

Sujet 102: Installation de Linux et gestion de paquetages

Présenté par : Mohamed Belhassen
2019–2020

Installation de Linux et gestion de paquets

- ▶ 102.1 Concevoir un **schéma de partitionnement** du disque dur (**Weight 2**)
- ▶ 102.2 Installation d'un **gestionnaire de démarrage** (**Weight 2**)
- ▶ 102.3 Gérer les **bibliothèques partagées** (**Weight 1**)
- ▶ 102.4 Utilisation de **gestionnaire de paquetage Debian** (**Weight 3**)
- ▶ 102.5 Utilisation de **gestionnaire de paquetage RPM** et **YUM** (**Weight 3**)
- ▶ 102.6 Linux en tant que **système virtuel hébergé** (**Weight 1**)

Concevoir un schéma de partitionnement du disque dur

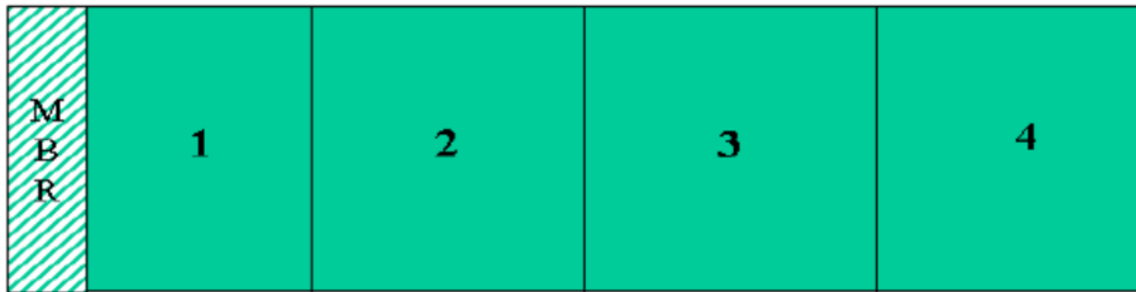
Concevoir un schéma de partitionnement du disque dur

- ▶ **Description :** Les candidats doivent être capables de concevoir un schéma de partitionnement du disque dur pour un système Linux.
- ▶ **Termes, fichiers et utilitaires utilisés :**
 - / : le système de fichiers racine (root)
 - /var
 - /home
 - swap
 - points de montage
 - partitions

Disque

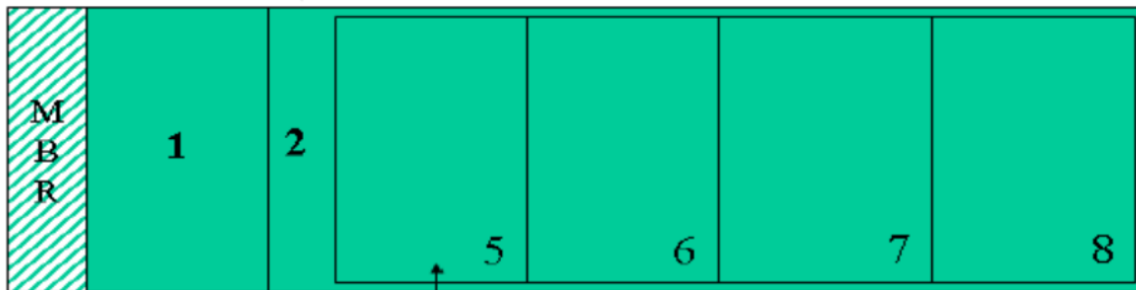
- ▶ /dev/hda
 - périphérique IDE **primaire maître** (Disque dure)
- ▶ ☐ /dev/hdb
 - périphérique IDE **primaire esclave** IDE
- ▶ ☐ /dev/hdc
 - périphérique IDE **secondaire maître** (CD-ROM)
- ▶ ☐ /dev/hdd
 - périphérique IDE **secondaire esclave**
- ▶ ☐ /dev/sda
 - premier disque **SCSI /sata**
- ▶ ☐ /dev/sdb
 - Second disque **SCSI /sata**

Partitions



primary

extended



logical

Seules les partitions
primaires peuvent
être bootables

Installation de Linux

► Espace disque / mémoire limité :

- swap 100 MB,
- / : 1 GB ,
- pas de GUI

► Laptop récent :

- Swap = taille RAM
- / : 5 GB
- /home : le reste

► Serveur

- Pas de X Window
- / : 500 MB (minimum).
- /usr : partagé via NFS.
- /var : 2 GB.
- /tmp 500 MB.
- /home 90 GB. partagé via NFS

Taille du swap (RedHat)

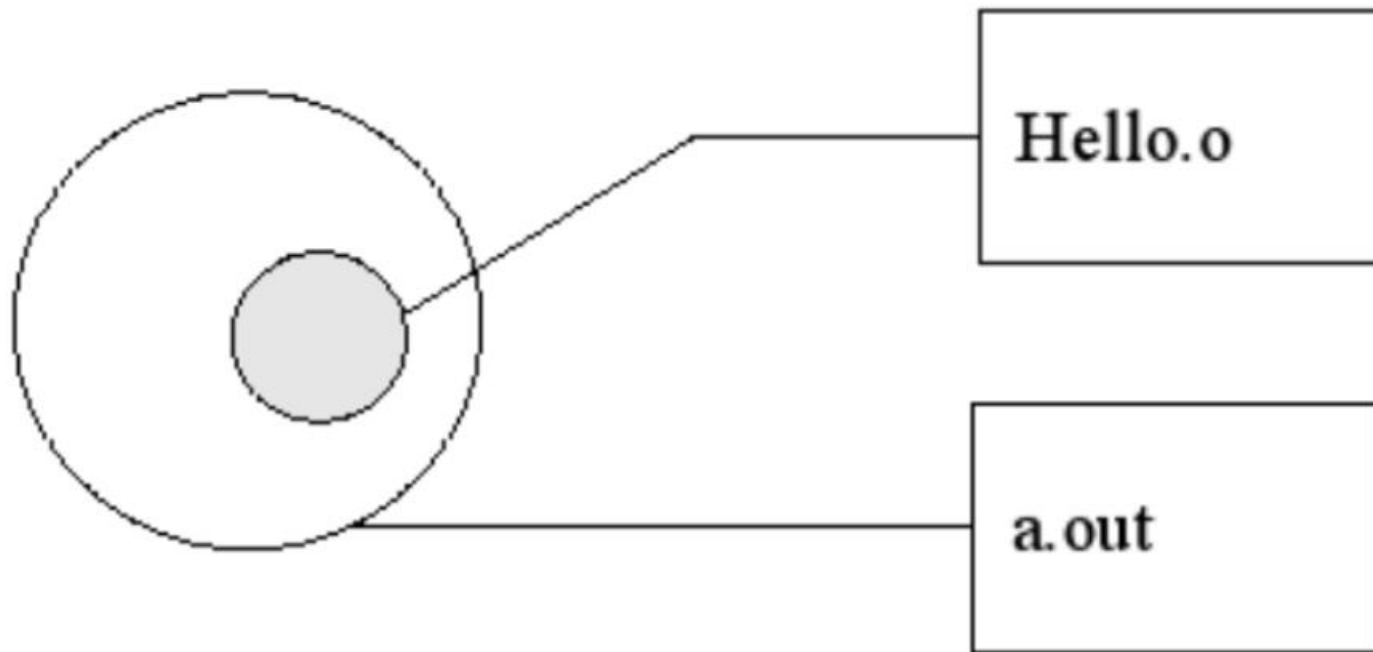
Amount of RAM in the system	Recommended swap space	Recommended swap space if allowing for hibernation
≤ 2GB	2 times the amount of RAM	3 times the amount of RAM
> 2GB - 8GB	Equal to the amount of RAM	2 times the amount of RAM
> 8GB - 64GB	0.5 times the amount of RAM	1.5 times the amount of RAM
> 64GB	4GB of swap space	No extra space needed

Gérer les librairies partagées

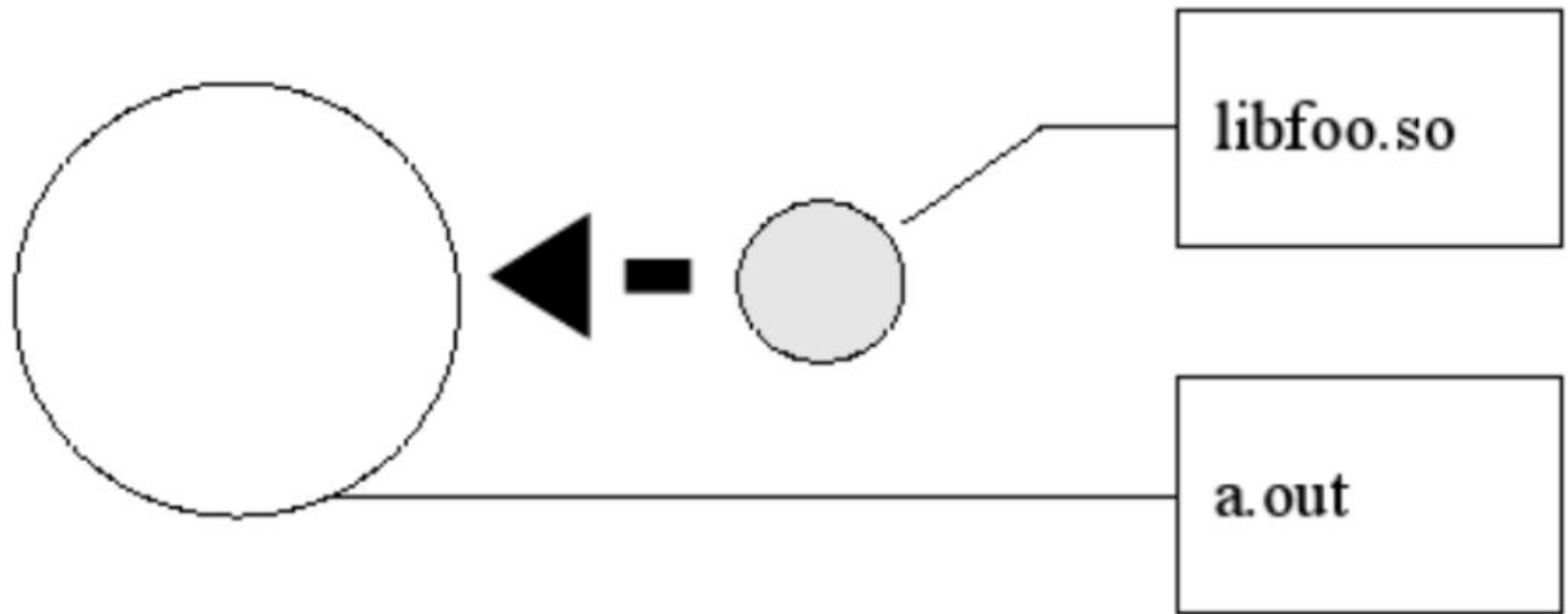
Gérer les bibliothèques partagées

- ▶ **Description** : Les candidats doivent être capables de déterminer les bibliothèques dont dépend les programmes exécutables et de les installer au cas de besoin
- ▶ **Termes, fichiers et utilitaires utilisés** :
 - ldd
 - ldconfig
 - /etc/ld.so.conf
 - LD_LIBRARY_PATH

Bibliothèque Statique



Bibliothèques partagées ou dynamiques



Édition de lien

- ▶ **ld.so** permet **l'édition des liens** avec une bibliothèque dynamique au moment de l'exécution du programme par le système
- ▶ **Tout** programme **est lié** à la bibliothèque **ld.so**.
- ▶ Le chargeur de liens ld.so **recherche** les bibliothèques dans **plusieurs endroits** dans cet ordre :
 - **LD_LIBRARY_PATH**. Les chemins sont séparés par : ,
 - le contenu du fichier **/etc/ld.so.cache** (format binaire) des bibliothèques
 - les répertoires **/lib** et **/usr/lib**

Détermination des dépendances des programmes

- ▶ **ldd** permet de déterminer quelles sont les bibliothèques liées à un programme
- ▶ **Exemple**

```
belhassen@belhassen-VirtualBox: /etc/ld.so.conf.d
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
belhassen@belhassen-VirtualBox:/etc/ld.so.conf.d$ ldd /bin/cp
linux-vdso.so.1 (0x00007ffc9b33e000)
libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007f3c47738000)
libacl.so.1 => /lib/x86_64-linux-gnu/libacl.so.1 (0x00007f3c47530000)
libattr.so.1 => /lib/x86_64-linux-gnu/libattr.so.1 (0x00007f3c4732b000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f3c46f3a000)
libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007f3c46cc8000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f3c46ac4000)
/lib64/ld-linux-x86-64.so.2 (0x00007f3c47b83000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f3c468a5000)
belhassen@belhassen-VirtualBox:/etc/ld.so.conf.d$
```

ldd /bin/cp

ldconfig

- ▶ **ldconfig** permet de construire le cache ld.so.cache est construit depuis le contenu du fichier /etc/ld.so.conf .
- ▶ **ldconfig -p** : permet d'afficher la cache
- ▶ **ldconfig -v** : Mode bavard : indique ce que ldconfig effectue

Application: Bibliothèque statique vs dynamique

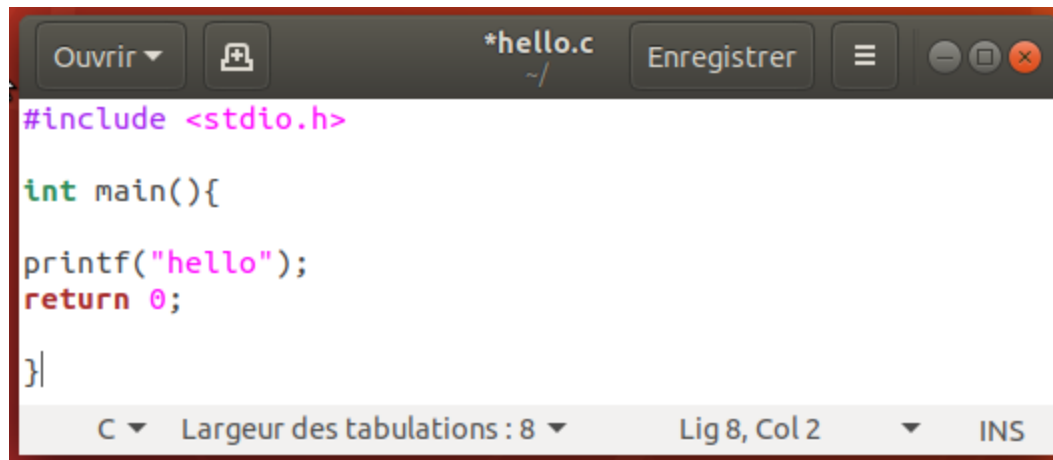
► Les paquets nécessaires:

- Installer en priorité le paquet build-essential.
- Les langages de programmation (non-interprétés) les plus populaires sous Linux sont le C et le C++.
- Si vous récupérez le code source d'un programme sur internet, il sera très probablement écrit dans l'un de ces langages
- Exécutez la commande

```
sudo apt-get install build-essential
```


Application: Bibliothèque statique vs dynamique

- ▶ Dans un terminal, exécutez la commande:
 - `gedit hello.c`
- ▶ Dans l'éditeur de texte, saisir le code suivant:



The screenshot shows the gedit text editor window. The title bar indicates the file is named `*hello.c` and is located in the `~/` directory. The menu bar includes 'Ouvrir', 'Enregistrer', and a hamburger menu icon. The main text area contains the following C code:

```
#include <stdio.h>

int main(){
    printf("hello");
    return 0;
}
```

The status bar at the bottom shows the file encoding is 'C', the tab width is 'Largeur des tabulations : 8', the cursor is at 'Lig 8, Col 2', and the input mode is 'INS'.

- ▶ Enregistrez le fichier puis fermer le programme

Application: Bibliothèque statique vs dynamique

- ▶ Dans le terminal, tapez les commandes suivantes:
 - gcc hello.c -o hello_dynamique
 - gcc hello.c -static -o hello_statique
 - ls -l

Le fichier reposant sur une bibliothèque dynamiquement chargée est beaucoup plus petite

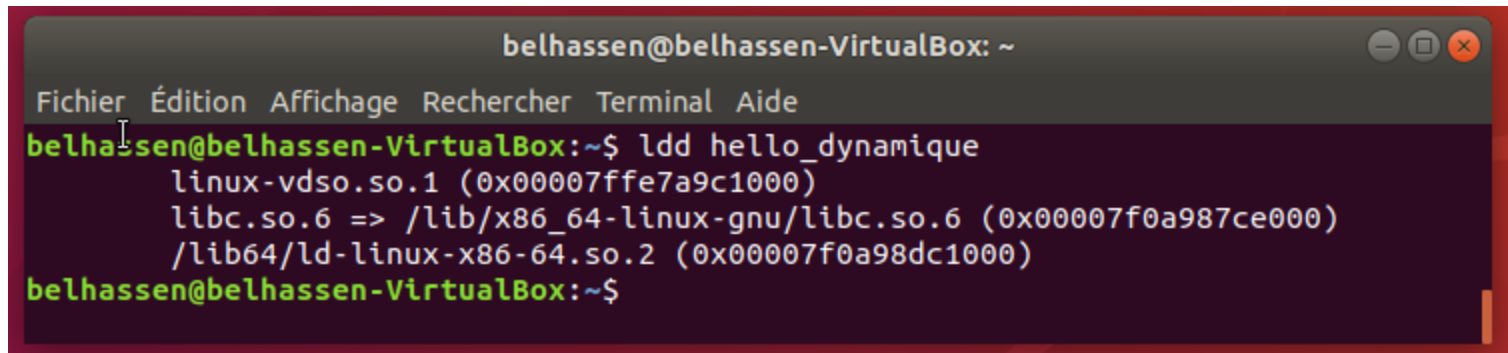
```
belhassen@belhassen-VirtualBox: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
belhassen@belhassen-VirtualBox:~$ gcc hello.c -static -o hello_statique  
belhassen@belhassen-VirtualBox:~$ gcc hello.c -o hello_dynamique  
belhassen@belhassen-VirtualBox:~$ ls -l  
total 888  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Bureau  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Documents  
-rw-r--r-- 1 belhassen belhassen 8980 6 16:55 examples.desktop  
-rw-r--r-- 1 belhassen belhassen 60 25 20:17 hello.c  
-rw-r-xr-x 1 belhassen belhassen 8304 25 20:18 hello_dynamique  
-rw-r-xr-x 1 belhassen belhassen 844736 25 20:18 hello_statique  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Images  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Modèles  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Musique  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Public  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Téléchargements  
drwxr-xr-x 2 belhassen belhassen 4096 7 17:58 Vidéos  
belhassen@belhassen-VirtualBox:~$  
belhassen@belhassen-VirtualBox:~$
```

Application: Bibliothèque statique vs dynamique

- ▶ Pour tester les deux programmes, il suffit d'exécuter les commandes:
 - ./hello_dynamique
 - ./hello_statique
- ▶ **Remarques:**
 - Les deux programmes donnent **le même résultat**
 - Le programme compilé en utilisant une **bibliothèque dynamique** est **plus petit** que celui compilé statiquement.

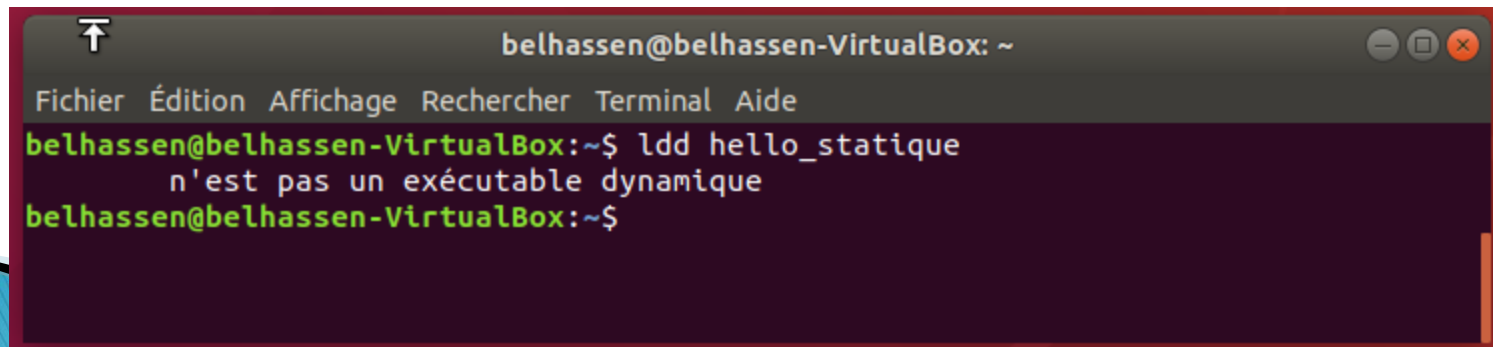
Application: Bibliothèque statique vs dynamique

- ▶ Exécutez la commande suivante:
 - ldd hello_dynamique



```
belhassen@belhassen-VirtualBox: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
belhassen@belhassen-VirtualBox:~$ ldd hello_dynamique  
linux-vdso.so.1 (0x00007ffe7a9c1000)  
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f0a987ce000)  
/lib64/ld-linux-x86-64.so.2 (0x00007f0a98dc1000)  
belhassen@belhassen-VirtualBox:~$
```

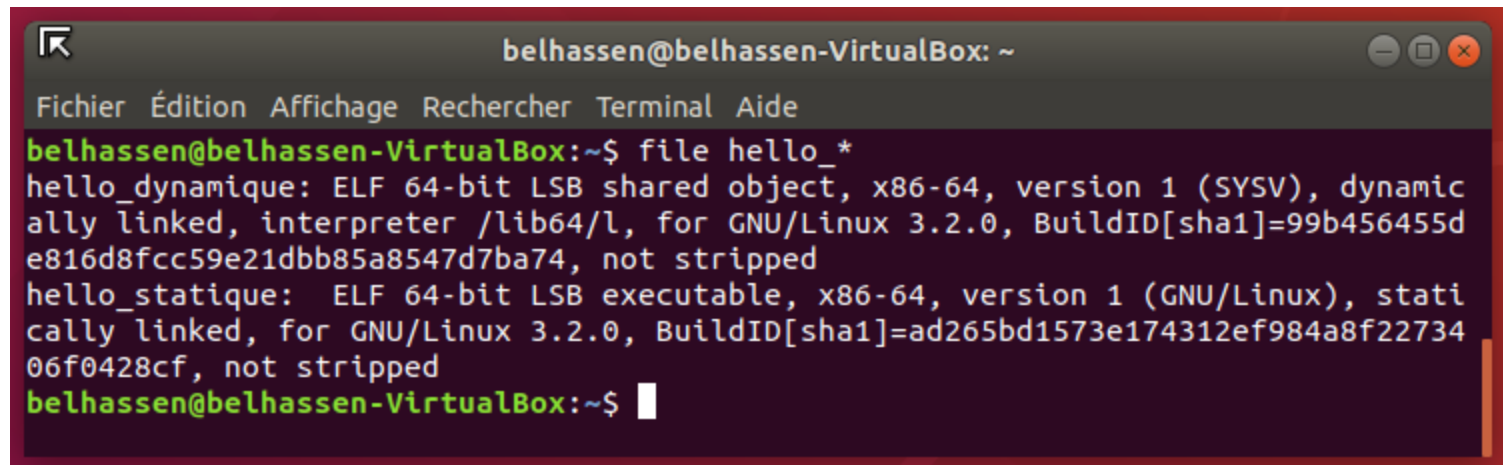
- ldd hello_statique



```
belhassen@belhassen-VirtualBox: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
belhassen@belhassen-VirtualBox:~$ ldd hello_statique  
n'est pas un exécutable dynamique  
belhassen@belhassen-VirtualBox:~$
```

Application: Bibliothèque statique vs dynamique

- Pour afficher le types des deux fichiers ainsi créés, tapez la commande:
 - `file hello_*`

A terminal window titled 'belhassen@belhassen-VirtualBox: ~' with a menu bar containing 'Fichier', 'Édition', 'Affichage', 'Rechercher', 'Terminal', and 'Aide'. The terminal shows the command 'file hello_*' being executed. The output identifies 'hello_dynamique' as an ELF 64-bit LSB shared object and 'hello_statique' as an ELF 64-bit LSB executable, both for x86-64 architecture and GNU/Linux 3.2.0.

```
belhassen@belhassen-VirtualBox: ~$ file hello_*
hello_dynamique: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamic
ally linked, interpreter /lib64/l, for GNU/Linux 3.2.0, BuildID[sha1]=99b456455d
e816d8fcc59e21dbb85a8547d7ba74, not stripped
hello_statique:  ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), stati
cally linked, for GNU/Linux 3.2.0, BuildID[sha1]=ad265bd1573e174312ef984a8f22734
06f0428cf, not stripped
belhassen@belhassen-VirtualBox:~$
```

Utilisation de gestionnaire de paquetage Debian

Utilisation de gestionnaire de paquetage Debian

- ▶ **Description:** Les candidats doivent être capables gérer les paquetages en utilisant les outils de gestion de paquetage sous Debian.
- ▶ **Termes, fichiers et utilitaires utilisés :**
 - /etc/apt/sources.list
 - dpkg
 - dpkg-reconfigure
 - apt-get
 - apt-cache
 - aptitude

dpkg : résumé

Action	Description
-l	Prints a list of the packages installed on the system, or matching a pattern if any is given. The first three characters on each line show the state, selection state, and flags of the package
-s	Shows the status and information about particular installed package(s)
-I	Show information about a package in a .deb file
-L	List the files included in a package
-S	Show the package which includes the file specified
-i	Install (or upgrade) and configure a package from a .deb file
--unpack	Unpack (only) a package in a .deb file
--configure	Configure an unpacked package. With -a (or --pending) configures all packages requiring configuration
-r	Remove a package (but leave its configuration files)
-P	Purge – remove a package along with its configuration files
--get-selections	Get a list of package selections from a system (to stdout)
--set-selections	Set the list of package selections for a system (from stdin)

Cloner l'état d'installation de packages d'une machine à une autre

- ▶ Quel est l'état actuel de votre machine ?
 - Votre machine ubuntu/debian est une collection de nombreux paquets.
 - Chaque fois que votre machine tombe en panne, en plus de perdre les données précieuses, vous pouvez également perdre les informations sur les différents paquets que vous avez installés sur votre machine.
- ▶ `dpkg --get-selections` donne la liste de tous les noms de paquets et leur état actuel comme installé, désinstallé, etc.
- ▶ Pour sauvegarder la liste de packages installés dans un fichier texte:
 - `dpkg --get-selections > packages_list`

Cloner l'état d'installation de packages d'une machine à une autre

- ▶ Le clonage d'une machine signifie l'installation des mêmes paquets (parfois même des paramètres) que ceux trouvés dans la machine source.
- ▶ dpkg vous permet d'obtenir l'état actuel de votre machine – l'ensemble des paquets installés dans la machine et leur état (qu'ils soient installés ou désinstallés)

Cloner l'état d'installation de packages d'une machine à une autre

- ▶ Supposons que vous disposiez de la liste `packages_list` contenant l'état de la machine source.
- ▶ Maintenant, allez à la machine de destination et exécutez la commande suivante
 - `$ sudo dpkg --set-selections < ./packages_list`
 - Cette commande n'installera aucun paquet, mais marquera seulement un état correspondant à chaque paquet.
- ▶ Pour l'installation réelle, exécutez la commande suivante
 - `$ sudo apt-get -u dselect-upgrade`
 - Cela installera tous les paquets nécessaires. Certains paquets peuvent être supprimés. En fin de compte, les machines source et de destination auront le même état

apt-get : Advanced Packaging Tool

- ▶ Contrôler les dépendances des packages pour autoriser ou non leur installation
- ▶ **/etc/apt/sources.list**

```
#deb cdrom:[Ubuntu 18.04.1 LTS _Bionic Beaver_ – Release  
amd64 (20180725)]/ bionic main restricted  
deb http://tn.archive.ubuntu.com/ubuntu/ bionic main restricted  
# deb-src http://tn.archive.ubuntu.com/ubuntu/ bionic main  
restricted  
## Major bug fix updates produced after the final release of the  
## distribution.  
deb http://tn.archive.ubuntu.com/ubuntu/ bionic-updates main  
restricted  
# deb-src http://tn.archive.ubuntu.com/ubuntu/ bionic-updates  
main restricted
```

apt-get : Résumé

Action	Description
update	Update the list of packages from the sources in /etc/apt/sources.list
install <i>package</i>	Install the package(s) specified, along with any dependencies
upgrade	Upgrade any packages which have newer versions available
dist-upgrade	Upgrade entire distribution to the latest release (best to read the release notes first!)
remove	Remove the package(s) specified

Utilisation de gestionnaire de paquetage RPM et YUM

Utilisation de gestionnaire de paquetage RPM et YUM

- ▶ **Description:** Les candidats doivent être capables de gérer les paquetages avec les outils RPM et YUM
- ▶ **Termes, fichiers et utilitaires utilisés:**
 - rpm
 - rpm2cpio
 - /etc/yum.conf
 - /etc/yum.repos.d/
 - yum
 - yumdownloader

rpm : résumé

Short	Long	Description
-i	--install	Installs the package
-U	--update	Updates or installs a package
-F	--freshen	Updates only installed package
-V	--verify	file size, MD5, permissions, type ...
-q	--query	Queries installed/uninstalled packages, and files
-e	--erase	Uninstall package

Short	Description
a	applies to all installed packages
c	together with q lists c onfiguration files
d	together with q lists d ocumentation files
f	together with q queries which package installed a given file
h	adds hashes while processing
i	together with q lists i nformation about a package
l	together with q lists all files and directories in a package
p	together with q specifies that the query is performed on the package file
v	verbose

Application

- ▶ Appliquer les différentes utilisations de RPM illustrées dans le tutoriel suivant:
 - <https://www.computernetworkingnotes.com/linux-tutorials/rpm-command-in-linux-explained-with-examples.html>
 - → https://bit.ly/RPM_TUTO

YUM

- ▶ Récupère les packages au sein de dépôts et gère les dépendances.
- ▶ Fichier de configuration `/etc/yum.conf`
- ▶ Les dépôts sont placés : `/etc/yum.conf` ou `/etc/yum.repos.d`
- ▶ **Exemple:** Fichier `/etc/yum.repos.d/dag.repo`
[dag]
name=Dag RPM Repository for Fedora Core \$releasever
baseurl=[http://apt.sw.be/fedora/\\$releasever/en/\\$basearch/dag](http://apt.sw.be/fedora/$releasever/en/$basearch/dag)
enabled=10
gpgcheck=1
gpgkey=<http://dag.wieers.com/packages/RPM-GPG-KEY.dag.txt>

yum résumé

Commande	Description
# yum -y install [package]	télécharge et installe un paquet
# yum localinstall [package.rpm]	cela installera un RPM, et tentera de résoudre toutes les dépendance pour vous en utilisant votre dépôt
# yum -y update	met à jour tous les paquets rpm installés sur le système
# yum update [package]	met à jour un paquet rpm
# yum remove [package]	enlève un paquet rpm
# yum list	liste tous les paquets installés sur le système
# yum search [package]	trouve le paquet sur le dépôt rpm
# yum clean [package]	nettoie le cache rpm et supprimant les paquets téléchargés
# yum clean headers	supprime toutes les entêtes des fichiers que le système utilise pour résoudre les dépendances
# yum clean all	supprime le cache des paquets et les entêtes des fichiers