1.7 Real-Time Systems

Another form of a special-purpose operating system is the **real-time system.** A real-time system is used when rigid time requirements have been placed on the operation of a processor or the flow of data; thus, it is often used as a control device in a dedicated application.

A real-time system has well-defined, fixed time constraints. Processing *must* be done within the defined constraints, or the system will fail. A real-time system functions correctly only if it returns the correct result within its time constraints. Contrast this requirement to a time-sharing system, where it is desirable (but not mandatory) to respond quickly, or to a batch system, which may have no time constraints at all.

Real-time systems come in two flavors: hard and soft. A hard real-time system guarantees that critical tasks be completed on time. This goal requires that all delays in the system be bounded, from the retrieval of stored data to the time that it takes the operating system to finish any request made of it.

Secondary storage of any sort is usually limited or missing, with data instead being stored in read-only memory (ROM). ROM is located on nonvolatile storage devices that retain their contents even in the case of electric outage; most other types of memory are volatile. Most advanced operating-system features are absent too, since they tend to separate the user from the hardware, and that separation results in uncertainty about the amount of time an operation will take. For instance, virtual memory is almost never found on real-time systems. Therefore, hard real-time systems conflict with the operation of time-sharing systems, and the two cannot be mixed. Since none of the existing general-purpose operating systems support hard real-time functionality. A less restrictive type of real-time system is a soft real-time system, where a critical real-time task gets priority over other tasks,

and retains that priority until it completes. As in hard real-time systems, the operating-system kernel delays need to be bounded: A real-time task cannot be kept waiting indefinitely for the kernel to run it. Soft real time is an achievable goal that can be mixed with other types of systems. They are useful, however in several areas, including multimedia, virtual reality, and advanced scientific projects. These systems need advanced operating-system features that cannot be supported by hard real-time systems. Because of the expanded uses for soft real-time functionality, it is finding its way into most current operating systems, including major versions of UNIX.

1.8 Handheld Systems

Handheld systems include personal digital assistants (PDAs), such as *Palm*-Pilots or cellular telephones with connectivity to a network such as the Internet. Developers of handheld systems and applications face many challenges, most of which are due to the limited size of such devices. For example, a PDA is typically about 5 inches in height and **3** inches in width, and it weighs less than one-half pound. Due to this limited size, most handheld devices have a small amount of memory, include slow processors, and feature small display screens.

Many handheld devices have between 512 KB and 8 MB of memory. (Contrast this with a typical PC or workstation, which may have several hundred megabytes of memory!) As a result, the operating system and applications must manage memory efficiently.

Currently, many handheld devices do not use virtual memory techniques, thus forcing program developers to work within the confines of limited physical memory.

A second issue of concern to developers of handheld devices is the speed of the processor used in the device. Processors for most handheld devices often run at a fraction of the speed of a processor in a PC. Faster processors require more power. To include a faster processor in a handheld device would require a larger battery that would have to be replaced (or recharged) more frequently. To minimize the size of most handheld devices, smaller, slower processors which consume less power are typically used. Therefore, the operating system and applications must be designed not to tax the processor. The last issue confronting program designers for handheld devices is the small display screens typically available. Whereas a monitor for a home computer may measure up to 21 inches, the display for a handheld device is often no more than **3** inches square. Familiar tasks, such as reading e-mail or browsing web pages, must be condensed onto smaller displays.

Some handheld devices may use wireless technology, such as BlueTooth, allowing remote access to e-mail and web browsing. Cellular telephones with connectivity to the Internet fall into this category. However, many PDAs currently do not provide wireless access. To download data to these devices, typically one first downloads the data to a PC or workstation, and then downloads the data to the PDA. Some PDAs allow data to be directly copied from one device to another using an infrared link. Generally, the limitations in the functionality of PDAs are balanced by their convenience and portability. Their use continues to expand as network connections become more available and other options, such as cameras and MP3 players, expand their utility.