

# Préparation LPI

## Exam 101

### 102.3. Librairies partagées

## Sommaire

- Librairie statiques et dynamiques
- Emplacement des librairies
- ldconfig
- /etc/ld.so.conf
- ldd
- LD\_LIBRARY\_PATH

- Poids : 1
- Identifier les libraires
- Déclarer les chemins d'installation des librairies
- Chargement des librairies partagées

- Libraires statiques et librairies partagées
- Librairies statiques
  - Liées directement dans le code de l'exécutable
  - Plus gros exécutables : empreinte mémoire plus importantes
  - Moins de souplesse qu'avec les librairies partagées : si modification d'une fonction dans un librairie, obligation de recompiler la totalité de l'exécutable
  - Intéressant pour utiliser des applications anciennes qui poseraient problème avec des librairies partagées plus récentes
  - Sécurisation des exécutables : suppression du risque de compromission d'une librairie
  - Extension : `.a`

- Libraires partagée ou dynamique
  - L'exécutable contient des références à la librairie dynamique
  - Chargée en mémoire uniquement à l'utilisation
  - Peut être partagée par plusieurs applications
  - Gérer l'accessibilité des librairies partagées
  - Possibilité d'héberger plusieurs version d'une librairie
  - Extensions :
    - `.so`
    - `.so.version`
  - Librairie indispensable au fonctionnement de Linux
    - `glibc : /lib/libc-2.10.1.so`

- R epertoires habituels d'emplacement des librairies
  - `/lib`
  - `/usr/lib`
  - `/usr/local/lib`
- Chemins d efinis dans le fichier `/etc/ld.so.conf`
  - Certaines distributions utilisent plusieurs fichiers de configurations
  - En d ebut de fichier `/etc/ld.so.conf` :  
`include /etc/ld.so.conf.d/*.conf`
  - Des fichiers avec extension `.conf`
  - Apr es ajout d'un chemin, utilisation de la commande `ldconfig` pour prise en compte par le syst eme

- Chemins définis dans la variable `LD_LIBRARY_PATH`
  - Chemins séparés par le caractère ':'
  - Export  
`LD_LIBRARY_PATH=/opt/appli/lib:/opt/appli2/lib`
  - Attention : les chemins déclarés dans `LD_LIBRARY_PATH` sont utilisés en priorité  
Pratique pour tester un nouvelle version de librairie
- Lorsque la librairie est installée dans un répertoire mais l'application la recherche dans un répertoire différent (défini lors de la phase de compilation par exemple) ou lorsque l'application recherche la librairie sous un autre nom
  - Solution : le lien symbolique
  - `ln -s /usr/lib/mozilla/plugins/nppdf.so /home/franck/.mozilla/plugins/nppdf.so`

- Lorsqu'une librairie est inexistante
  - Rechercher (sur internet ou sur les d p ts alternatifs de paquets) une version compil e pour l'architecture cible ou le code source   compiler
  - Rechercher avec le gestionnaire de paquet de haut niveau un paquet qui contiendra la librairie recherch e
    - `yum whatprovides */lib.foo.so.3`

- Lorsque l'application ne fonctionne pas à cause d'une librairie manquante ou d'une version incompatible  
Lorsque qu'une librairie dépend d'une autre librairie inexistante
  - Commande `ldd` : affiche la liste des bibliothèques dynamiques indispensables à l'exécution de la commande
- Commande utilisable par tout utilisateur
- En fonction d'une déclaration complète du chemin
  - Si le chemin absolu est spécifié : `/lib/ld-linux.so.2`  
(`0x0049c000`)
  - Sinon, `ldd` cherche la librairie dans les chemins spécifiés  
`libselinux.so.1 => /lib/libselinux.so.1`  
(`0x008f2000`)
- Option :
  - `-v` : affiche toutes les informations (version en particulier)

```
$ ldd -d /bin/mv
linux-gate.so.1 => (0x00eaa000)
libselinux.so.1 => /lib/libselinux.so.1 (0x00548000)
libacl.so.1 => /lib/libacl.so.1 (0x00a58000)
libattr.so.1 => /lib/libattr.so.1 (0x006e7000)
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0x007d4000)
libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0x00d7f000)
/lib/ld-linux.so.2 (0x00e27000)
```

```
$ ldd -v /bin/mv
linux-gate.so.1 => (0x00a31000)
```

```
....
```

Version information:

/bin/mv:

libattr.so.1 (ATTR\_1.1) => /lib/libattr.so.1

libacl.so.1 (ACL\_1.0) => /lib/libacl.so.1

libc.so.6 (GLIBC\_2.6) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.3) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.4) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.2) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.3.4) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.1) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.1.3) => /lib/tls/i686/cmov/libc.so.6

libc.so.6 (GLIBC\_2.0) => /lib/tls/i686/cmov/libc.so.6

/lib/libselinux.so.1:

ld-linux.so.2 (GLIBC\_2.3) => /lib/ld-linux.so.2

libdl.so.2 (GLIBC\_2.1) => /lib/tls/i686/cmov/libdl.so.2

libdl.so.2 (GLIBC\_2.0) => /lib/tls/i686/cmov/libdl.so.2

```
.....
```

## Quelques conseils LPI...

