

Gestionar las librerías compartidas

1. Fundamentos

Una librería compartida es un archivo particular que contiene una lista de funciones, o API, accesible a cualquier programa que lo necesite sin tener que volver a escribirlas. Al contrario de lo que sucede con las librerías estáticas, el programa accede de manera dinámica a las funciones situadas en un archivo aparte. N programas diferentes pueden acceder a las funciones propuestas por la librería. Las librerías agrupan funciones propias de un dominio o un conjunto de dominios coherentes: tratamiento de imágenes, sonido, acceso a una base de datos, etc.

Un conjunto de funciones propuestas por una o varias librerías compartidas forma una **API**, *Application Programming Interface*, y a veces se agrupan en un framework que ofrece una solución completa para un dominio dado.

Se establece un vínculo entre el programa y una librería compartida durante la etapa de la edición de los vínculos por el editor de vínculos **ld**, al cual puede llamar el compilador **gcc** con la opción **-l<lib>**.

Otra posibilidad para un programa consiste en utilizar la función **C dlopen**, que abre una librería dinámica como un archivo y que accede a las funciones contenidas mediante punteros de funciones.

Si un programa depende de una librería compartida y ésta está ausente, el programa no podrá funcionar nunca más.

En Linux (y Unix en general), las librerías compartidas se llaman **Shared Objects** (so) en el sentido en el que se trata de archivos de objetos sin bloque de instrucción **main**. Los nombres de los archivos asociados llevan el sufijo **.so**.

Una librería puede disponer de varias versiones, que pueden ser o no compatibles, y se puede precisar la versión durante la edición de los vínculos. También es posible determinar una versión por defecto.

2. Lugar de almacenamiento

Por convención se colocan las librerías compartidas en directorios llamados lib:

- ✓ **/lib**: librerías de sistema básicas, vitales.
- ✓ **/lib64**: librerías de sistema de 64 bits.
- ✓ **/usr/lib**: librerías de usuario básicas, no necesarias para el boot.
- ✓ **/usr/local/lib**: librerías locales para los programas para la máquina.
- ✓ **/usr/X11R6/lib**: librerías del entorno X Window.
- ✓ ...

```
$ find /usr/lib -name "*.so" -print | head
/usr/lib/indicators3/7/libapplication.so
/usr/lib/indicators3/7/libappmenu.so
/usr/lib/indicators3/7/libprintersmenu.so
/usr/lib/libmpathcmd.so
/usr/lib/ghostscript/9.50/X11.so
/usr/lib/open-vm-tools/plugins/common/libvix.so
/usr/lib/open-vm-tools/plugins/common/libhgfsServer.so
/usr/lib/open-vm-tools/plugins/vmsvc/libapplInfo.so
/usr/lib/open-vm-tools/plugins/vmsvc/libdeployPkgPlugin.so
/usr/lib/open-vm-tools/plugins/vmsvc/libpowerOps.so
...
```

En las distribuciones Linux recientes (las basadas en systemd), los directorios **/lib** y **/lib64** ya no están en la raíz, su contenido se ha cambiado a **/usr/lib** o **/usr/lib64**.

Por otro lado, con el periodo de transición de 32 a 64 bits terminado, y como casi todo está ahora disponible en 64 bits, un directorio como **/usr/lib** solo contiene bibliotecas de 64 bits.

```
$ find /usr/lib -name "*.so" -exec file {} \;
/usr/lib/indicators3/7/libapplication.so: ELF 64-bit LSB shared object,
```

```
x86-64, version 1 (SYSV), dynamically linked, BuildID[sha1]=
07e7f23da0d05192fe0cef57fba6c044f6d85e50, stripped
/usr/lib/indicators3/7/libappmenu.so: ELF 64-bit LSB shared object, x86-64,
version 1 (SYSV), dynamically linked, BuildID[sha1]=
366e5ab1efebd833719879ab9eca85e8367e9597, stripped
/usr/lib/indicators3/7/libprintersmenu.so: ELF 64-bit LSB shared object,
x86-64, version 1 (SYSV), dynamically linked, BuildID[sha1]=
d8ad0786791a421831f2dfb9fe0804d1da09f019, stripped
...
```

La librería más importante del sistema es la C. Todos los programas compilados están vinculados a libc. Basta suprimir este archivo (un error de principiante) para hacer caer todo el sistema.

```
$ ls -l /usr/lib/x86_64-linux-gnu/libc.so.6
lrwxrwxrwx 1 root root 12 dic 16 11:04 /usr/lib/x86_64-linux-gnu/libc.so.6
-> libc-2.31.so
```

Los directorios de las librerías contienen muchos vínculos simbólicos. Estos vínculos están aquí, entre otras cosas, para gestionar las versiones y la compatibilidad entre las versiones (como sucede, por ejemplo, cuando dos versiones coexisten), pero también para indicar cuál es la biblioteca principal, cuyo sufijo es ".so" o la versión de la biblioteca, ".so.2":.

```
lrwxrwxrwx 1 root root 17 sept. 19 12:08 libsane.so.1 -> libsane.so.1.0.27
-rw-r--r-- 1 root root 186928 sept. 19 12:08 libsane.so.1.0.27 lrwxrwxrwx
1 root root 17 sept. 19 12:08 libsane.so.1 -> libsane.so.1.0.27
-rw-r--r-- 1 root root 186928 sept. 19 12:08 libsane.so.1.0.27
```

3. ¿Qué librerías vinculadas hay?

El comando **ldd** permite determinar cuáles son las librerías vinculadas a un programa, y

también si están presentes o no.

```
$ ldd nginx
linux-vdso.so.1 (0x00007ffcee8ce000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f036b8d7000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007f036b8b4000)
libcrypt.so.1 => /lib/x86_64-linux-gnu/libcrypt.so.1 (0x00007f036b879000)
libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007f036b806000)
libz.so.1 => /lib/x86_64-linux-gnu/libz.so.1 (0x00007f036b7ea000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f036b5f8000)
/lib64/ld-linux-x86-64.so.2 (0x00007f036b9d6000)
```

Pongamos ahora por caso que falta una librería (movida voluntariamente para poder hacer esta demostración):

```
$ ldd /usr/bin/esd
linux-gate.so.1 => (0xffffe000)
libwrap.so.0 => /lib/libwrap.so.0 (0xb7f6d000)
libesd.so.0 => not found
libasound.so.2 => /usr/lib/libasound.so.2 (0xb7eb1000)
libaudiofile.so.0 => /usr/lib/libaudiofile.so.0 (0xb7e8e000)
libm.so.6 => /lib/libm.so.6 (0xb7e69000)
libc.so.6 => /lib/libc.so.6 (0xb7d36000)
libdl.so.2 => /lib/libdl.so.2 (0xb7d31000)
libpthread.so.0 => /lib/libpthread.so.0 (0xb7d1a000)
/lib/ld-linux.so.2 (0xb7f9f000)
```

Falta la librería `libesd.so.0`. Es imposible ejecutar el programa:

```
$ ./esd
./esd: error while loading shared libraries: libesd.so.0: cannot
open shared object file: No such file or directory
```

4. Configurar la caché del editor de vínculos

La edición de los vínculos con una librería compartida es dinámica y el sistema lo hace en el momento de la ejecución del programa con la ayuda de la librería **ld.so**. El binario proporciona el nombre de las librerías a vincular con la ejecución, pero no la ruta. Las funciones de **ld.so** determinan la librería a utilizar en función de su nombre, entre las rutas que conocen.

Cualquier programa se vincula a la librería **ld.so** o más bien **ld-linux.so** (ld-linux.so.2).

```
alejandro@ubuntu:/lib/x86_64-linux-gnu$ ls -l ld-*.so*
-rwxr-xr-x 1 root root 191472 dic 16 11:04 ld-2.31.so
lrwxrwxrwx 1 root root 10 dic 16 11:04 ld-linux-x86-64.so.2 -> ld-2.31.so
```

El cargador de vínculos **ld.so** busca las librerías en varios lugares, entre los cuales, se encuentran, por este orden:

- ~ las rutas precisadas en la variable de entorno **LD_LIBRARY_PATH**. Se separan las rutas, como para PATH, por «:».
- ~ el contenido del archivo **/etc/ld.so.cache**, que contiene una lista compilada (formato binario) de las librerías encontradas en las rutas predefinidas.
- ~ los directorios **/lib** y **/usr/lib**.

La búsqueda en **/lib** y **/usr/lib** está implícita. Por otro lado, el hecho de rellenar la variable **LD_LIBRARY_PATH** no impide la búsqueda de las librerías en otros sitios, si no está en una de las rutas de la lista.

Para evitar la instalación de una variable cuyo contenido se puede manejar con dificultad, **ld.so** propone una caché que puede modificar por sí mismo. Se construye la caché desde el contenido del archivo **/etc/ld.so.conf** y del comando **ldconfig**. A menudo, solamente contiene una línea para cargar el contenido del directorio **/etc/ld.so.conf.d**.

Este archivo contiene la lista de los directorios que contienen las librerías compartidas:

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

En vez de modificar este archivo, un paquete o usted mismo puede decidir añadir un archivo en `/etc/ld.so.conf.d` que contenga la ruta o las rutas de sus nuevas librerías.

No basta con añadir la ruta: debe regenerar la caché con el comando **ldconfig**.

```
# ldconfig
```

El comando **ldconfig**:

- ▾ actualiza la caché para las rutas definidas en `/etc/ld.so.conf` y asociadas, así como para `/usr/lib` y `/lib`;
- ▾ actualiza los vínculos simbólicos en las librerías;
- ▾ permite también listar las librerías conocidas en la caché.

Se aceptan las siguientes opciones:

Opción	Papel
-v	Modo verboso: indica lo que ldconfig efectúa
-N	No vuelve a construir la caché
-X	No actualiza los vínculos
-p	Lista el contenido de la caché

Para listar las librerías conocidas del editor de vínculos utilice:

```
# ldconfig -p | head -20
903 bibliotecas se encontraron en la caché «/etc/ld.so.cache»
libzypp.so.1702 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzypp.so.1702
```

```

libzstd.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libzstd.so.1
libz.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libz.so.1
libz.so (libc6,x86-64) => /lib/x86_64-linux-gnu/libz.so
libyelp.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libyelp.so.0
libyaml-0.so.2 (libc6,x86-64) => /lib/x86_64-linux-gnu/libyaml-0.so.2
libyajl.so.2 (libc6,x86-64) => /lib/x86_64-linux-gnu/libyajl.so.2
libxtables.so.12 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxtables.so.12
libxslt.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxslt.so.1
libxshmfence.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxshmfence.so.1
libxml2.so.2 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxml2.so.2
libxmlsec1.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxmlsec1.so.1
libxmlsec1-openssl.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxmlsec1-openssl.so.1
libxmlsec1-nss.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxmlsec1-nss.so.1
libxmlb.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxmlb.so.1
libxklavier.so.16 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxklavier.so.16
libxkbfile.so.1 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxkbfile.so.1
libxkbcommon.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxkbcommon.so.0
libxkbcommon-x11.so.0 (libc6,x86-64) => /lib/x86_64-linux-gnu/libxkbcommon-x11.so.0
...

```

Para saber lo que haría `ldconfig` pero sin actualizar nada, use:

```

# ldconfig -N -X -v

/usr/lib/x86_64-linux-gnu/libfakeroot:
  libfakeroot-0.so -> libfakeroot-tcp.so
/usr/local/lib:
/lib/x86_64-linux-gnu:
  libsmbldap.so.2 -> libsmbldap.so.2
  libsemanage.so.1 -> libsemanage.so.1
  libisns.so.0 -> libisns.so.0
  libe2p.so.2 -> libe2p.so.2.3
  liblightdm-gobject-1.so.0 -> liblightdm-gobject-1.so.0.0.0
  libnss_mdns6.so.2 -> libnss_mdns6.so.2
...

```

Finalmente, para actualizar y ver el resultado use:



```
# ldconfig -v
```

(la salida es la misma que la del comando anterior).