

Problem Set 2

OUT: 9 FEBRUARY, 1993
DUE: 16 FEBRUARY, 1993

MIDTERM REPORT! Your mid-term papers will be due 2 March.

1. Help me solve this problem. I have two computers: Passaic and Venice. I am trying to make a computer animation requiring the resources of both computers. Passaic is smart, but Venice is artistically inclined. Passaic can create images using mathematical calculations, but cannot display them. Venice can display images beautifully but isn't fast enough to crunch all the numbers.

The interconnection between the two computers is the file system. Both computers can read and write a file by name to a common file system. Here are some of the difficulties to be circumvented:

- Just because a file appears on the disk doesn't mean that Venice can display it. Passaic will take some time to write the entire file to disk and it must signal Venice when it is done.
- Passaic must produce more files than can possibly fit on the disk. Once Venice displays a file, it can erase it. If the disk fills up, Passaic should wait until Venice has deleted an older file.
- The only method of communication is through the shared disk. The solutions in the book assume the processes can talk through shared memory.

Suggest a method of coordination.

2. The set-up is similar to the preceding problem. However, now it is a matter of the two computers wanting to share a single file. Adapt Dekker's algorithm to this situation, where the computers have access only to a common file system. (HINT: Use the existence and non-existence of files as a flag.)
3. The MacIntosh and other windowing systems use *events* to direct input from the keyboard or the mouse to the interested process. Windowing systems consist of many user processes, each running in a separate

window, a rectangular region on the screen which you can point to, click upon, and type into.

In fact, windows are arranged as subwindows in a parent/child structure. The screen is the *root* window, the common ancestor of all windows. A pop-up menu or scroll bar is a child subwindow of the window in which it physically resides.

The user communicates to the window by pointing at the window with the mouse and either typing at the keyboard or clicking the mouse buttons. An appropriate *keyboard event* or *mouse event* is created by the operating system and sent to the window or subwindow which is the current *focus*, that is: the focus of attention. If the subwindow is not interested in the event, for instance, typing into a window which just has a button to mouse-click upon, the event is automatically passed upwards to the parent window. This might continue all the way to the root, which simply discards the event.

A subwindow, when created, calls the operating system to *register interest* in an event. During the delivery of an event the operating system checks with the window's registered interests to decide whether the event has found its target or must be passed to the parent.

Use Message-Passing discussed in class to sketch an implementation of events.

4. State five short-term scheduling policies.