to up to 62 (31 each for system and user) by segment switching. Only a single user library and system library map could be active at any one time.

- Message queue size proved to be a problem. The monitor processes sent status messages at regular intervals to every process that wanted them. If the process didn't read the messages, large numbers of resources (LCBs and message buffers) could be used for duplicate messages. To address this problem, Guardian II introduced a *messenger* process that kept a single copy of these status messages and sent them to a process when it called `listen`.

Missed Opportunities

The T/16 was a revolutionary machine, but it also offered an environment that few other machines of the day had. Ultimately, though, it was the small things that got in the way. For example, device independence is one of the most enduring aims of operating systems, and Tandem went a long way toward this goal. Ultimately, though, they missed their full potential because of naming issues and almost gratuitous incompatibilities. Why was it not possible to use `read` in interprocess communication? Why did process names have to differ in format from device names? Why did they need a `#` character in the ninth byte?

Split Brain

A more serious issue was with the basic way of detecting errors. It worked fine as long as only one component failed, and usually quite well if two failed. But what if both interprocessor buses failed? Even in a two-CPU system, the results could be catastrophic. Each CPU would assume that the other had failed and take over the I/O devices—not once, but continually. Such circumstances did occur, fortunately very rarely, and they often resulted in complete corruptions of the data on disks shared between the two CPUs.

Posterity

From 1990 on, a number of factors contributed to a decline in Tandem's sales:

- Computer hardware in general was becoming more reliable, which narrowed Tandem's edge.
- Computer hardware was becoming *much* faster, highlighting some of the basic performance limitations of the architecture.

In the 1990s, the T/16 processor architecture was replaced by a MIPS-based solution, though much of the remaining architecture remained in place. On the other hand, the difference in performance was big enough that as late as 2000, Tandem was still using the MIPS processors to emulate the T/16 instructions. One of the reasons was that most Tandem system-level software was still written in TAL, which was closely coupled to the T/16 architecture. Moves to migrate the codebase to C were rejected because of the cost involved.