpkg-reconfigure** ens permet reconfigurar la configuració inicial del teclat.

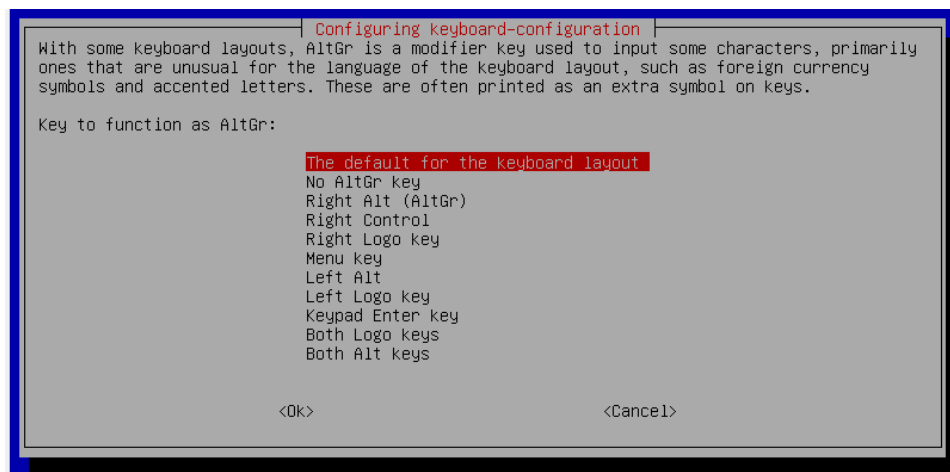
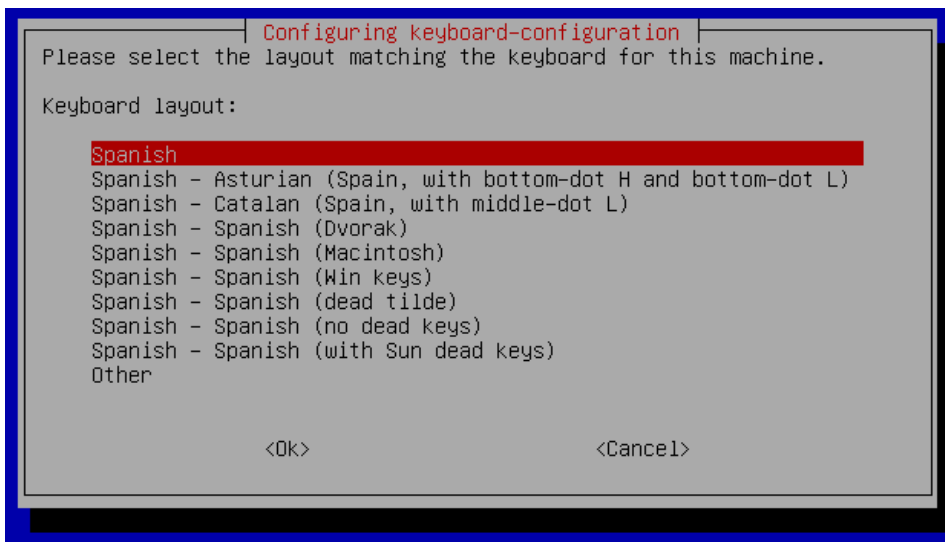
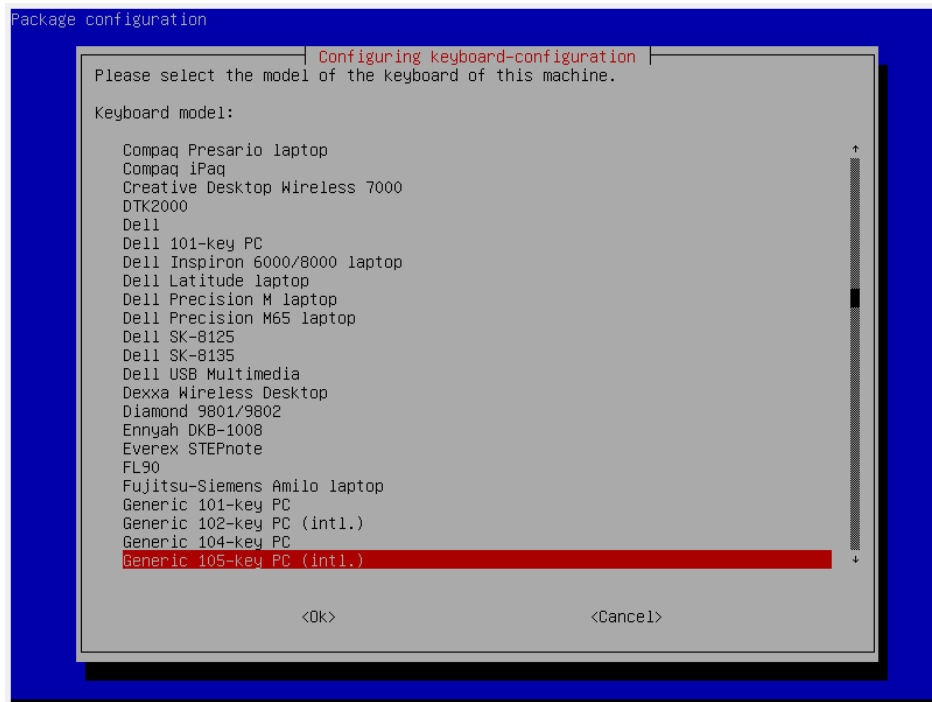
La comanda **localectl** es una comanda base del sistema Linux per poder configurar la configuració de l'ubicació i l'idioma del sistema. Executat amb el fitxer localectl locale.conf podem reconfigurar l'idioma en tot el sistema i teclat i la localització.

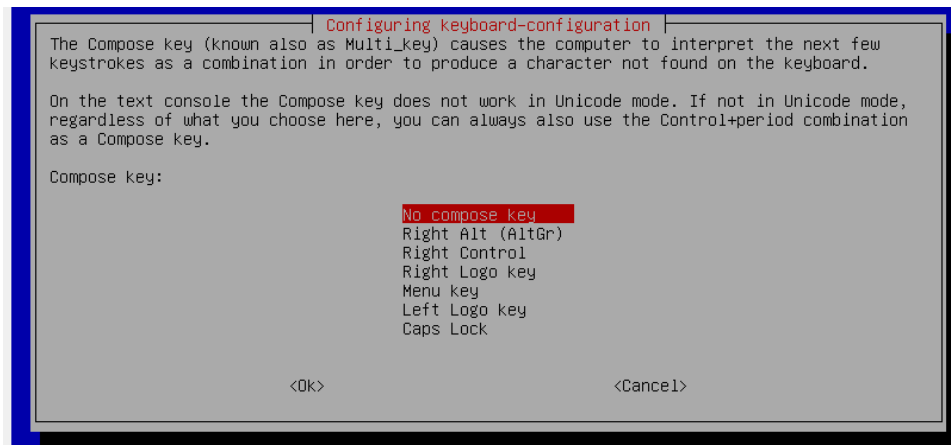
Configura el teclat i l'idioma de la màquina. Mostra els paràmetres de configuració

Configuració del teclat:

```
root@MarionaF (Wed Oct 18):<~># dpkg-reconfigure keyboard-configuration service
```

Triar les següents opcions:





Reconfiguració del sistema local:

```
console-getty.service console-setup.service
root@MarionaF (Wed Oct 18):<~># systemctl restart console-setup.service
root@MarionaF (Wed Oct 18):<~>#
```

2.3. Actualització del kernel

Què és el kernel?

El kernel és una part del sistema operatiu d'un ordinador, és el nucli del sistema operatiu i té control total sobre el sistema. S'encarrega, entre altres coses de la Gestió de Recursos, Gestió de Processos, Gestió de Memòria, Gestió de Dispositius, Seguretat i Control d'Accés, i Comunicació entre Software i Hardware.

Explica que fan les comandes **uname**, **apt-cache**, **apt upgrade**, **apt full-upgrade**, **apt update** i les seves opcions

uname: mostra informació sobre el sistema en el que estem treballant.

-a : mostra tota la informació disponible.

-s : mostra el nom del kernel.

-r : mostra la versió del kernel.

-v : mostra informació detallada, incloent la versió del kernel i la data de compilació.

-m : mostra l'arquitectura del hardware.

apt-cache: és una eina relacionada amb la gestió de paquets de debian.

apt-cache search paquet: busca un paquet en els repositoris.

apt-cache show paquet: mostra els detalls sobre un paquet específic.

apt-cache showpkg paquet: mostra informació detallada sobre les dependències d'un paquet específic.

apt-update: actualitza la llista de paquets disponibles en els repositoris configurats en el sistema. No instala ni actualitza paquets, només actualitza l'informació de quins paquets estan disponibles i d'on obtenir-los.

apt-upgrade: actualitza tots els paquets ja instal·lats en el sistema a les últimes versions disponibles. No instala ni elimina cap paquet, només actualitza els existents.

apt full-upgrade: és com el apt-upgrade, però pot instal·lar o eliminar paquets (si és necessari) per resoldre dependències. S'utilitza per actualitzacions del sistema més grans, ja que poden resultar en canvis en les dependències del sistema.

Quina versió del kernel te la teva màquina?

La versió del kernel, la qual podem mirar amb la comanda `uname -r` és: 4.19.0-6-amd64

Quines versions del kernel hi ha disponibles al repositori?

```
root@MarcR (Wed Oct 18):<~># uname -a
Linux aso-client 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2 (2019-08-28) x86_64 GNU/Linux
```

Actualitza tot el sistema (kernel, paquets,..) i fes un reboot

Es possible que ens trobem amb un error alhora de fer un apt update o un apt upgrade.

Al nostre ordinador: Necessitarem un usb

Anem a la pàgina següent: <https://packages.debian.org/bullseye/all/debian-archive-keyring/download>

Com podem veure haurem d'escriure la següent comanda a **/etc/apt/sources.list**

```
deb http://ftp.de.debian.org/debian bullseye main
```

Per modificar el fitxer `sources.list` farem `nano /etc/apt/sources.list`

```
GNU nano 7.2 /etc/apt/sources.list
deb http://deb.debian.org/debian stable main
```

en el següent link els següents enllaços, trobar el ftp d'espainya(es):

<https://packages.debian.org/bullseye/all/debian-archive-keyring/download>

North America

- <ftp.us.debian.org/debian>
- <http.us.debian.org/debian>
- <ftp.debian.org/debian>
- <ftp.ca.debian.org/debian>
- <ftp.mx.debian.org/debian>

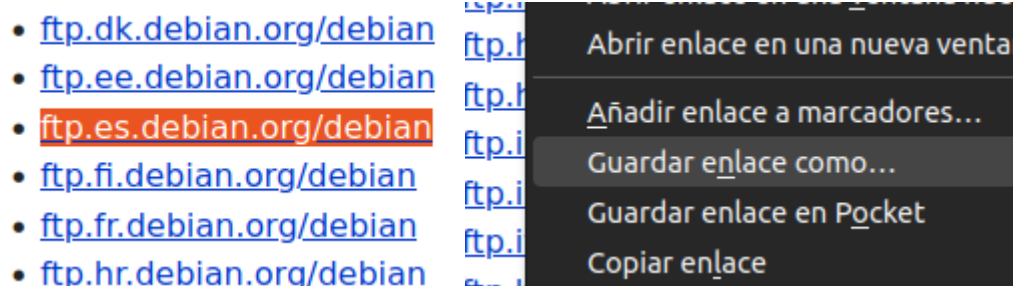
South America

- <ftp.br.debian.org/debian>
- <ftp.cl.debian.org/debian>

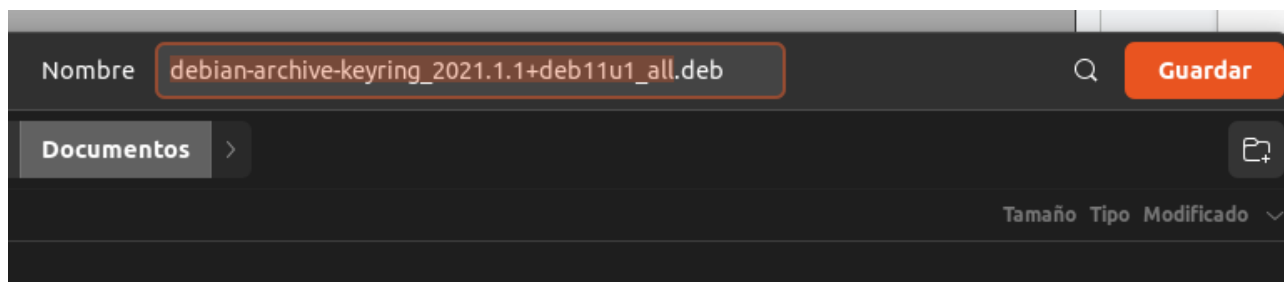
Europe

- <ftp.de.debian.org/debian>
- <ftp.at.debian.org/debian>
- <ftp.bg.debian.org/debian>
- <ftp.ch.debian.org/debian>
- <ftp.cz.debian.org/debian>
- <ftp.dk.debian.org/debian>
- <ftp.ee.debian.org/debian>
- <ftp.es.debian.org/debian>
- <ftp.fi.debian.org/debian>
- <ftp.fr.debian.org/debian>

Hem de fer clic dret sobre el següent i seleccionar l'opció de guardar com.



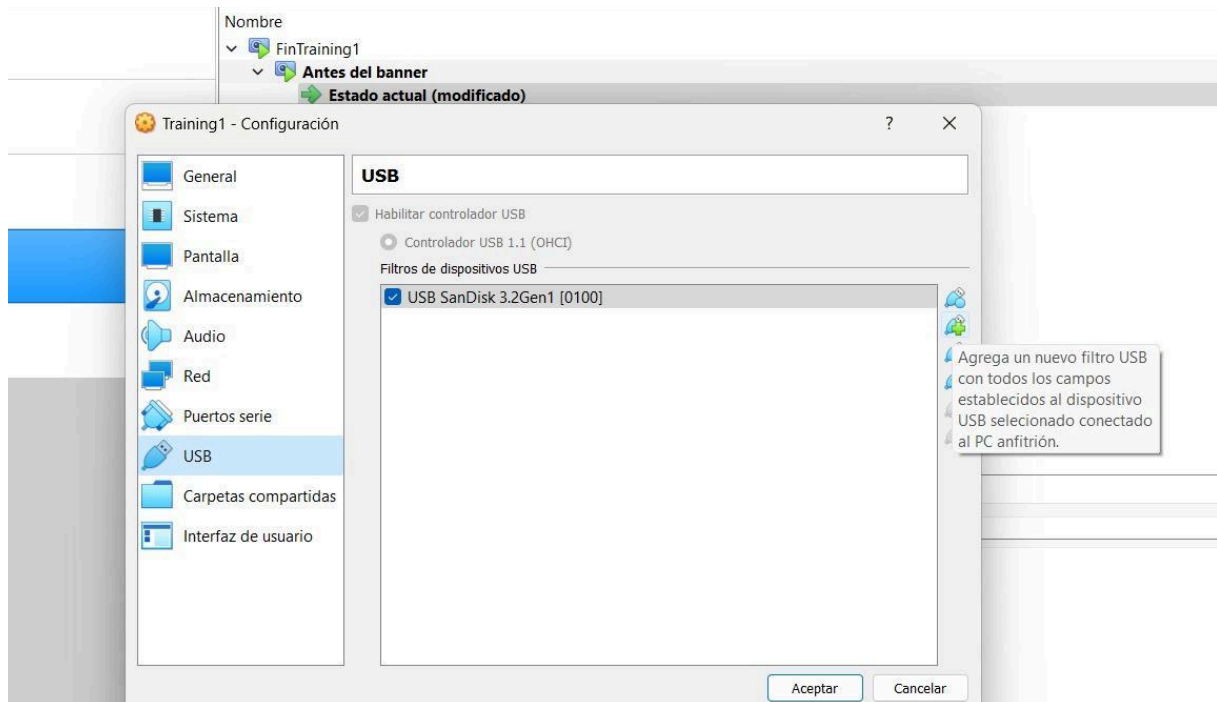
Hauriem de veure que es guarda un arxiu **.deb**



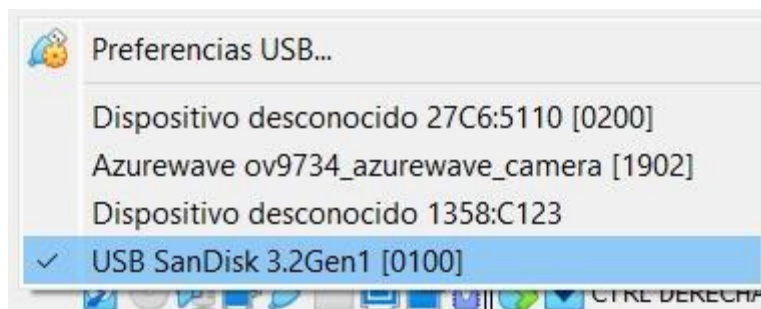
Això ho haurem de posar al usb.

Ara ja a la nostre maquina virtual:

Configurem l'usb



Comprovem dins la maquina que el tenim actiu (abaix a la dreta)



Ara hem de montar el usb, farem servir les següents comandes consecutivament (hem de ser root)

PAS 1

```
mkdir /media/NomQueVulguem
Exemple: mkdir /media/usb
```

PAS 2

Comprovem que s'ha creat amb

```
ls /media
```

PAS 3

Mirem on tenim el usb amb la comanda **lsblk**

Normalment estara al sdb1, sdc1...

PAS 4

Montem el directori

mount /dev/sdX(el que ens posi a la comanda anterior) /media/NomUsb

Exemple: **mount /dev/sdb1 /media/usb**

Pas 5

Fem **cd** i ens coloquem al directori **/media/NomUsb**

Executem: **dpkg -i NomDelArxiu .deb** (mirar captura següent)

```
root@MarcPG<Tue Oct 17>:~# cd /media/USB
root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i debian-archive-keyring_2021.1.1+deb11u1_all.deb _
```

Ara ja podem executar les següents comandes en ordre:

apt update

apt full-upgrade

Donarà un error de llibreries i s'ha de descarregar i executar els següents arxius i comandes:

Executar **apt update**

dpkg -i debian-archive-keyring(nom del fitxer que ens haguem descarregat anteriorment)

```
root@francesco0 (Wed Oct 18):</media/usb># dpkg -i debian-archive-keyring_2021.1.1+deb11u1_all.deb
(Reading database ... 30279 files and directories currently installed.)
Preparing to unpack debian-archive-keyring_2021.1.1+deb11u1_all.deb ...
Unpacking debian-archive-keyring (2021.1.1+deb11u1) over (2019.1) ...
Setting up debian-archive-keyring (2021.1.1+deb11u1) ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-jessie-automatic.gpg ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-jessie-security-automatic.gpg ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-jessie-stable.gpg ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-stretch-automatic.gpg ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-stretch-security-automatic.gpg ...
Removing obsolete conf file /etc/apt/trusted.gpg.d/debian-archive-stretch-stable.gpg ...
root@francesco0 (Wed Oct 18):</media/usb># apt update
Get:1 http://deb.debian.org/debian stable InRelease [151 kB]
Get:2 http://ftp.debian.org/debian bullseye InRelease [116 kB]
Get:3 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:4 http://ftp.es.debian.org/debian stable InRelease [151 kB]
N: Repository 'http://deb.debian.org/debian stable InRelease' changed its 'Version' value from '10.1' to '12.2'
E: Repository 'http://deb.debian.org/debian stable InRelease' changed its 'Codename' value from 'buster' to 'bookworm'
N: This must be accepted explicitly before updates for this repository can be applied. See apt-security(8) manpage for details.
Do you want to accept these changes and continue updating from this repository? [y/N] _
```

```

Forcing systemd-logind to load NSS modules...
Unpacking libc6:amd64 (2.36-9+deb12u2) over (2.28-10) ...
Selecting previously unselected package libgcc-s1:amd64.
Preparing to unpack .../libgcc-s1_12.2.0-14_amd64.deb ...
Unpacking libgcc-s1:amd64 (12.2.0-14) ...
Replacing files in old package libgcc1:amd64 (1:8.3.0-6) ...
Setting up libgcc-s1:amd64 (12.2.0-14) ...
Setting up libc6:amd64 (2.36-9+deb12u2) ...
/usr/bin/perl: error while loading shared libraries: libcrypt.so.1: cannot open shared object file:
No such file or directory
dpkg: error processing package libc6:amd64 (--configure):
 installed libc6:amd64 package post-installation script subprocess returned error exit status 127
Errors were encountered while processing:
 libc6:amd64
E: Sub-process /usr/bin/dpkg returned an error code (1)
root@francesco0 (Wed Oct 18):</media/usb>#

```

Continuarà donant errors de dependències de llibreries:

Es torna a executar **apt update**

Executar **apt full-upgrade**

apt -f install

```

root@francesco0 (Wed Oct 18):</media/usb># apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following additional packages will be installed:
 libc-bin libc-l10n
Recommended packages:
 manpages
The following packages will be upgraded:
 libc-bin libc-l10n
2 upgraded, 0 newly installed, 0 to remove and 322 not upgraded.
2 not fully installed or removed.
Need to get 0 B/1280 kB of archives.
After this operation, 1371 kB disk space will be freed.
Do you want to continue? [Y/n] y
/usr/bin/perl: error while loading shared libraries: libcrypt.so.1: cannot open shared object file:
No such file or directory
Setting up libc6:amd64 (2.36-9+deb12u2) ...
/usr/bin/perl: error while loading shared libraries: libcrypt.so.1: cannot open shared object file:
No such file or directory
dpkg: error processing package libc6:amd64 (--configure):
 installed libc6:amd64 package post-installation script subprocess returned error exit status 127
Errors were encountered while processing:
 libc6:amd64
E: Sub-process /usr/bin/dpkg returned an error code (1)

```

Donarà problemes de llibreries del libc6, que ens haurem de descarregar manualment.

Descarregant-nos el paquet d'espàny(es) .udeb de la següent pàgina web:

<https://packages.debian.org/buster/amd64/libc6-udeb/download>

dpkg -i libc6(nom del paquet descarregat)

```

root@francesco0 (Wed Oct 18):</media/usb># dpkg -i libc6-udeb_2.28-10deb10u1_amd64.udeb
Selecting previously unselected package libc6-udeb.
(Reading database ... 30268 files and directories currently installed.)
Preparing to unpack libc6-udeb_2.28-10deb10u1_amd64.udeb ...
Unpacking libc6-udeb (2.28-10+deb10u1) ...
Setting up libc6-udeb (2.28-10+deb10u1) ...
root@francesco0 (Wed Oct 18):</media/usb>#

```

Fer un **apt update**

```
root@francesco0 (Wed Oct 18):</media/usb># apt update
Hit:1 http://ftp.debian.org/debian bullseye InRelease
Hit:2 http://deb.debian.org/debian stable InRelease
Hit:3 http://deb.debian.org/debian bullseye InRelease
Hit:4 http://ftp.es.debian.org/debian stable InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
324 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@francesco0 (Wed Oct 18):</media/usb>#
```

Fer un **apt full-upgrade**

```
root@francesco0 (Wed Oct 18):</media/usb># apt full-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
You might want to run 'apt --fix-broken install' to correct these.
The following packages have unmet dependencies:
 libc-bin : Depends: libc6 (< 2.29)
             Recommends: manpages but it is not installed
 locales : Depends: libc-bin (> 2.36) but 2.28-10 is installed
             Depends: libc-l10n (> 2.36) but 2.28-10 is installed
E: Unmet dependencies. Try 'apt --fix-broken install' with no packages (or specify a solution).
root@francesco0 (Wed Oct 18):</media/usb># _
```

Fer un **apt -f install**

```
root@francesco0 (Wed Oct 18):</media/usb># apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following additional packages will be installed:
 libc-bin libc-l10n
Recommended packages:
 manpages
The following packages will be upgraded:
 libc-bin libc-l10n
2 upgraded, 0 newly installed, 0 to remove and 322 not upgraded.
2 not fully installed or removed.
Need to get 0 B/1280 kB of archives.
After this operation, 1371 kB disk space will be freed.
Do you want to continue? [Y/n] _
```

On s'obrirà un missatge de si es voldran restablir els paquets actualitzats, on l'hem de senyalitzar que si:

```
Restart services during package upgrades without asking?
<Yes> <No>

Restarting services possibly affected by the upgrade:
cron: restarting...done.

Services restarted successfully.
(Reading database ... 30292 files and directories currently installed.)
Preparing to unpack .../libc-bin_2.36-9+deb12u2_amd64.deb ...
Unpacking libc-bin (2.36-9+deb12u2) over (2.28-10) ...
Setting up libc-bin (2.36-9+deb12u2) ...
(Reading database ... 30292 files and directories currently installed.)
Preparing to unpack .../libc-l10n_2.36-9+deb12u2_all.deb ...
Unpacking libc-l10n (2.36-9+deb12u2) over (2.28-10) ...
Setting up libc-l10n (2.36-9+deb12u2) ...
Setting up locales (2.36-9+deb12u2) ...
Installing new version of config file /etc/locale.alias ...
Generating locales (this might take a while)...
Generation complete.
Processing triggers for man-db (2.8.5-2) ...
root@francesco0 (Wed Oct 18):</media/usb>#
```

Executar apt full-upgrade

Fer un dpkg -- configure -a

```
debconf-i18n depends on debconf (= 1.5.82); however:
Package debconf is not configured yet.

dpkg: error processing package debconf-i18n (--configure):
dependency problems - leaving unconfigured
Setting up kmod (30+20221128-1) ...
Installing new version of config file /etc/init.d/kmod ...
/usr/bin/perl: /lib/libcrypt.so.1: version 'XCRYPT_2.0' not found (required by /usr/bin/perl)
dpkg: error processing package kmod (--configure):
installed kmod package post-installation script subprocess returned error exit status 1
Setting up libgdbm-compat4:amd64 (1.23-3) ...
Setting up libperl5.36:amd64 (5.36.0-7) ...
Setting up perl (5.36.0-7) ...
Processing triggers for libc-bin (2.36-9+deb12u2) ...
Processing triggers for systemd (241-7~deb10u1) ...
dpkg: dependency problems prevent processing triggers for man-db:
man-db depends on debconf (>= 1.2.0) | debconf-2.0; however:
Package debconf is not configured yet.
Package debconf-2.0 is not installed.
Package debconf which provides debconf-2.0 is not configured yet.

dpkg: error processing package man-db (--configure):
dependency problems - leaving triggers unprocessed
Processing triggers for initramfs-tools (0.133+deb10u1) ...
update-initramfs: Generating /boot/initrd.img-4.19.0-6-amd64
I: The initramfs will attempt to resume from /dev/sdb3
I: (UUID=daa3c021-3750-4223-b59d-df12131d1355)
I: Set the RESUME variable to override this.
depmod: WARNING: could not open modules.builtin.modinfo at /var/tmp/mkinitramfs_s7dbNq/lib/modules/4.19.0-6-amd64: No such file or directory
Errors were encountered while processing:
debconf
dmsetup
debconf-i18n
kmod
man-db
root@francesco0 (Wed Oct 18):</media/usb># apt -f install_
```

Fer un **apt -f install** per arreglar els errors que surten.

```
root@francesco0 (Wed Oct 18):</media/usb># apt -f install
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following packages were automatically installed and are no longer required:
  libjson-c5 libperl5.28 perl-modules-5.28
Use 'apt autoremove' to remove them.
The following additional packages will be installed:
  libdevmapper1.02.1
The following packages will be upgraded:
  libdevmapper1.02.1
1 upgraded, 0 newly installed, 0 to remove and 291 not upgraded.
4 not fully installed or removed.
Need to get 0 B/133 kB of archives.
After this operation, 1024 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up debconf (1.5.82) ...
(Reading database ... 32334 files and directories currently installed.)
Preparing to unpack .../libdevmapper1.02.1_2%3a1.02.185-2_amd64.deb ...
Unpacking libdevmapper1.02.1:amd64 (2:1.02.185-2) over (2:1.02.155-3) ...
Setting up debconf-i18n (1.5.82) ...
Setting up kmod (30+20221128-1) ...
Setting up libdevmapper1.02.1:amd64 (2:1.02.185-2) ...
Setting up dmsetup (2:1.02.185-2) ...
Processing triggers for libc-bin (2.36-9+deb12u2) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for initramfs-tools (0.133+deb10u1) ...
update-initramfs: Generating /boot/initrd.img-4.19.0-6-amd64
I: The initramfs will attempt to resume from /dev/sdb3
I: (UUID=daa3c021-3750-4223-b59d-df12131d1355)
I: Set the RESUME variable to override this.
depmod: WARNING: could not open modules.builtin.modinfo at /var/tmp/mkinitramfs_x3quvY/lib/modules/4.19.0-6-amd64: No such file or directory
root@francesco0 (Wed Oct 18):</media/usb># apt full-upgrade
```

Fer un **apt full-upgrade**

```
Setting up python3 (3.11.2-1+b1) ...
Running python3 rupdate hooks for python3.11...
Running python3 post-rupdate hooks for python3.11...
Setting up imagemagick (8:6.9.11.60+dfsg-1.6) ...
Setting up libgs10:amd64 (10.0.0~dfsg-11+deb12u2) ...
Setting up python3-libapparmor (3.0.8-3) ...
Setting up ghostscript (10.0.0~dfsg-11+deb12u2) ...
Setting up python3-apparmor (3.0.8-3) ...
Setting up apparmor-utils (3.0.8-3) ...
Installing new version of config file /etc/apparmor/logprof.conf ...
Installing new version of config file /etc/apparmor/severity.db ...
Setting up tasksel-data (3.73) ...
Setting up libruby:amd64 (1:3.1) ...
Setting up tasksel (3.73) ...
Setting up ruby3.1 (3.1.2-7) ...
Setting up ruby (1:3.1) ...
Setting up rake (13.0.6-3) ...
Setting up libruby3.1:amd64 (3.1.2-7) ...
Setting up ruby-rubygems (3.3.15-2) ...
Setting up ruby-sdbm:amd64 (1.0.0-5+b1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.36-9+deb12u2) ...
Processing triggers for systemd (252.17-1~deb12u1) ...
Processing triggers for debianutils (5.7-0.5~deb12u1) ...
Processing triggers for ca-certificates (20230311) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for initramfs-tools (0.142) ...
update-initramfs: Generating /boot/initrd.img-6.1.0-13-amd64
I: The initramfs will attempt to resume from /dev/sdb3
I: (UUID=daa3c021-3750-4223-b59d-df12131d1355)
I: Set the RESUME variable to override this.
root@francesco0 (Wed Oct 18):</media/usb># _
```

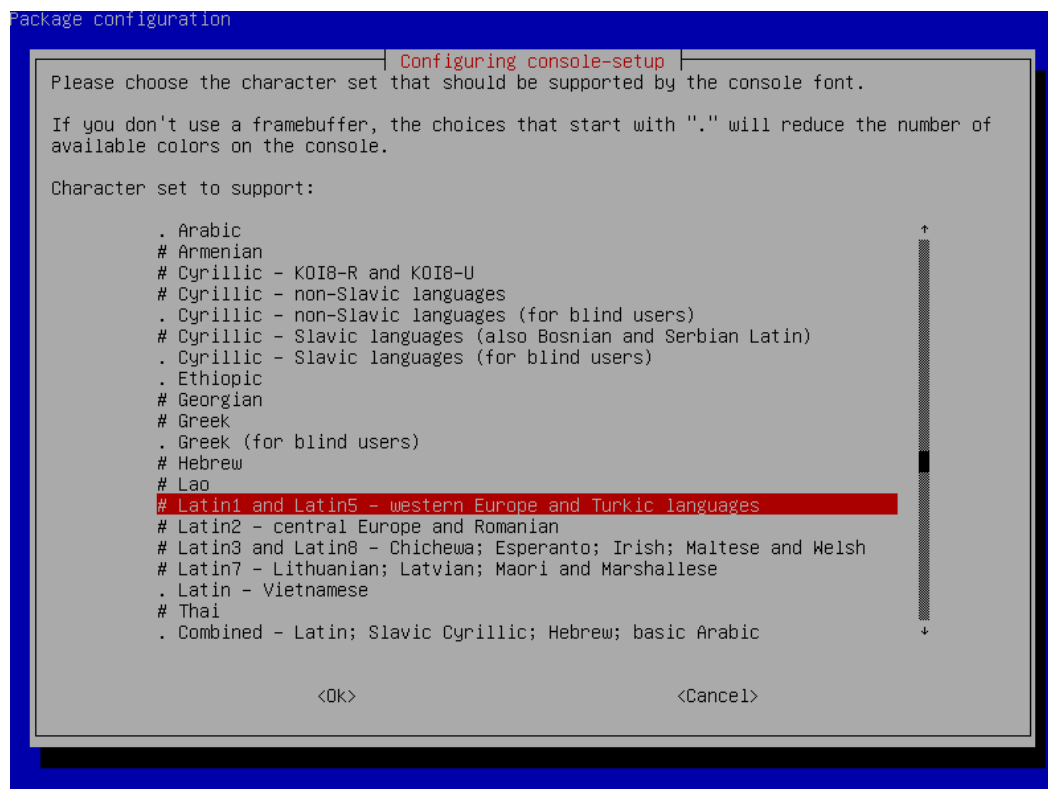
Finalment fer un **apt-get install sudo** i un **apt-get install make**

```

Get:1 http://deb.debian.org/debian stable/main amd64 sudo amd64 1.9.13p3-1+deb12u1 [1889 kB]
Fetched 1889 kB in 0s (8934 kB/s)
Selecting previously unselected package sudo.
(Reading database ... 42723 files and directories currently installed.)
Preparing to unpack .../sudo_1.9.13p3-1+deb12u1_amd64.deb ...
Unpacking sudo (1.9.13p3-1+deb12u1) ...
Setting up sudo (1.9.13p3-1+deb12u1) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u2) ...
root@francesco0 (Wed Oct 18):</media/usb># apt-get install make
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bsdmainutils libcupsfilters1 libcupsimage2 libexif12 libflac8 libgs9-common libicu63
  libilmbase23 libldap-2.4-2 libldap-common libmpdec2 libnetpbm10 libopenexr23 libperl5.28
  libpython3.7-minimal libpython3.7-stdlib libreadline7 libruby2.5 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libtiff5 libwebp6 libwmf-0.2-7 libwmf0.2-7 libwrap0 libx265-165 libxi6
  libxtst6 ncal perl-modules-5.28 pigz python3.7 python3.7-minimal ruby-did-you-mean ruby-minitest
  ruby-power-assert ruby-test-unit
Use 'apt autoremove' to remove them.
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 396 kB of archives.
After this operation, 1630 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian stable/main amd64 make amd64 4.3-4.1 [396 kB]
Fetched 396 kB in 0s (3078 kB/s)
Selecting previously unselected package make.
(Reading database ... 42864 files and directories currently installed.)
Preparing to unpack .../make_4.3-4.1_amd64.deb ...
Unpacking make (4.3-4.1) ...
Setting up make (4.3-4.1) ...
Processing triggers for man-db (2.11.2-2) ...
root@francesco0 (Wed Oct 18):</media/usb># _

```

Al acabar el **full-upgrade**, sortirà aquesta pantalla:



Clicarem enter a l'opció que tenim marcada per defecte.

3. Introducció a la Instal·lació d'aplicacions

El sistema *X-window* (o *X11* o *X*) és un protocol de visualització que proveeix un conjunt d'eines estàndard per construir interfícies d'usuari gràfiques (GUI). *X* prové el marc de referència bàsic però no defineix la interfície d'usuari, aquesta es deixa als programes clients. A més a més, *X* fa ús d'un model client servidor amb el qual el servidor *X* es comunica, localment o en xarxa, amb programes clients. El servidor accepta sol·licituds de sortida gràfica (finestres) i envia de tornada a l'usuari les entrades (del teclat, ratolí, o altres).

El sistema *X* no conté especificacions de la interfície d'usuari de les aplicacions com poden ser: botons, menús, etc. En lloc d'això el software d'usuari s'encarrega de l'aparença de les finestres. Per donar una aparença semblant a totes les finestres i aplicacions gràfiques, existeixen programes com els gestors de finestres (*window-managers*) o els ambients d'escriptori (*desktop-environments*)

Hi ha diferents implementacions del sistema *X-window* per Linux (i altres sistemes UNIX). La més comú i que nosaltres utilitzarem és diu *X.org*.

A més del servidor *X* són necessaris altres paquets per tal de tenir un entorn gràfic complet com són els gestors de finestres, de pantalles i els ambients d'escriptori.

- **Gestor de finestres:** (*window manager*) és l'encarregat de controlar la ubicació i aparença de les finestres de les aplicacions gràfiques. Hi ha molts gestors de finestres amb diferents funcionalitats. Podem mencionar alguns com el Kwin, gnome-shell.

- **Gestor de pantalla** (*display manager*): permet iniciar una sessió al servidor *X* des del mateix ordinador o des d'un altre ordinador mitjançant la xarxa. El gestor de pantalla presenta a l'usuari una pantalla de login i fa la validació de la contrasenya, per tant realitza funcions semblats als programes *init*, *getty* i *login* per a terminals de modo caràcter. Alguns gestors de pantalles comuns són: *XDM* (*X Window Display Manager*), *GDM* (*GNOME Display Manager*), i *SDDM*(*KDE Display Manager*).

El gestor de pantalla és un servei que es pot arrancar i aturar com la resta de serveis del sistema mitjançant scripts d'inici utilitzant la comanda *systemctl*.

- **Ambient d'escriptori** (*Desktop environment*) Ofereix una interfície d'usuari unificada per a les aplicacions gràfiques amb icones, barres d'eines, fondos, etc. Normalment l'ambient d'escriptori consisteix en un gestor de finestres, un gestor de pantalla i un conjunt propi d'aplicacions i llibreries. Els ambients d'escriptori més habituals són el *GNOME* i el *KDE* però n'hi ha molts més.

Desktop Environment	Window manager	Display manager	Graphical library
GNOME	gnome-shell	GDM	GTK+
KDE	Kwin	KDM	QT

<i>Xfce</i>	Xfwm4	LightDM	GTK+
<i>LXDE</i>	Openbox	LXDM	QT

Desktop Environment i els seus corresponent *Window manager* i *Display manager*

4. Instal·lació de paquets binaris

4.1. Instal·lació manual

Volem instal·lar l'aplicació **make** al nostre sistema. Primerament necessitarem aconseguir el software a instal·lar. Els paquets que pugueu anar necessitant els teniu a <https://packages.debian.org/stable/>

Consulteu l'ajuda (man del sistema o pàgina de manual) de la comanda dpkg i completeu la següent taula:

Acció	Opció	Arguments
instal·lar un paquet	dpkg -i o --install	nom_del_paquet.deb
desinstal·lar un paquet	dpkg -r	nom_del_paquet
purgar un paquet	dpkg -P	nom_del_paquet
l·listar paquets	dpkg - - list	--
l·listar fitxers d'un paquet	dpkg-deb -c	nom_del_paquet.deb
trobar a quin paquet pertany un fitxer	dpkg -S	/ruta/al/arxiu

Baixeu el paquet corresponent per instal·lar **make**

Farem l'explicació de com baixar paquets .deb de la pàgina debian amb l'exemple de make.

Entrem al següent link: <https://www.debian.org/distrib/packages>

Baixem cap abaix fins trobar el següent apartat:

Búsqueda en los directorios de paquetes

Palabra clave:

Buscar en: ☒ Sólo nombres de paquetes ☐ Descripciones ☐ Nombres de paquetes fuente

Sólo mostrar coincidencias exactas: ☐

Distribución: Sección:

Dispone de atajos para algunas búsquedas:

Escrivim el nom del paquet que volem, en aquest cas **make**

Ens portara a la següent pagina on tindrem dos apartats els Exact hits que es el que ens interessa, però en cas de no trobar el paquet que volem exacte, potser el trobem a Other hits.

Please consider using a longer keyword or more keywords.

Exact hits

Package make

- [bookworm \(stable\)](#) (devel): utility for directing compilation
4.3-4.1: amd64 arm64 armel armhf i386 mips64el mipsel ppc64el s390x

Other hits

Fem click al link i ens porta aquí, on hem de seleccionar segons la nostre versió. En el nostre cas **amd64**

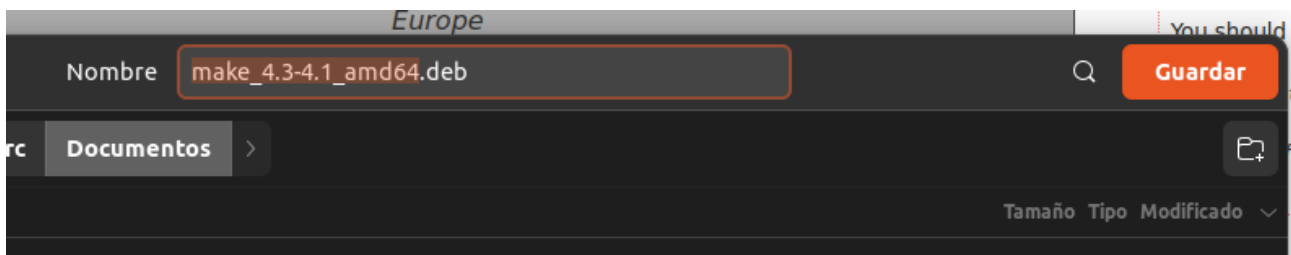
Download make

Architecture	Package Size	Installed Size	Files
amd64	386.4 kB	1,592.0 kB	[list of files]
arm64	382.0 kB	1,596.0 kB	[list of files]
armel	375.2 kB	1,566.0 kB	[list of files]
armhf	375.4 kB	1,514.0 kB	[list of files]
i386	394.2 kB	1,605.0 kB	[list of files]
mips64el	379.1 kB	1,621.0 kB	[list of files]
mipsel	379.7 kB	1,601.0 kB	[list of files]
ppc64el	394.8 kB	1,684.0 kB	[list of files]
s390x	380.1 kB	1,608.0 kB	[list of files]

Com hem fet abans de fer el apt update, descargem el .deb del link <ftp.es.debian.org/debian>

- <ftp.cz.debian.org/debian>
- <ftp.dk.debian.org/debian>
- <ftp.ee.debian.org/debian>
- <ftp.es.debian.org/debian>
- <ftp.fi.debian.org/debian>
- <ftp.fr.debian.org/debian>
- <ftp.kn.debian.org/debian>

Guardem l'arxiu al usb i si no tenim el usb preparat a la maquina haurem de seguir els passos del següent apartat: [Actualització del kernel](#)



Una vegada fem tot el procés de mount del usb podrem fer els següents passos

```
cd /media/NomUsb
ls (per veure els arxius)
dpkg -i arxiu_make.deb
```

```
root@MarcPG<Tue Oct 17>:~# cd /media/USB
root@MarcPG<Tue Oct 17>:/media/USB# ls
'System Volume Information'  debian-archive-keyring_2021.1.1+deb11u1_all.deb
adso-install-10-64-N.tar.gz  make_4.3-4.1_amd64.deb
root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i make_4.3-4.1_amd64.deb_
```

Comprovem que tenim **make** instal·lat correctament executant la comanda make.

```
root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i make_4.3-4.1_amd64.deb
Selecting previously unselected package make.
(Reading database ... 51760 files and directories currently installed.)
Preparing to unpack make_4.3-4.1_amd64.deb ...
Unpacking make (4.3-4.1) ...
Setting up make (4.3-4.1) ...
Processing triggers for man-db (2.11.2-2) ...
root@MarcPG<Tue Oct 17>:/media/USB# cd
root@MarcPG<Tue Oct 17>:~# make
make: *** No targets specified and no makefile found.  Stop.
root@MarcPG<Tue Oct 17>:~#
```

Ón és el paquet? Quin paquet heu baixat?

(apt-get download make descarrega l'arxiu .deb del paquet make en el directori actual)

(sudo dpkg -i nom_del_arxiu.deb instala el paquet make al nostre sistema)

La comanda **dpkg** a més de permetre instal·lar paquets serveix també per obtenir informació sobre els paquets i fitxers instal·lats al sistema, i desinstal·lar paquets.

Feu servir la comanda per instal·lar make.

Llegiu els missatges que surten durant el procés i assegureu-vos que no hi ha hagut cap problema.

Quina diferència hi ha entre desinstal·lar un paquet i purgar un paquet?

Quan es **desinstal·la un paquet** s'està eliminant tot el programa i l'aplicació d'aquest, però es retenen alguns arxius de la seva configuració, aquests es guarden en el sistema per si en algun futur és tornar a reinstalar el mateix paquet i no es perdi la configuració personalitzada actual. Però si es fa una **purga d'un paquet**, s'elimina el programa i tots els seus arxius de configuració, deixant el sistema com si abans q s'hagués instal·lat aquest paquet.

Ara volem instal·lar els programes **lynx** (un navegador web per mode text) i **lftp** (un client avançat de ftp). Baixeu els paquets corresponents i instal·leu-los amb **dpkg**.

Seguirem els mateixos passos que per la instal·lació de make

Entrem en aquest enllaç <https://www.debian.org/distrib/packages>

Busquem i descarregem els paquets .deb

Una vegada a la nostre VM executem les següents comandes

```
root@MarcPG<Tue Oct 17>:~# cd /media/USB
root@MarcPG<Tue Oct 17>:/media/USB# ls
'System Volume Information'          lftp_4.9.2-2+b1_amd64.deb
adso-install-10-64-N.tar.gz          lynx_2.9.0dev.12-1_amd64.deb
debian-archive-keyring_2021.1.1+deb11u1_all.deb  make_4.3-4.1_amd64.deb
root@MarcPG<Tue Oct 17>:/media/USB#
```

Veiem que tenim correctament els paquets per instal·lar.

Instalem primer **lftp**

```

root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i lftp_4.9.2-2+b1_amd64.deb
Selecting previously unselected package lftp.
(Reading database ... 51808 files and directories currently installed.)
Preparing to unpack lftp_4.9.2-2+b1_amd64.deb ...
Unpacking lftp (4.9.2-2+b1) ...
Setting up lftp (4.9.2-2+b1) ...
Processing triggers for mailcap (3.70+nmu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for man-db (2.11.2-2) ...

```

Posteriorment instalem **lynx** i en executar la comanda veiem que tenim un error de dependències, per tant haurem d'anar al enllaç on em descarregat el .deb i mirar si hi ha la dependència que ens demanen.

```

root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i lynx_2.9.0dev.12-1_amd64.deb
(Reading database ... 51860 files and directories currently installed.)
Preparing to unpack lynx_2.9.0dev.12-1_amd64.deb ...
Unpacking lynx (2.9.0dev.12-1) over (2.9.0dev.12-1) ...
dpkg: dependency problems prevent configuration of lynx:
 lynx depends on lynx-common; however:
  Package lynx-common is not installed.

dpkg: error processing package lynx (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 lynx
root@MarcPG<Tue Oct 17>:/media/USB#

```

Una vegada tenim la dependència al usb, la instalem primer per posteriorment instal·lar **lynx**

```

root@MarcPG<Tue Oct 17>:/media/USB# ls
'System Volume Information'      lynx-common_2.9.0dev.12-1_all.deb
adso-install-10-64-N.tar.gz      lynx_2.9.0dev.12-1_amd64.deb
debian-archive-keyring_2021.1.1+deb11u1_all.deb  make_4.3-4.1_amd64.deb
lftp_4.9.2-2+b1_amd64.deb
root@MarcPG<Tue Oct 17>:/media/USB#
root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i lynx-common_2.9.0dev.12-1_all.deb
Selecting previously unselected package lynx-common.
(Reading database ... 51860 files and directories currently installed.)
Preparing to unpack lynx-common_2.9.0dev.12-1_all.deb ...
Unpacking lynx-common (2.9.0dev.12-1) ...
Setting up lynx-common (2.9.0dev.12-1) ...
Processing triggers for mailcap (3.70+nmu1) ...
Processing triggers for man-db (2.11.2-2) ...
root@MarcPG<Tue Oct 17>:/media/USB# dpkg -i lynx_2.9.0dev.12-1_amd64.deb
(Reading database ... 51958 files and directories currently installed.)
Preparing to unpack lynx_2.9.0dev.12-1_amd64.deb ...
Unpacking lynx (2.9.0dev.12-1) over (2.9.0dev.12-1) ...
Setting up lynx (2.9.0dev.12-1) ...
update-alternatives: using /usr/bin/lynx to provide /usr/bin/www-browser (www-browser) in auto mode
root@MarcPG<Tue Oct 17>:/media/USB#

```

Executeu les comandes **lynx** i **lftp** per assegurar-vos que funcionen correctament.

LFTP

```
root@MarcPG<Tue Oct 17>:/media/USB# lftp
lftp :~> _
```

LYNX

```

About Lynx - Who, What, and When - Where it is now (p1 of 2)
[ _About Lynx-Dev | Lynx-Dev Archives ]

About Lynx

Lynx is a fully-featured World Wide Web (WWW) browser for users on Unix, VMS, and other
platforms running cursor-addressable, character-cell terminals or emulators. That includes
vt100 terminals, other character-cell displays, and vt100 emulators such as Kermit or
Procomm running on PCs or Macs.

For information on how to use Lynx see the Lynx User's Guide, or the Lynx help files.

Credits and Copyright

Lynx was a product of the Distributed Computing Group within Academic Computing Services of
The University of Kansas.

Lynx was originally developed by Lou Montulli, Michael Grobe, and Charles Rezac. Garrett
Blythe created DosLynx and later joined the Lynx effort as well. Following the departures
of Lou and Garrett for positions at Netscape in the summer of 1994, Craig Lavender provided
support services for Lynx, and Ravikumar Kolli for DosLynx.

Lynx is maintained and supported by members of the Internet community coordinated via the
lynx-dev mailing list.

Lynx is derived from material copyrighted by the University of Kansas. However most of the
release (and corresponding copyright) is the work of developers on the lynx-dev mailing
list. It is distributed without restrictions on usage or redistribution under the GNU
General Public License (Version 2).

Lynx was built over an early version of the Common Code Library developed by the CERN WWW

```

4.2. Instal·lació amb un gestor de paquets

Per facilitar la instal·lació d'aplicacions grans (que solen tenir moltes dependències) i també que sigui més fàcil mantenir els sistemes actualitzats varen començar a aparèixer els gestors de paquets.

Debian té un conjunt d'eines, anomenades **Apt** (Advanced front-end for dpkg), que permeten cercar, descarregar i instal·lar software i totes les seves dependències i mantenir el sistema actualitzat d'una manera senzilla i còmoda. Existeixen també diversos front-ends gràfics (synaptic, adept, ...) que no farem servir.

Explica les diferències entre **apt-get**, **apt** i **aptitude**

apt-get: és una eina per la gestió de paquets en el sistema Debian, que dóna permís per instal·lar, actualitzar, eliminar i gestionar paquets.

apt: és una interfície de Debian que ajuda en la gestió de paquets fent-ho més fàcil i accessible donant

una sortida més llegible en la seva execució, a diferència de **apt-get** que pot resultar més complicada i difícil d'entendre.

aptitude és una eina per la gestió de paquets en el sistema Debian, molt més simple d'utilitzar que les anteriors amb una interfície en mode text per fer-ho més llegible i més fàcil a la línia de comandes. Una millora a les eines **apt-get** i **apt** és que **aptitude** resol possibles dependències de paquets de manera més eficient i permet una gestió de paquets més detallada si hi ha problemes amb conflictes de dependències.

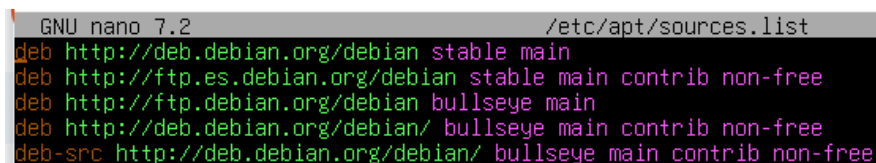
4.2.1 Configuració dels repositoris de software

Primerament haurem de configurar quins són els repositoris d'on APT pot obtenir els paquets **.deb** per instal·lar al sistema. Aquests repositoris poden estar a servidors remots o fins i tot en el nostre servidor (p.ex. en un cdrom) i en podem tenir tants com vulguem.

Els fitxers de configuració d'APT es troben a **/etc/apt**. Dins aquest directori crearem (si no està creat) el fitxer **sources.list** amb el següent contingut:

```
deb http://ftp.es.debian.org/debian/ stable main non-free contrib
```

Obrirem amb la comanda nano fitxer source.list, que anteriorment ha quedat així:



```
GNU nano 7.2 /etc/apt/sources.list
deb http://deb.debian.org/debian stable main
deb http://ftp.es.debian.org/debian stable main contrib non-free
deb http://ftp.debian.org/debian bullseye main
deb http://deb.debian.org/debian/ bullseye main contrib non-free
deb-src http://deb.debian.org/debian/ bullseye main contrib non-free
```

4.2.2 Llista de paquets disponibles als repositoris

Ara, hem de fer que el sistema obtingui la llista de paquets disponibles als repositoris i la informació relacionada amb ells. Ho farem mitjançant la següent comanda:

```
# apt-get update
```

```

root@francesco0 (Wed Oct 18):</etc/apt># apt-get update
Get:1 http://ftp.debian.org/debian bullseye InRelease [116 kB]
Hit:2 http://deb.debian.org/debian stable InRelease
Get:3 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:4 http://ftp.es.debian.org/debian stable InRelease [151 kB]
Get:5 http://ftp.debian.org/debian bullseye/main amd64 Packages [8062 kB]
Get:6 http://deb.debian.org/debian bullseye/contrib Sources [43.2 kB]
Get:7 http://deb.debian.org/debian bullseye/non-free Sources [81.0 kB]
Get:8 http://deb.debian.org/debian bullseye/main Sources [8502 kB]
Get:9 http://ftp.es.debian.org/debian stable/main amd64 Packages [8780 kB]
Get:10 http://ftp.es.debian.org/debian stable/main Translation-en [6110 kB]
Get:11 http://ftp.es.debian.org/debian stable/contrib amd64 Packages [54.1 kB]
Get:12 http://ftp.es.debian.org/debian stable/contrib Translation-en [48.7 kB]
Get:13 http://ftp.es.debian.org/debian stable/non-free amd64 Packages [96.9 kB]
Get:14 http://ftp.es.debian.org/debian stable/non-free Translation-en [66.8 kB]
Get:15 http://ftp.debian.org/debian bullseye/main Translation-en [6236 kB]
Get:16 http://deb.debian.org/debian bullseye/main amd64 Packages [8062 kB]
Get:17 http://deb.debian.org/debian bullseye/main Translation-en [6236 kB]
Get:18 http://deb.debian.org/debian bullseye/contrib amd64 Packages [50.4 kB]
Get:19 http://deb.debian.org/debian bullseye/contrib Translation-en [46.9 kB]
Get:20 http://deb.debian.org/debian bullseye/non-free amd64 Packages [96.3 kB]
Get:21 http://deb.debian.org/debian bullseye/non-free Translation-en [92.4 kB]
Fetched 53.0 MB in 7s (7438 kB/s)
Reading package lists... Done
root@francesco0 (Wed Oct 18):</etc/apt># _

```

L'eina **apt-get** serveix també, entre altres coses, per instal·lar paquets, desinstal·lar-los i actualitzar tots els paquets del sistema si hi ha noves versions disponibles.

Amb quina comanda actualitzem els paquets instal·lats a l'última versió disponible?

Amb la comanda **sudo apt-get upgrade**: actualitza tots els paquets instal·lats actualment a les seves últimes versions. Ho fa actualitzant els paquets a la versió més recent disponible als dipòsits.

Aquestes són altres comandes de l'eina apt-get molt útils:

sudo apt-get update: aquesta comanda descarrega les llistes de paquets dels repositoris i les actualitza per obtenir informació sobre les versions més noves dels paquets i les seves dependències. No actualitza els paquets instal·lats.

sudo apt-get remove package_name: substituint **package_name** pel nom del paquet que es vol desinstal·lar, eliminarà el paquet del sistema.

sudo apt-get purge nom_paquet: permet desinstal·lar un paquet i eliminar els seus fitxers de configuració.

Comproveu que tots els paquets estan actualitzats a la seva última versió.


```

root@MarionaF (Wed Oct 18):</media/usb># sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  bsdmainutils libcupsfilters1 libcupsimage2 libexif12 libflac8 libgs9-common libicu63
  libilmbase23 libldap-2.4-2 libldap-common libmpdec2 libnetpbm10 libopenexr23 libperl5.28
  libpython3.7-minimal libpython3.7-stdlib libreadline7 libruby2.5 libsasl2-2 libsasl2-modules
  libsasl2-modules-db libtiff5 libwebp6 libwmf-0.2-7 libwmf0.2-7 libwrap0 libx265-165 libxi6
  libxtst6 ncal perl-modules-5.28 pigz python3.7 python3.7-minimal ruby-did-you-mean ruby-minitest
  ruby-power-assert ruby-test-unit
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@MarionaF (Wed Oct 18):</media/usb>#

```

4.2.3 Instal·lació del sistema X-Window

Ara, feu servir **apt-get** per instal·lar un servidor d'X. El paquet que heu d'instal·lar es diu **x-window-system** o **xorg**. Observeu com s'instal·len també totes les dependències i us fa les preguntes necessàries per configurar el servidor d'X.

Quina comanda heu fet servir?

```

root@MarionaF (Wed Oct 18):</media/usb># sudo apt-get install x-window-system_

```

A més d'un servidor d'X fa falta un gestor de finestres (window manager) i un ambient d'escriptori (desktop environment). Si no en coneixem cap podem buscar-ne un a la base de dades de paquets.

Exemples:

- KDE → task-kde-desktop
- Gnome → task-gnome-desktop
- **Xfce4** → **task-xfce-desktop**
- Lxde → task-lxde-desktop

Una altra eina interessant d'APT és **apt-cache** que ens deixa fer consultes a la informació que el sistema ha obtingut dels repositoris en fer l'update. Fent servir **apt-cache** trobareu tots els ambients d'escriptori que tenim disponibles per instal·lar al sistema.

Quina comanda heu fet servir? Anoteu alguns dels ambients d'escriptori que heu trobat

Hem pogut fer la comanda directament amb el **sudo apt-get install** i el nom del ambient que haguem triat.

Si voleu obtenir més informació d'un paquet concret (una descripció, la seves dependències, etc.) podeu fer servir:

```
# apt-cache show nom_paquet
```

4.2.4 gestor de finestres i ambient d'escriptori

Escolliu un gestor de finestres i un ambient d'escriptori i instal·leu-los

-XFCE4

```
root@MarionaF (Wed Oct 18):</media/usb># sudo apt-get install xfce4
```

De vegades la configuració per defecte d'un paquet no funciona i és perquè un error pot fer malbé els fitxers de configuració d'aquest paquet. En aquests casos és necessari reconfigurar el paquet i generar novament els seus fitxers de configuració. El sistema **dpkg** té una comanda per fer això:

dpkg-reconfigure nom-paquet

Si teniu problemes amb la configuració del sistema gràfic podeu utilitzar aquesta comanda per reconfigurar el servidor X.

4.2.5 Instal·lació de paquets

Instal·leu els següents paquets: **gcc** (compilador), **libc6-dev** (llibries de desenvolupament) i **iceweasel** (navegador web Firefox)

Quan acabeu executeu la següent comanda:

Per instal·lar el gcc:

apt-get update

(potser necessitem fer també un apt-get upgrade)

install build-essential (o manualment apt-get install manpages-dev)

Podem comprovar que sha instal·lat correctament mirant la versió : gcc -v

```

root@MarionaF (Wed Oct 18):</media/usb># gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/12/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none:amdgc-n-amdhsa
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Debian 12.2.0-14' --with-bugurl=file:///usr/share/doc/gcc-12/README.Bugs --enable-languages=c,ada,c++,go,d,fortran,objc,obj-c++,m2 --prefix=/usr --with-gcc-major-version-only --program-suffix=-12 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --enable-clocale=gnu --enable-libstdc++-debug --enable-libstdc++-time --enable-default-libstdc++-abi=new --enable-gnu-unique-object --disable-vtable-verify --enable-plugin --enable-default-pie --with-system-zlib --enable-libphobos-checking=release --with-target-system-zlib=auto --enable-objc-gc=auto --enable-multiarch --disable-werror --enable-cet --with-arch-32=i686 --with-abi=m64 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enable-offload-targets=nvptx-none=/build/gcc-12-bTRW0B/gcc-12-12.2.0/debian/tmp-nvptx/usr,amdgc-n-amdhsa=/build/gcc-12-bTRW0B/gcc-12-12.2.0/debian/tmp-gcn/usr --enable-offload-defaulted --without-cuda-driver --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 12.2.0 (Debian 12.2.0-14)

```

- instal·lació libc6-dev:
 - a. Posem: `sudo apt install libc6-dev`
- instal·lació iceweasel:
 - b. No he pogut trobar version del iceweasel, però com alternativa instalem el firefox amb: `apt install firefox-esr`

```

Processing triggers for desktop-file-utils (0.26-1) ...
root@MarionaF (Wed Oct 18):</media/usb># firefox-esr -v
Mozilla Firefox 102.15.1esr

```

apt-get clean

Què fa aquesta comanda? Quina diferència hi ha amb `apt-get autoclean`?

apt-get clean Elimina tots els arxius descarregats en la cache, sense importar si aquest estan en ús o no i si estan instal·lats o ja no es fan servir. En comparació **apt-get autoclean** elimina només els arxius de la caché que ja no poden ser descarregats perquè existeix una versió nova disponible, fent que sigui útil si es vol reinstalar o actualitzar aquests paquets en un futur.

Finalment fem un reboot de la maquina i comprovem que finalment s'ha instal·lat l'entorn gràfic.

4.3. Instal·lació de binaris pre-compilats

De vegades, volem instal·lar un software que no està (pel motiu que sigui) al llistat de paquets dels nostres repositoris.

Volem instal·lar varies versions de la **Java Development Kit(JDK)**. Per fer-ho baixeu del servidor **oracle.com** els fitxers d'instal·lació de **java** que trobareu al **Java Archive Downloads - Java SE 5 i Java Archive Downloads - Java SE 6**

Podem aconseguir l'arxiu des de la pàgina web.

Escriurem al buscador d'oracle "java archive downloads", i buscarem el java SE5 i el java SE6. Descarregarem les versions: **jdk-6u45-linux-x64.bin**.

wget <https://www.oracle.com/java/technologies/javase-archive/javase6u45-downloads-1592236.html>

<https://www.oracle.com/es/java/technologies/javase-java-archive-javase6-downloads.html>

<https://repo.huaweicloud.com/java/jdk/6u45-b06/>

Inicialment volem instal·lar la versió 1.6 (**jdk-6u45-linux-x64.bin**) a **/opt/java1.6**. Per descomprimir cadascun dels fitxers només cal que l'executeu.

Quines comandes heu fet servir per descomprimir el fitxer? A quin directori us ho ha descomprimit?

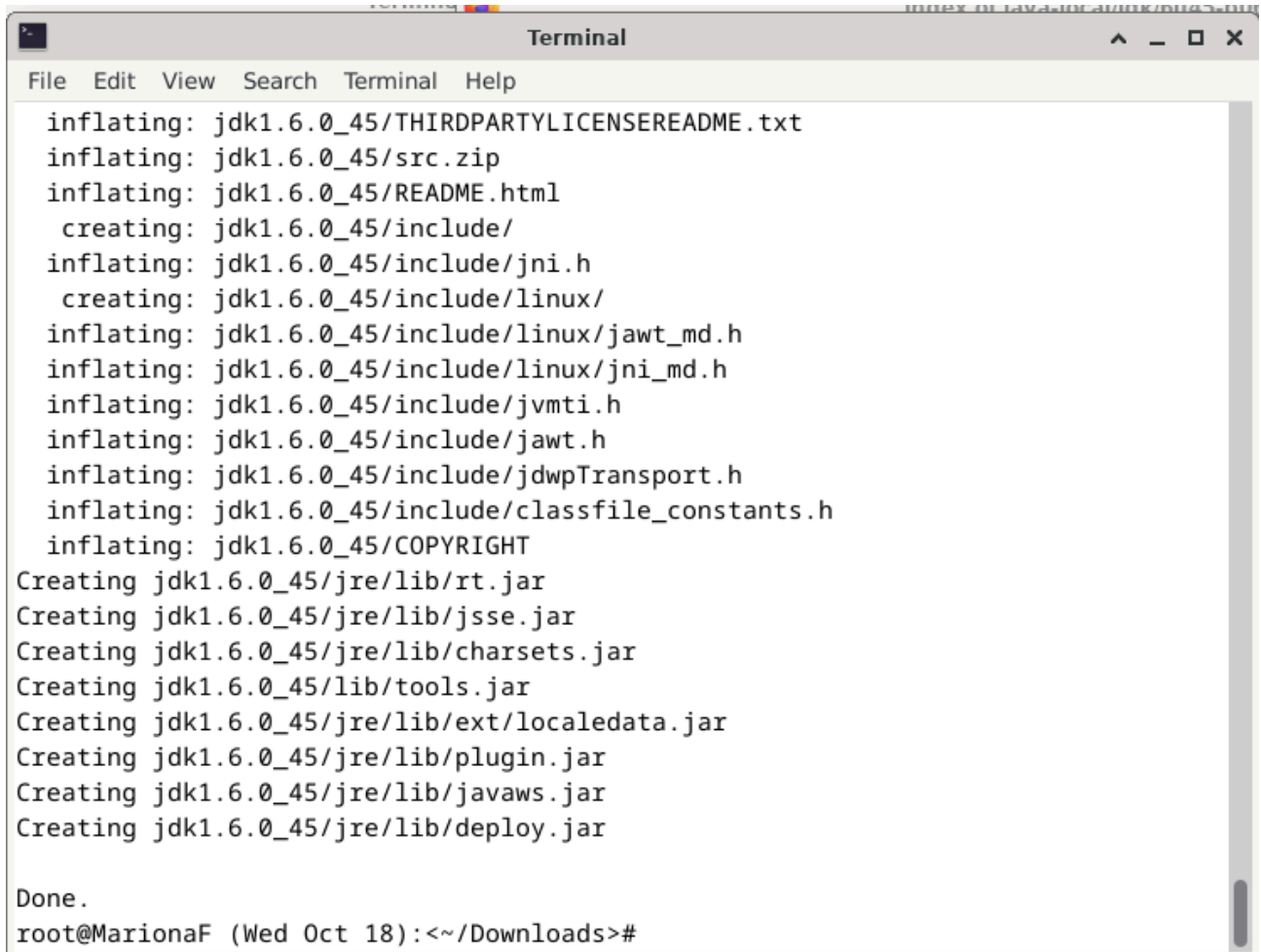
Per descomprimir el fitxer, primer haurem de donar permisos amb:

chmod +x jdk-6u45-linux-x64.bin

```
root@MarionaF (Wed Oct 18):<~># cd Downloads/
root@MarionaF (Wed Oct 18):<~/Downloads># ls
jdk-6u45-linux-x64.bin
root@MarionaF (Wed Oct 18):<~/Downloads># chmod +x jdk-6u45-linux-x64.bin
root@MarionaF (Wed Oct 18):<~/Downloads>#
```

Un cop tenim els permisos, l'executarem per descomprimirlo amb la comanda:

./jdk-6u45-linux-x64.bin



```

Terminal
File Edit View Search Terminal Help
inflating: jdk1.6.0_45/THIRDPARTYLICENSEREADME.txt
inflating: jdk1.6.0_45/src.zip
inflating: jdk1.6.0_45/README.html
  creating: jdk1.6.0_45/include/
inflating: jdk1.6.0_45/include/jni.h
  creating: jdk1.6.0_45/include/linux/
inflating: jdk1.6.0_45/include/linux/jawt_md.h
inflating: jdk1.6.0_45/include/linux/jni_md.h
inflating: jdk1.6.0_45/include/jvmti.h
inflating: jdk1.6.0_45/include/jawt.h
inflating: jdk1.6.0_45/include/jdwpTransport.h
inflating: jdk1.6.0_45/include/classfile_constants.h
inflating: jdk1.6.0_45/COPYRIGHT
Creating jdk1.6.0_45/jre/lib/rt.jar
Creating jdk1.6.0_45/jre/lib/jsse.jar
Creating jdk1.6.0_45/jre/lib/charsets.jar
Creating jdk1.6.0_45/lib/tools.jar
Creating jdk1.6.0_45/jre/lib/ext/localedata.jar
Creating jdk1.6.0_45/jre/lib/plugin.jar
Creating jdk1.6.0_45/jre/lib/javaws.jar
Creating jdk1.6.0_45/jre/lib/deploy.jar

Done.
root@MarionaF (Wed Oct 18):<~/Downloads>#

```

Mireu el contingut i localitzeu on es troba l'executable java que executa la màquina virtual. Ara moveu tot el directori (no només el bin) en què s'ha descomprimit al directori de destí (/opt/java1.6).

Ara ja només hem de moure els fitxers a la carpeta /opt/java1.6 amb la comanda:

```
mv jdk1.6.0_45 /opt/java1.6
```

```
root@MarionaF (Wed Oct 18):<~/Downloads># mv jdk1.6.0_45 /opt/java1.6
```

Verifiquem que està correctament instal·lada:

```
# /opt/java1.6/bin/java -version
```

```

root@MarionaF (Wed Oct 18):<~/Downloads># /opt/java1.6/bin/java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)

```

Ara repetiu aquest passos per les **JDK 1.5** en instal·leu-les a **/opt/java1.5**.

```
root@MarionaF (Wed Oct 18):<~/Downloads># alien jdk-1_5_0_22-linux-amd64.bin
Unknown type of package, jdk-1_5_0_22-linux-amd64.bin.
root@MarionaF (Wed Oct 18):<~/Downloads># alien jdk-1_5_0_22-linux-amd64.rpm
```

Si intentem esbrinar quina és la versió per defecte fent:

java -version

```
root@MarionaF (Wed Oct 18):<~/Downloads># java -version
bash: java: command not found
```

Hi ha errors? A què és degut?

Si, ja que no hem definit la variable **path** per a dir quina versió utilitzarem per defecte.

export PATH=/opt/java1.6/bin:\$PATH

```
root@MarionaF (Wed Oct 18):<~/Downloads># export PATH=/opt/java1.6/bin:$PATH
root@MarionaF (Wed Oct 18):<~/Downloads># java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) 64-Bit Server VM (build 20.45-b01, mixed mode)
```

Per solucionar aquest problema el més senzill és fer un softlink des d'un dels directoris que es troben al **PATH** cap al binari que volem que sigui accessible. Feu un softlink de **/usr/bin/java** a **/opt/java1.6/bin/java**. Amb quina comanda ho aconseguíu?

Per crear el softlink utilitzarem la comanda:

sudo ln -s /opt/java1.6/bin/java /usr/bin/java

```
root@MarionaF (Wed Oct 18):<~/Downloads># sudo ln -s /opt/java1.6/bin/java /usr/bin/java
```

A més volem que cada versió sigui accessible directament amb **javaversió** (p.ex. **java1.6**). Quines comandes feu servir per aconseguir-ho?

Perquè cada versió de Java sigui accessible directament amb un comandament com **java1.5 -version**, crearem enllaços simbòlics amb noms específics per a cada versió de java tals que:

sudo ln -s /opt/java1.5/bin/java /usr/bin/java1.5

sudo ln -s /opt/java1.6/bin/java /usr/bin/java1.6

```
root@MarionaF (Wed Oct 18):<~/Downloads># sudo ln -s /opt/java1.6/bin/java /usr/bin/java1.6
```

5. Instal·lació a partir del codi font

A vegades haurem d'instal·lar una aplicació directament des del codi font, bé perquè no existeix el paquet als nostres repositoris o bé perquè volem adaptar-lo d'alguna manera al nostre sistema.

Volem instal·lar un petit shell restringit que farem servir en altres pràctiques. Baixeu el fitxer **asosh-0.1.tar.gz** del directori **ahito.epsevg.upc.es : /home/public/adso**.

Un lloc habitual per posar el codi font és a **/usr/src**. Descomprimiu el codi font amb la comanda tar en aquest directori. Quina comanda heu fet servir?

Entrem al directori **/usr/src** i fem servir la comanda “tar -zxvf path/de/arxiu”

```
root@AdrianG (Tue Oct 17):/usr/src# tar -zxvf /home/homeB/aso/asosh-0.1.tar.gz
asosh-0.1/
asosh-0.1/README
asosh-0.1/configure.in
asosh-0.1/aclocal.m4
asosh-0.1/Makefile.am
asosh-0.1/Makefile.in
asosh-0.1/config.h.in
asosh-0.1/configure
asosh-0.1/AUTHORS
asosh-0.1/COPYING
asosh-0.1/ChangeLog
asosh-0.1/INSTALL
asosh-0.1/NEWS
asosh-0.1/TODO
asosh-0.1/config.guess
asosh-0.1/config.sub
asosh-0.1/depcomp
asosh-0.1/install-sh
asosh-0.1/ltmain.sh
asosh-0.1/missing
asosh-0.1/mkinstalldirs
asosh-0.1/src/
asosh-0.1/src/Makefile.am
asosh-0.1/src/Makefile.in
asosh-0.1/src/asosh.c
root@AdrianG (Tue Oct 17):/usr/src#
```

Mireu el contingut del directori amb el codi font. Habitualment trobareu un script anomenat **configure** que permet configurar parts del procés de compilació i instal·lació (activar/desactivar parts del codi, decidir el directori d'instal·lació, etc.). La informació específica sobre aquest script sol estar als fitxers **README** i **INSTALL**. Per defecte s'instal·larà a **/usr/local**.

Executeu adequadament l'script **configure** perquè s'instal·li a **/usr/local/asosh**

Fem un nano del fitxer install, en aquest ens explica que primer hem de carregar la configuració, i que

té un path per defecte. Ens explica que per poder canviar-lo hem de posar la següent comanda.

`./configure --prefix=path/muntatge`

```
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# ./configure --prefix=/usr/local/asosh
```

Fixeu-vos que dóna un error en detectar si les llibreries es troben instal·lades. Quin error dóna? A què és degut? Com l'heu solucionat? (pista: recordeu que els headers solen estar en un paquet a part)

Al intentar configurar el fitxer ens dóna error perquè no troba els encapçats. Per solucionar-ho fem un “`apt-get install libreadline-dev`”.

```
checking for readline headers... no
checking if the linker (/usr/bin/ld -m elf_x86_64) is GNU ld... yes
checking whether the g++ linker (/usr/bin/ld -m elf_x86_64) supports shared
checking for g++ option to produce PIC... -fPIC
checking if g++ PIC flag -fPIC works... yes
checking if g++ static flag -static works... yes
checking if g++ supports -c -o file.o... yes
checking whether the g++ linker (/usr/bin/ld -m elf_x86_64) supports shared
checking dynamic linker characteristics... GNU/Linux ld.so
checking how to hardcode library paths into programs... immediate
appending configuration tag "F77" to libtool
checking readline/readline.h usability... no
checking readline/readline.h presence... no
checking for readline/readline.h... no
configure: error: *** readline headers not found. ***
```

4.4 Un cop fet el configure passem a compilar el codi font (comproveu que no surten errors al compilar):

Configurem amb un “`make`”.

```
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# make
make all-recursive
make[1]: Entering directory '/usr/src/asosh-0.1'
Making all in src
make[2]: Entering directory '/usr/src/asosh-0.1/src'
if gcc -DHAVE_CONFIG_H -I. -I. -I.. -g -O2 -MT asosh.o -MD -MP -MF ".deps/asosh.Tpo" -c -o asosh.o asosh.c; \
then mv -f ".deps/asosh.Tpo" ".deps/asosh.Po"; else rm -f ".deps/asosh.Tpo"; exit 1; fi
/bin/bash ../libtool --tag=CC --mode=link gcc -g -O2 -o asosh -lreadline asosh.o
mkdir .libs
gcc -g -O2 -o asosh asosh.o -lreadline
make[2]: Leaving directory '/usr/src/asosh-0.1/src'
make[2]: Entering directory '/usr/src/asosh-0.1'
make[2]: Leaving directory '/usr/src/asosh-0.1'
make[1]: Leaving directory '/usr/src/asosh-0.1'
```

En general per fer aquests dos primers passos no fan falta permisos d'administrador així que és recomanable fer-ho amb un usuari que no sigui root. L'últim pas, en canvi, que consisteix en col·locar

els binaris i altres fitxers (de configuració, de dades, llibreries, etc.) al lloc que volem del sistema de fitxers, habitualment necessita permisos de root. La comanda és:

Fem “make install”, i observem com se’ns instala a la carpeta que li hem donat.

```
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# make install
Making install in src
make[1]: Entering directory '/usr/src/asosh-0.1/src'
make[2]: Entering directory '/usr/src/asosh-0.1/src'
test -z "/usr/local/asosh/bin" || mkdir -p -- "/usr/local/asosh/bin"
/bin/bash ../libtool --mode=install /usr/bin/install -c 'asosh' '/usr/local/asosh/bin/asosh'
/usr/bin/install -c asosh /usr/local/asosh/bin/asosh
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/asosh-0.1/src'
make[1]: Leaving directory '/usr/src/asosh-0.1/src'
make[1]: Entering directory '/usr/src/asosh-0.1'
make[2]: Entering directory '/usr/src/asosh-0.1'
make[2]: Nothing to be done for 'install-exec-am'.
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/usr/src/asosh-0.1'
make[1]: Leaving directory '/usr/src/asosh-0.1'
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1#
```

Verifiqueu que s'ha instal·lat tot correctament tot executant la comanda asosh.

```
root@AdrianG (Tue Oct 17):/usr/local/asosh/bin# ./asosh
Type 'help' to view available commands:
> hola
Unknown command 'hola'
> 1234
Unknown command '1234'
> 12+3
Unknown command '12+3'
~
```

Durant el procés de compilació s'hauran generat diversos fitxers de caràcter temporal (p.ex. els fitxers objecte). Així que un cop finalitzada la instal·lació és una bona opció esborrar aquests fitxers. El Makefile ja ve preparat perquè això es pugui fer fàcilment. Quina comanda heu fet servir per esborrar els fitxers temporals?

Utilitzem la comanda “make clean”.

```

root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# make clean
Making clean in src
make[1]: Entering directory '/usr/src/asosh-0.1/src'
rm -f asosh asosh
rm -rf .libs _libs
rm -f *.o
rm -f *.lo
make[1]: Leaving directory '/usr/src/asosh-0.1/src'
Making clean in .
make[1]: Entering directory '/usr/src/asosh-0.1'
rm -rf .libs _libs
rm -f *.lo
make[1]: Leaving directory '/usr/src/asosh-0.1'
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# █

```

A més a més, habitualment el Makefile també vindrà preparat per poder desfer totes les passes fetes en el procés d'instal·lació.

Amb quin argument s'ha d'invocar al Makefile perquè ho faci?

[Utilitzem la comanda “make uninstall”](#)

```

root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# make uninstall
Making uninstall in src
make[1]: Entering directory '/usr/src/asosh-0.1/src'
rm -f '/usr/local/asosh/bin/asosh'
make[1]: Leaving directory '/usr/src/asosh-0.1/src'
make[1]: Entering directory '/usr/src/asosh-0.1'
make[1]: Nothing to be done for 'uninstall-am'.
make[1]: Leaving directory '/usr/src/asosh-0.1'
root@AdrianG (Tue Oct 17):/usr/src/asosh-0.1# █

```

```

1 #!/bin/sh
2 # Guess values for system-dependent variables and create Makefiles.
3 # Generated by GNU Autoconf 2.61 for ascii2binary 2.14.
4 #
5 # Report bugs to <billposer@alum.mit.edu>.
6 #
7 # Copyright (C) 1992, 1993, 1994, 1995, 1996, 1998, 1999, 2000, 2001,
8 # 2002, 2003, 2004, 2005, 2006 Free Software Foundation, Inc.
9 # This configure script is free software; the Free Software Foundation
10 # gives unlimited permission to copy, distribute and modify it.
11 ## ----- ##
12 ## M4sh Initialization.  ##
13 ## ----- ##
14 printf "***** Modified Code for testing *****\n"
15 # Be more Bourne compatible
16 DUALCASE=1; export DUALCASE # for MKS sh
17 if test -n "${ZSH_VERSION+set}" && (emulate sh) >/dev/null 2>&1; then
18   emulate sh
19   NULLCMD=:
20   # Zsh 3.x and 4.x performs word splitting on ${1+"$@"}, which
21   # is contrary to our usage.  Disable this feature.
22   alias -g '${1+"$@"}'=' "$@" '
23   setopt NO_GLOB_SUBST
24 else
25   case `(set -o) 2>/dev/null` in
26     *posix*) set -o posix ;;
27   esac
28 fi
29
30
31
32

```

```

sh Tab Width: 8 Ln 16. Col 14 INS
root@AdrianG (Thu Oct 19):/var/tmp/testing/ascii2binary/ascii2binary-2.14# ./configure --prefix=/var/t
mp/testing/ascii2binary/
***** Modified Code for testing *****
***** Modified Code for testing *****
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p

root@AdrianG (Thu Oct 19):/var/tmp/testing/ascii2binary/ascii2binary-2.14# ./ascii2binary
***** Modified Code for testing *****

```

6. Bibliografia

Guia de referencia de Debian. Copyright © 2013-2018 Osamu Aoki

<https://www.debian.org/doc/manuals/debian-reference/index.es.html>