The GNU C library, popularly known as Glibc, is the unseen force that makes GCC, most C language applications compiled with GCC on Linux systems, and all GNU/Linux systems work. Any code library provides a set of functions that simplify writing certain types of applications. In the case of the GNU C library, the applications facilitated by Glibc are any C programming language applications. The functions provided by Glibc range from functions as fundamental to C as printf() to Portable Operating System Interface for Computer Environments (POSIX) functions for opening low-level network connections (more about the latter later). The C programming language actually has a relatively small number of keywords; most of what people think of as C is actually standard C library functions that are implemented in Glibc.

Though Glibc is quite stable (aptly demonstrated by the fact that most Linux applications depend on it), it is indeed software. You may want to upgrade to a newer version for a variety of reasons, the most common of which are to resolve some problem that you’ve discovered, to take advantage of improvements in how existing functions are implemented in a newer version of Glibc, or to take advantage of new functions provided in the latest and greatest versions.

Rebuilding and changing the library on which almost every GNU/Linux application depends can be an intimidating thought. Luckily, it is not as problematic or complex as you might think. This chapter explains how to obtain, build, and install newer versions of the GNU C library, discusses problems that you might encounter, and also explains how to work around those problems in order to get your system(s) up and running with newer or alternate versions of the GNU C library that may be present by default on the system(s) that you are using.

Note  The most important requirement for upgrading Glibc on your system is having sufficient disk space to download and build the source code, back up files that you want to preserve in case you encounter upgrade problems, and install the new version of Glibc. You should make sure that your system has approximately 200MB of free space in order to build and install Glibc.

What Is in Glibc?

Because most of the fundamental Un*x, Linux, HURD, and BSD applications are written in the C programming language, every Unix-like operating system needs a C library to provide the basic capabilities required in C applications and to provide the system calls that enable C applications to interface with the operating system. Glibc is the one true C library in the GNU system, and in most newer systems with the Linux kernel.
The contents of interfaces provided as part of Glibc have evolved over time and reflect the history of Unix and relevant standards. Glibc provides support for the following standards and major Un*x variants:

- **Berkeley Standard Distribution (BSD):** No one with any history on Un*x systems could be unaware of the enhancements to Un*x that were provided by the Berkeley Standard Distribution. NetBSD, FreeBSD, and even Apple’s Mac OS X still carry the flag of many of the networking, I/O, and usability improvements made to Un*x by the University of California at Berkeley and other academic institutions such as Carnegie Mellon University, and long-lived BSD-based Un*x implementations such as Sun Microsystems SunOS. Glibc supports many of the capabilities found in the 4.2 BSD, 4.3 BSD, and 4.4 BSD Unix systems, and in SunOS. This heightens code compatibility with 4.4 BSD and later SunOS 4.X distributions, which themselves support almost all of the capabilities of the ISO C and POSIX standards.

- **ISO C:** This is the C programming language standard adopted by the American National Standards Institute (ANSI) in the “American National Standard X3.159-1989—ANSI C” document, and later by the International Standardization Organization (ISO) in its “ISO/IEC 9899:1990, Programming Languages—C” document. This is colloquially referred to as the **ISO C standard** throughout the Glibc documentation.

**Note** The header files and functions provided by Glibc are a superset of those specified in the ISO C standard. If you need to write applications that strictly follow the ISO C standard, you must use the -ansi option when compiling programs with GCC. This will identify any non-ANSI constructs that you might have used accidentally.

- **POSIX:** This is the Portable Operating System Interface for Computer Environments document (ISO/IEC 9945-1:1996), later adopted by ANSI and the IEEE (Institute of Electrical and Electronics Engineers) as ANSI/IEEE Std 1003. The POSIX standard has its roots in Unix systems and was designed as a standard that would facilitate developing applications that could be compiled across all compliant Un*x systems. The POSIX standard is a superset of the ISO C standard, adding new functions and extending existing functions in the ISO C standard. If you need to sling applicable acronyms, the Glibc manual states that Glibc is compliant with POSIX, POSIX.1, IEEE Std 1003.1 (and IEEE Std 1003.2, Draft 11), ISO/IEC 9945-1, POSIX.2, and IEEE Std 1003.2. Glibc also implements some of the functionality required in the “POSIX Shell and Utilities” standard (a.k.a. POSIX.2 or ISO/IEC 9945-2:1993).

- **SYSV Unix:** Un*x history began at AT&T Bell Laboratories. Glibc supports the majority of the capabilities specified in the AT&T “System V Interface Description” (SVID) document, which is a superset of the POSIX standard mentioned earlier.

- **XPG:** The “X/Open Portability Guide,” published by the X/Open Company, Ltd., is a general Un*x standard. This document specifies the requirements for systems that are intended to be conformant Unix systems. Glibc complies with the “X/Open Portability Guide, Issue 4.2,” and supports all of the X/Open System Interface (XSI) and X/Open Unix extensions. This should not be a big surprise, since the majority of these are derived from enhancements to Unix made on System V or BSD Unix, and Glibc is compliant with those.

Today’s Glibc has come a long way from the statically linked version 1.x Glibc of the 1980s. Today, Glibc is a powerful set of shared libraries that is used on hundreds of thousands of computer systems all over the world. Like GCC, Glibc is a living testimonial to the power of open source software and the insight and philanthropy of its designers and contributors.