UNIX Kernel:

Technically speaking, the UNIX kernel "is" the operating system. It provides the basic full time software connection to the hardware. By full time, it means that the kernel is always running while the computer is turned on. When a system boots up, kernel is loaded. Likewise, the kernel is only exited when the computer is turned off.

The UNIX kernel is built specifically for a machine when it is installed. It has a record of all the pieces of hardware it needs to talk to and knows what languages they speak (how to turn switches on and off to get a desired result). Thus, a kernel is not easily ported to another computer. Each individual computer will have its own tailor- made kernel. If the computer's hardware configuration changes during its life, the kernel must be "rebuilt" (told about the new pieces of hardware).

However, though the connection between the kernel and the hardware is "hardcoded" to a specific machine, the connection between the user and the kernel is generic. That is the beauty of the UNIX kernel. From your perspective, regardless of how the kernel interacts with the hardware, no matter which UNIX computer you use, you will have the same kernel interface to work with. That is because the hardware is "hidden" by the kernel.

The kernel also handles memory management, input and output requests, and process scheduling for time-shared operations (we'll talk more about what this means later).
To help it with its work, the kernel also executes daemon programs which stay alive as long as the machine is turned on and help perform tasks such as printing or serving web documents.

However, the task of hiding the hardware is a pretty much full time job for the kernel. As such, it does not have too much time to provide for a fancy user-friendly interface. Thus, though the kernel is much easier to talk to than the hardware, the language of the kernel is still pretty cryptic.

Fortunately, the UNIX operating system has built in "shells" which wrap around the kernel and provide a much more user-friendly interface. Let's take a look at shells.