– LPI 101 –

Manage Shared Libraries [3]

(Linux Professional Institute Certification)

a


```
   .~.          from IBM developerWorks tutorial
   /V\          geoffrey robertson
  // \\         geoffrey@zip.com.au
  @._.@
```



```
$Id: gl1.102.4.slides.tex,v 1.4 2003/05/30 05:09:04 waratah Exp $
```

Manage Shared Libraries

Objective

Candidates should be able to determine the shared libraries that executable programs depend on and install them when necessary. Candidates should be able to state where system libraries are kept.

Manage Shared Libraries

Key files, terms, and utilities

`ldd`
`ldconfig`

`/etc/ld.so.conf`
`LD_LIBRARY_PATH`

Manage Shared Libraries

Resources of interest

Shared-Library HOWTO :

`http://linuxdocs.org/HOWTOs/Program-Library-HOWTO/`

LPI certification 102 exam prep, Part 1 :

`http://ibm.com/developerWorks`

Introducing shared libraries

On Linux systems there are two fundamentally different types of Linux executable programs.

Introducing shared libraries

On Linux systems there are two fundamentally different types of Linux executable programs.

statically linked executables :

contain all the functions that they need to execute

Introducing shared libraries

On Linux systems there are two fundamentally different types of Linux executable programs.

statically linked executables :

contain all the functions that they need to execute

dynamically linked executables :

require libraries of functions

Static vs. Dynamic Executables

`ldd`

We can use the `ldd` command to determine if a particular executable program is static:

```
$ ldd /sbin/sln ↩
        not a dynamic executable
```

Static vs. Dynamic Executables

ldd

We can use the `ldd` command to determine if a particular executable program is static:

```
$ ldd /sbin/sln ↩
        not a dynamic executable
```

Relative Size

```
$ ls -l /bin/ln /sbin/sln ↩
-rwxr-xr-x 1 root root 23000  Jan 14 00:36 /bin/ln
-rwxr-xr-x 1 root root 381072 Jan 14 00:31 /sbin/sln
```

Dynamically Linked Dependencies

- To view a list of all the shared libraries upon which ln depends, use the ldd command:

```
$ ldd /bin/ln ↩
libc.so.6 => /lib/libc.so.6 (0x40021000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Dynamically Linked Dependencies

- To view a list of all the shared libraries upon which ln depends, use the ldd command:

```
$ ldd /bin/ln ↩
libc.so.6 => /lib/libc.so.6 (0x40021000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

- As a rule, dynamically linked programs are much smaller than their statically-linked equivalents.

Dynamically Linked Dependencies

- To view a list of all the shared libraries upon which ln depends, use the ldd command:

```
$ ldd /bin/ln ↩
libc.so.6 => /lib/libc.so.6 (0x40021000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

- As a rule, dynamically linked programs are much smaller than their statically-linked equivalents.
- `/lib/ld-linux.so.2` is the loader

The dynamic loader

- The dynamic loader takes care of loading the shared libraries that dynamically linked executables need in order to run.

The dynamic loader

- The dynamic loader takes care of loading the shared libraries that dynamically linked executables need in order to run.
- The dynamic loader finds shared libraries thanks to two files – `/etc/ld.so.conf` and `/etc/ld.so.cache`.

The dynamic loader

- The dynamic loader takes care of loading the shared libraries that dynamically linked executables need in order to run.
- The dynamic loader finds shared libraries thanks to two files – `/etc/ld.so.conf` and `/etc/ld.so.cache`.
- The contents of `/etc/ld.so.conf`:

```
cat /etc/ld.so.conf
/usr/X11R6/lib
/usr/lib/gcc-lib/i686-pc-linux-gnu/2.95.3
/usr/lib/mozilla
/usr/lib/qt-x11-2.3.1/lib
/usr/local/lib
```

ld.so.cache

Before the dynamic loader can ”see” this information, it must be converted into an `ld.so.cache` file.

This is done by running the ldconfig command:

```
$ ldconfig ↩
```

When `ldconfig` completes, you now have an up-to-date `/etc/ld.so.cache` file that reflects any changes you've made to `/etc/ld.so.conf`.

`ldconfig` tips

To view all the shared libraries that `ldconfig` can ”see,” type:

```
$ ldconfig -p | less ↩
```

Sometimes, you’ll want to tell the dynamic loader to try to use shared libraries in a specific directory before trying any of your `/etc/ld.so.conf` paths.

For older application requiring older libraries.

LD_LIBRARY_PATH

- To instruct the dynamic loader to check a certain directory first, set the `LD_LIBRARY_PATH` variable to the directories that you would like searched.

LD_LIBRARY_PATH

- To instruct the dynamic loader to check a certain directory first, set the LD_LIBRARY_PATH variable to the directories that you would like searched.
- Separate multiple paths using colons; for example:

```
# export LD_LIBRARY_PATH="/usr/lib/old:/opt/lib" ↩
```

LD_LIBRARY_PATH

- To instruct the dynamic loader to check a certain directory first, set the `LD_LIBRARY_PATH` variable to the directories that you would like searched.
- Separate multiple paths using colons; for example:

  ```
  # export LD_LIBRARY_PATH="/usr/lib/old:/opt/lib" ↩
  ```

- After `LD_LIBRARY_PATH` has been exported, any executables started from the current shell will use libraries in `/usr/lib/old` or `/opt/lib` if possible, falling back to the directories specified in `/etc/ld.so.conf` if some shared library dependencies are still unsatisfied.

Summary

- **ldd**

Summary

- **ldd**
- Lists the required libraries

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**
- list of library paths

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**
- list of library paths
- **`ldconfig`**

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**
- list of library paths
- **`ldconfig`**
- Updates `/etc/ls.so.cache` from `ld.so.conf`

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**
- list of library paths
- **`ldconfig`**
- Updates `/etc/ls.so.cache` from `ld.so.conf`
- **`LD_LIBRARY_PATH`**

Summary

- **ldd**
- Lists the required libraries
- **ld.so.conf**
- list of library paths
- **`ldconfig`**
- Updates `/etc/ls.so.cache` from `ld.so.conf`
- **`LD_LIBRARY_PATH`**
- Holds library path for current shell

The End