The Cooperative Multitasking Environment

The cooperative multitasking environment is a standard part of system 7.0. The Macintosh Operating System and the Finder work together to provide this environment. MultiFinder is now transparent to the user; the user always has the capability to run more than one application at a time. Because the user may choose to run other applications in addition to your application, your application needs to be capable of existing in a shared environment.

The Operating System schedules the processing of all applications and desk accessories. When a user opens a document or application, the Operating System loads the application code into memory and schedules the application to run. The application runs at the next available opportunity. The next available opportunity usually means when the current process or application gives up the CPU. In most cases, the application runs immediately (or appears to the user to run immediately).

Once an application is executing, the CPU is available only to that application. The application can only be interrupted by hardware interrupts, and these are transparent to the application. However, to allow the user to interact with your application and others, you must periodically relinquish the CPU using the **WaitNextEvent** or **EventAvail** function. Using these event routines in your application lets the user interact with your application and also with other applications.

Although the user can have a number of open documents and applications, only one application is the active application. The active application is the application currently interacting with the user; its icon appears in the right side of the menu bar. The active application displays its menu bar and is responsible for highlighting the controls of its frontmost window.

When your application is the active application and the user switches to another application (by clicking in the window of a document belonging to another application, for example), the Operating System sends your application a suspend event. When your application receives a suspend event, it should prepare to suspend processing, allowing the user to switch to the other application. For example, in response to a suspend event, your application should remove the highlighting from the controls of its frontmost window and take any other necessary actions. The suspension actually occurs the next time your application calls **WaitNextEvent** or **EventAvail**.

Your application also needs to be able to resume processing when the user chooses to work with your application again. Your application receives a resume event when the user switches back to your application. In response to a resume event, your application should update the contents of its windows and highlight the controls of its frontmost window.

The Operating System preserves the environment of your application when it is suspended and restores that environment before sending it a resume event. Your application does not need to preserve or restore the operating environment in response to suspend or resume events.

When you perform user testing of your application, you might want to observe people using other applications as well as your application, to make sure that your application works well in a cooperative environment.

See the **Compatibility Guidelines** and the **Event Manager** for specific information on how your application can handle suspend and resume events and how your application can take advantage of the cooperative multitasking environment.