

# Practica02

## Before you start

- ftp, sftp
- ls
- get
- (No la tinc clara)
- tar -t
- tar –ungzip
- tar -x (--extract)
- tar -xz || tar -xj
- ln
- ln -s
- Serveix per poder executar els fitxers que hi ha dins d'aquests directoris sense escriure la ruta completa.

## Installation of binary packages

### Manual installation

1. Com que volem instal·lar el programa “make” haurem d’obtindre'l d’algun lloc. En aquest cas d’un servidor local de la universitat.

NOTA: *En el moment de fer aquesta documentació el servidor sftp no funcionava.*

NOTA: wget no ve instalat -> “sudo apt-get update” + “sudo apt-get install wget”

“**wget https://asoserver.pc.ac.upc.edu/aso\_public/make\_4.4.1-2\_amd64.deb**”

2. Quan ja el tenim en la nostra màquina, haurem d’instal·lar-lo amb “dpkg”.

“**dpkg --install make\_4.4.1-2\_amd64.deb**”

Action	Options	Arguments
Install	-i or –install	package-file
Uninstall	-r or –remove	package
Purge	-p or –purge	package
List installed packages	-L or –listfiles	no argument
List files in a package	-c or –contents	archive

“Uninstall” → Treure tot menys els fitxers de configuració (Per si tornem a instal·lar no haver de fer-ho tot una altra vegada).

“Purge” → Treure-ho tot de tot. Fitxers de configuració inclosos.

### 3. NO PUC FER PQ NO SE CONTRASENYA DE SFTP

4. Si volem instal·lar lynx amb dpkg sense utilitzar -common, haurem d'utilitzar la següent comanda:

“`wget https://asoserver.pc.ac.upc.edu/aso_public/lynx`”

“`dpkg --install lynx_2.9.2-1_amd64.deb`”

Això fa que la instalació falli, ja que lynx té una dependència (lynx-common package).

La comanda dpkg que faria una instalació bona seria:

“`dpkg --install lynx-common_2.9.2-1_all.deb`”

“`dpkg --install lynx_2.9.2-1_amd64.deb`”

“`lynx`”

## Installation with a package managers

5. Si volem instal·lar a través d'un packet manager (“apt” en cas Debian) haurem de fer unes configuracions prèvies.

Haurem d'indicar de quins servidors volem que obtingui els packets.

“`sudo vim /etc/apt/sources.list`”

```
aso@aso-client:~$ cat /etc/apt/sources.list
deb http://ftp.es.debian.org/debian stable main non-free contrib
```

6. Per forçar que el programa “apt” busqui a aquest nou servidor, haurem de fer:

“`sudo apt update`”

Si volguessim actualitzar a l'última versió tots els nostres paquets hauríem de fer

“`sudo apt upgrade`”

7. Com que volem instal·lar “X-windows system”, per fer-ho còmode pel que fa a dependències, ho farem amb “apt”.

“`sudo apt install x-window-system`”

*NOTA: Si tens alguna cosa LOCK in cache ->*

“`ps aux|grep num_lock`”

“`jobs`”

“`fg`” Ctrl-C

“`kill -CONT num_lock`” if fg no work

8. Adicionalment voldrem instal·lar un window manager. Per buscar quin podem descarregar, podem fer ús de la base de dades de “apt”.

**“apt list”**

9. Com que un escritori pot dependre de diversos paquets, Debian implementa els “bundle”. Aquest és un packet que inclou la resta.

**“apt list | grep “^task””**

10. Si volem buscar un packet que compleixi un nom específic podem usar el fet que “grep” permet regex.

**“apt list | grep “task-\*-desktop””**

11. També podem fer servir “apt search” que permet la cerca (amb regex també) de la llista.

**“apt search task-\*-desktop”**

12. Finalment decidim instal·lar el KDE.

*NOTA: Per començar a veure l'entorn gràfic, s'ha de reiniciar la màquina.*

**“sudo apt install task-kde-desktop”**

13. Si volem instal·lar més aplicacions ho haurem de fer amb aquest gestor de paquet.

*NOTA: firefox no fa falta perquè ja ve instal·lat.*

**“sudo apt install gcc libc6-dev”**

14. Quan hem acabat d'instal·lar, és bona pràctica netejar la cache que conté els repositoris temporals requerits per fer instal·lacions prèvies.

**“sudo apt clean”**

La diferència amb “autoclean” és que aquesta només elimina aquells repositoris que no s'han fet servir o que són usless.

15. Ara volem instal·lar una versió de Java (precompilada) que està en el directori d'ASO (I posteriorment voldrem una que estigui en els repositoris Debian).

**“wget [https://asoserver.pc.ac.upc.edu/aso\\_public/java/openjdk-25\\_linux-x64\\_bin.tar.gz](https://asoserver.pc.ac.upc.edu/aso_public/java/openjdk-25_linux-x64_bin.tar.gz)”**

16. Donat que és una aplicació propietària, volem guardar-la a “/opt” d'aquesta forma.

A més a més, volem que el directori estigui anomenat d'una forma especial.

**“mkdir /opt/java25”**

**“cd /opt/java25”**

**“sudo tar -xzf \$HOME/openjdk-25\_linux-x64\_bin.tar.gz”**

**“mv jdk-25 /opt/java1.25”**

```
aso@aso-client:~$ /opt/java1.25/bin/java -version
openjdk version "25" 2025-09-16
OpenJDK Runtime Environment (build 25+36-3489)
OpenJDK 64-Bit Server VM (build 25+36-3489, mixed mode, sharing)
aso@aso-client:~$ |
```

17. Per acabar amb aquesta part, voldrem instal·lar la versió 21JRE que hi hagi al repositori i poder fer servir les dues indistintament.

**“sudo apt install openjdk-21-jre”**

*NOTA: Post instal·lació, per defecte s'executarà el 21-jre pq l'executable està dins un directori del PATH.*

18. Si volem que l'usuari pugui executar dependent de la comanda → Crear alias.

**“vim \$HOME/.bashrc”**

**“source \$HOME/.bashrc” # Per actualitzar sense fer logout**

```
alias java1.25='/opt/java1.25/bin/java'  
alias java1.21='/usr/bin/java'
```

```
aso@aso-client:~$ java1.25 -version  
openjdk version "25" 2025-09-16  
OpenJDK Runtime Environment (build 25+36-3489)  
OpenJDK 64-Bit Server VM (build 25+36-3489, mixed mode, sharing)  
aso@aso-client:~$ java1.21 -version  
openjdk version "21.0.8" 2025-07-15  
OpenJDK Runtime Environment (build 21.0.8+9-Debian-1)  
OpenJDK 64-Bit Server VM (build 21.0.8+9-Debian-1, mixed mode, sharing)  
aso@aso-client:~$
```

## Installation from source code

**IMPO:** No existeix "asosh". Com alternativa instal·larem "buggy-1.0"

1. Volem instal·lar "buggy" que és un programa que necessitarem el seu source code.

Com que és un programa de l'usuari:

- Guardar en "/usr/src" el codi font
- Instal·lar en "/usr/local" el programa

"**wget <https://m.seehuhn.de/programs/moon-buggy-1.0.tar.gz>**"

2. Quan ja hem obtingut el source code → El descomprimim.

"**cd /usr/src**"

"**sudo tar -zxfv ~/moon-buggy-1.0.tar.gz**"

3. Si en algun cas volguéssim instal·lar el programa en un altre directori que no sigui el per defecte, llegint el "INSTALLATION", veiem que ho podem fer de la següent manera.

"**sudo mkdir /usr/local/buggy-1.0**"

"**./configure --prefix=/usr/local/buggy-1.0**"

Quan el script ha creat tots els fitxers necessaris per a la instal·lació, haurem de compilar i instal·lar.

**Donat que és source code, pot ser que fallin "#includes" perquè no tenim els ".h" corresponents.** En aquest cas passarà i ho solucionem d'aquesta forma.

*NOTA: Davant d'aquests tipus d'error, es busca en internet i hi haurà algú que et digui quina llibreria instal·lar.*

"**make**"

"**sudo apt install libncurses-dev**"

4. Ara que ja ha compilat i no s'ha queixat, podem instal·lar-lo.

"**sudo make install**"

```
aso@aso-client:/usr/src/moon-buggy-1.0$ ls /usr/local/buggy-1.0/
bin  com  info  man
```

5. Per ser més endreçats, netegem el directori amb un paràmetre que té el mateix Makefile.

"**make clean**"

Si en algun cas volguéssim desinstal·lar, és millor fer-ho amb el paràmetre del Makefile:

"**sudo make uninstall**"