6.4 Shared Libraries

As you study this section, answer the following questions:

- What is the purpose of a shared library?
- What is the difference between a dynamic shared library and a static shared library?
- What does it mean if a library file name contains **.so**?
- Which command would you use to identify library dependencies?
- What are the management complications for shared libraries?
- Which file contains a cached list of dynamic shared libraries?
- What are the methods for configuring additional dynamic libraries on a Linux system?

In this section, you will learn to:

- Identify a daemon's required shared libraries.
- Recognize a broken library link.

Key terms for this section include the following:

Term	Definition
Shared library	A library of routines that can be used by other applications.
Dynamic library	A shared library that is not directly integrated into the code of a software application.
Static library	A shared library, usually installed with an application, the is integrated into the code of a software application.
ldd	A utility that shows the shared libraries required by an application.
Locally compiled software	Software that is compiled from downloaded or custom source code.
Local repository	A local storage location that contains software packages for a Linux distribution.

This section helps you prepare for the following certification exam objectives:

Exam	Objective
CompTIA Linux+	2.1 Given a scenario, conduct software installations, configurations, updates, and removals. • Build tools ○ Commands ■ make ■ make install ■ ldd

	◦ Compilers ◦ Shared libraries • Repositories ◦ Configuration ◦ Creation ◦ Syncing ◦ Locations • Acquisition commands ◦ wget ◦ curl