Platform dependencies

In the first part of this book, we looked at the various activities needed to port and install software on a UNIX system. We carefully avoided getting too involved with the nitty-gritty of *why* we should need to go to so much trouble. In this part of the book, we'll look at those differences between platforms which require us to modify software.

As we saw in Chapter 4, *Package configuration*, configuration can be required for local preferences, software dependencies and hardware dependencies. We looked at local preferences in Chapter 4. In this part of the book, we'll look at differences in hardware and software platforms.

Software Dependencies

Probably the biggest problem you will have with configuration will be with the underlying software platform. Even if you limit your scope to the various UNIX versions, 25 years of continuing (and mainly uncoordinated) evolution have left behind a plethora of marginally compatible versions. The only good thing about the situation is that porting between UNIX versions is still an order of magnitude easier than porting to or from a non-UNIX environment.

It's easy to misjudge the effort required to port to a different platform. It helps to make the following very clear distinctions between the following kinds of functionality:

- Functionality that relies on system calls (section 2 of the UNIX manual). These calls interface directly with the kernel. If the kernel doesn't supply the functionality, you may have serious difficulty in porting the product. Good examples are the System V function `shmget`, which allocates an area of shared memory, or the BSD system call `symlink`, which creates a symbolic link.

- Functionality dependent on system library calls (section 3 of the UNIX manual). If these do not rely on system calls, you may be able to port a corresponding call from another library. A good example of this is the function `strcasecmp`, which compares strings ignoring case. This function is supplied with later versions of the BSD library and also with GNU libc, but not with System V libraries. If you don't have it, it's trivial to port.

- Functionality contained totally *inside* the package, like math routines that don't call external libraries. This should work on any platform.

Some systems, such as OSF, have merged sections 2 and 3 of the manual pages. While that has some advantages (if you know a function name, you don't have to go to two different places to look for them), it *doesn't* mean that there is no longer a difference.

Kernel dependencies are significantly more difficult to handle than library dependencies, since there's relatively little you can do about them. We'll look at kernel-related problems in Chapter 12, *Kernel dependencies*, Chapter 13, *Signals*, Chapter 14, *File systems*, Chapter 15, *Terminal drivers*, and Chapter 16, *Timekeeping*. In Chapter 17 we'll look at header files, and in Chapter 18 we'll look at libraries.

In addition to these program dependencies, two tools can differ significantly: the *make* program and the C compiler. We'll look at these aspects in Chapter 19, *Make*, and Chapter 20, *Compilers*. Finally, in Chapter 21, *Object files and friends*, we'll look at some of the more esoteric aspects of object files.

When discussing differences between kernels and libraries, the big difference is usually between System V and BSD, with other systems such as SunOS taking a midfield position. System V.4 incorporates nearly everything in BSD. When programming, you have the choice between using the native System V development tools or the BSD tools. Some admixture is possible, but it can cause problems.

When using BSD development tools, everything that is supported by BSD should also be supported by System V.4. On the other hand, System V.4 also includes some functionality that no other system provides. When, in the following chapters, I say that a function is supported by System V.4, I mean that it is supported by System V.4 using the standard development tools and libraries. If I state that it is supported by BSD, it also implies that it is supported by System V.4 using the BSD libraries.