x04 - Shared libraries

Curs 2021 - 2022

ASIX M01-ISO UF1-A01-16 Gestió de shared libraries

Shared libraries

Descripció

Conceptes clau:

- ❏ Identify shared libraries.
- ❏ Identify the typical locations of system libraries.
- ❏ Load shared libraries.

Ordres a treballar:

- ❏ ldd
- ❏ ldconfig
- ❏ /etc/ld.so.conf
- ❏ LD_LIBRARY_PATH

Shared libraries

Shared libraries o bé també anomenades shared objects. Un programa en compilar-se pot incorporar codi extern de dues maneres:

Static libraries

A static library is merged with the program at link time. A copy of the library code is embedded into the program and becomes part of it. Thus, the program has no dependencies on the library at run time because the program already contains the libraries code.

Shared (or dynamic) libraries

In the case of shared libraries, the linker simply takes care that the program references libraries correctly. The linker does, however, not copy any library code into the program file. At run time, though, the shared library must be available to satisfy the program's dependencies. This is an economical approach to managing system resources as it helps reduce the size of program files and only one copy of the library is loaded in memory, even when it is used by multiple programs.

Shared Object File Naming Conventions

The name of a shared library, soname, is made up of three elements:

- Library name (normally prefixed by lib)
- so (which stands for "shared object")
- Version number of the library

```
$ ls -la /lib64/libgcalc-2.so.1
```

```
-rwxr-xr-x. 1 root root 190816 Mar  8  2020 /lib64/libgcalc-2.so.1
```

```
$ ls -la /lib64/libcrypt
libcrypto.so.10         libcrypto.so.1.1.1k     libcrypt.so             libcrypt.so.2           libcryptui.so.0.0.0
libcrypto.so.1.0.2o     libcryptsetup.so.12     libcrypt.so.1           libcrypt.so.2.0.0
libcrypto.so.1.1        libcryptsetup.so.12.6.0 libcrypt.so.1.1.0       libcryptui.so.0
```

Directoris principals de llibreries

- /lib
- /lib32
- /lib64
- /usr/lib
- /usr/local/lib

```
$ ls /lib[tab][tab]
lib/    lib32/  lib64/  libx32/

$ ls -l /
…
lrwxrwxrwx   1 root root     7 sep 29  2020 lib -> usr/lib
lrwxrwxrwx   1 root root     9 sep 29  2020 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 sep 29  2020 lib64 -> usr/lib64
lrwxrwxrwx   1 root root    10 sep 29  2020 libx32 -> usr/libx32
…


$ ls /usr/local/lib
python2.7  python3.7
```

Funcionament: ldconfig, configuration, cache

- ld.so o ld-linux.so és el programa que fa de dynamic linker.
- És a dir, l'encarregat de buscar les biblioteques dinàmica que demanen els programes en executar-se.
- Per saber on estan les biblioteques de funcions es mira la configuració del fitxer /etc/lo.so.conf o el directori de configuració /etc/lo.so.conf.d.
- ldconfig és el programa que permet gestionar la configuració. Genera el caché i l'estructura de symbolic links necessaris per trobar les biblioteques de funcions.

```
$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf

$ ls -l /etc/ld.so.conf.d/
total 8
-rw-r--r-- 1 root root  44 mar 21  2016 libc.conf
-rw-r--r-- 1 root root 100 may  1  2019 x86_64-linux-gnu.conf

$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf


$ cat /etc/ld.so.conf.d/libc.conf
# libc default configuration
```

```
/usr/local/lib

$ cat /etc/ld.so.conf.d/x86_64-linux-gnu.conf
# Multiarch support
/usr/local/lib/x86_64-linux-gnu
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
```

```
$ locate ld.so
/etc/ld.so.cache
/etc/ld.so.conf
/etc/ld.so.conf.d
/etc/ld.so.conf.d/libc.conf
/etc/ld.so.conf.d/x86_64-linux-gnu.conf
/etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket
/usr/lib/systemd/system/lvm2-lvmpolld.socket
/usr/lib/systemd/system/systemd-journald.socket
/usr/lib/systemd/system/sockets.target.wants/systemd-journald.socket
/usr/lib/x86_64-linux-gnu/libgphoto2/2.5.22/topfield.so
/usr/share/man/es/man8/ld.so.8.gz
/usr/share/man/man8/ld.so.8.gz
/usr/share/man/man8/systemd-journald.socket.8.gz
/var/lib/systemd/deb-systemd-helper-enabled/lvm2-lvmpolld.socket.dsh-also
/var/lib/systemd/deb-systemd-helper-enabled/sysinit.target.wants/lvm2-lvmpolld.socket
```

```
$ ls -l /etc/ld.so.cache
-rw-r--r--. 1 root root 109285 Oct  5 19:19 /etc/ld.so.cache
```

- The ldconfig command takes care of reading these config files, creating the aforementioned set of symbolic links that help to locate the individual libraries and finally of updating the cache file /etc/ld.so.cache. Thus, ldconfig must be run every time configuration files are added or updated.

```
$ sudo ldconfig -v
/lib/x86_64-linux-gnu:
        libLLVM-7.so.1 -> libLLVM-7.so.1
        libgcc_s.so.1 -> libgcc_s.so.1
        libsamba-errors.so.1 -> libsamba-errors.so.1
        libsepol.so.1 -> libsepol.so.1
        libdevmapper.so.1.02.1 -> libdevmapper.so.1.02.1
        libspice-client-glib-2.0.so.8 -> libspice-client-glib-2.0.so.8.6.0
        libnspr4.so -> libnspr4.so
        libgpm.so.2 -> libgpm.so.2
...
```

```
$ sudo ldconfig -p
1037 bibliotecas se encontraron en la caché `/etc/ld.so.cache'
        libzvbi.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzvbi.so.0
        libzvbi-chains.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzvbi-chains.so.0
        libzstd.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzstd.so.1
        libzmq.so.5 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzmq.so.5
        libzmf-0.0.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzmf-0.0.so.0
        libzeitgeist-2.0.so.0 (libc6,x86-64) =>
/lib/x86_64-linux-gnu/libzeitgeist-2.0.so.0
```

```
$ sudo ldconfig -p  |grep libcrypt
        libcryptsetup.so.12 (libc6,x86-64) => /lib/x86_64-linux-gnu/libcryptsetup.so.12
        libcrypto.so.1.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libcrypto.so.1.1
        libcrypt.so.1 (libc6,x86-64, ABI del SO: Linux 3.2.0) =>
/lib/x86_64-linux-gnu/libcrypt.so.1

$ ls -l /lib/x86_64-linux-gnu/libcrypt.so.1
lrwxrwxrwx 1 root root 16 may  1  2019 /lib/x86_64-linux-gnu/libcrypt.so.1 ->
```

```
libcrypt-2.28.so

$ file /lib/x86_64-linux-gnu/libcrypt-2.28.so
/lib/x86_64-linux-gnu/libcrypt-2.28.so: ELF 64-bit LSB shared object, x86-64, version 1
(SYSV), dynamically linked, BuildID[sha1]=c6966ba263bacba7f67c23643ba373aa51a162e9, for
GNU/Linux 3.2.0, stripped
```

- Observem que el nom symbolic (*soname*) de la llibreria `libcrypt.so.1` apunta en realitat al fitxer `libcrypt-2.28.so`.
- El matrix passa amb el *soname* `libcryptsetup.so.12` que en realitat és un link que apunta al fitxer `libcryptsetup.so.12.4.0`, aquest fitxer és el que conté realment el codi executable.
- Així per exemple es pot actualitzar per una nova versió el fitxer (per exemple 5.0) i simplement cal generar un nou enllaç del *soname* al fitxer.

```
$ ls -l /lib/x86_64-linux-gnu/libcrypt*
-rw-r--r-- 1 root root   43328 may  1  2019 /lib/x86_64-linux-gnu/libcrypt-2.28.so
-rw-r--r-- 1 root root 3031904 abr 20  2020 /lib/x86_64-linux-gnu/libcrypto.so.1.1
lrwxrwxrwx 1 root root      23 ago 26  2019 /lib/x86_64-linux-gnu/libcryptsetup.so.12 ->
libcryptsetup.so.12.4.0
-rw-r--r-- 1 root root  363920 ago 26  2019
/lib/x86_64-linux-gnu/libcryptsetup.so.12.4.0
lrwxrwxrwx 1 root root      16 may  1  2019 /lib/x86_64-linux-gnu/libcrypt.so.1 ->
libcrypt-2.28.so

$ file /lib/x86_64-linux-gnu/libcrypt*
/lib/x86_64-linux-gnu/libcrypt-2.28.so:       ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
BuildID[sha1]=c6966ba263bacba7f67c23643ba373aa51a162e9, for GNU/Linux 3.2.0, stripped
/lib/x86_64-linux-gnu/libcrypto.so.1.1:       ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
BuildID[sha1]=d533a7366b5721837c94b481c21ea2975b17dbad, stripped
/lib/x86_64-linux-gnu/libcryptsetup.so.12:    symbolic link to libcryptsetup.so.12.4.0
/lib/x86_64-linux-gnu/libcryptsetup.so.12.4.0: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
BuildID[sha1]=3a042be37ebe261e97c0df2b9a41c9c588eb7771, stripped
/lib/x86_64-linux-gnu/libcrypt.so.1:          symbolic link to libcrypt-2.28.so
```

```
$ locate libc.so
/usr/lib/x86_64-linux-gnu/libc.so.6

$ file /usr/lib/x86_64-linux-gnu/libc.so.6
/usr/lib/x86_64-linux-gnu/libc.so.6: symbolic link to libc-2.28.so

$ ls -l /usr/lib/x86_64-linux-gnu/libc.so.6
lrwxrwxrwx 1 root root 12 may  1  2019 /usr/lib/x86_64-linux-gnu/libc.so.6 ->
libc-2.28.so

$ file /usr/lib/x86_64-linux-gnu/libc-2.28.so
/usr/lib/x86_64-linux-gnu/libc-2.28.so: ELF 64-bit LSB pie executable, x86-64, version 1
(GNU/Linux), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,
BuildID[sha1]=18b9a9a8c523e5cfe5b5d946d605d09242f09798, for GNU/Linux 3.2.0, stripped
```

LD_LIBRARY_PATH

- The LD_LIBRARY_PATH environment variable can be used to add new paths for shared libraries temporarily. It is made up of a colon-separated (:) set of directories where libraries are looked up.
- Fa la mateixa funció per a les libraries que el PATH per a les ordres.
- Si es vol configurar un directori personal de bibliotecas cal assignar-lo a la variable (un o una llista separada per :) i exportar la variable.

```
$ echo $LD_LIBRARY_PATH

$ export LD_LIBRARY_PATH=/usr/local/mylib

$ echo $LD_LIBRARY_PATH
/usr/local/mylib
```

ldd per llistar dependències

- Amb l'ordre ldd es poden veure quines dependències té un programa executable, quines biblioteques de funcions necessita per poder-se executar.

```
$ ldd /usr/bin/ls
        linux-vdso.so.1 (0x00007ffc8ea99000)
        libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007fa578af4000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fa578933000)
        libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007fa5788bf000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fa5788ba000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fa578d5a000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fa578899000)

$ ldd /usr/bin/bash
        linux-vdso.so.1 (0x00007ffeec5fe000)
        libtinfo.so.6 => /lib/x86_64-linux-gnu/libtinfo.so.6 (0x00007fac5d67d000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fac5d678000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fac5d4b7000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fac5d7ed000)
```

- Volem saber si un programa executable necessita una determinada dependència. Usem grep per examinar-ho. També podem mirar quines dependències tñe una bibiliotca de funcions.

```
$ ldd /usr/bin/gpg | grep crypt
        libgcrypt.so.20 => /lib/x86_64-linux-gnu/libgcrypt.so.20 (0x00007f16bd1b5000)

$ ldd /usr/bin/ssh | grep krb5
        libgssapi_krb5.so.2 => /lib/x86_64-linux-gnu/libgssapi_krb5.so.2 (0x00007faf19d15000)
        libkrb5.so.3 => /lib/x86_64-linux-gnu/libkrb5.so.3 (0x00007faf199dd000)
        libkrb5support.so.0 => /lib/x86_64-linux-gnu/libkrb5support.so.0 (0x00007faf19992000)

$ locate libc.so
/usr/lib/x86_64-linux-gnu/libc.so.6

$ ldd /usr/lib/x86_64-linux-gnu/libc.so.6
        /lib64/ld-linux-x86-64.so.2 (0x00007f9ca7864000)
        linux-vdso.so.1 (0x00007ffd990a2000)
```

```
$ locate libc.so
/usr/lib64/libc.so
/usr/lib64/libc.so.6

$ ls -l /usr/lib64/libc.so*
-rw-r--r--. 1 root root 253 Feb  3  2021 /usr/lib64/libc.so
lrwxrwxrwx. 1 root root  12 Feb  3  2021 /usr/lib64/libc.so.6 -> libc-2.31.so

$ ldd /usr/lib64/libc.so.6
/lib64/ld-linux-x86-64.so.2 (0x00007f58b4a03000)
linux-vdso.so.1 (0x00007ffeaa3f3000)
```

Exercicis d’exemple

1. Which file contains the shared libraries configuration?
2. Which directory contains the shared libraries specific configurations?
3. Which command processes the shared libraries configuration and generates the symbolic link structure?
4. List two library directories.
5. Using the ldconfig command show all the shared libraries information.
6. Which shell variable contains alternate shared libraries directories?
7. Show the shared libraries used by the nmap program.

8. Realitza els exercicis indicats a: https://learning.lpi.org/en/learning-materials/101-500/102/102.3/102.3_01/

9. Realitza els exercicis del Question-Topics 102.3.