Thoth, a Portable Real-Time Operating System

David R. Cheriton
Michael A. Malcolm
Lawrence S. Melen
Gary R. Sager

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

Extended Abstract

Thoth is a portable real-time operating system which has been developed at the University of Waterloo. Various configurations of Thoth have been running since May 1976; it is currently running on two minicomputers with quite different architectures (Texas Instruments 990 and Data General NOVA).

This research is motivated by the difficulties encountered when moving application programs from one system to another; these difficulties arise when interfacing with the hardware and support software of the target machine. The problems encountered interfacing with the new software support are usually more difficult than those of interfacing with new hardware because of the wide variety of abstract machines presented by the compilers, assemblers, loaders, file systems and operating systems of the various target machines. We have taken the approach of developing portable system software and porting it to "bare" hardware. Because the *same* system software is used on different hardware, the same abstract machine is available to application programs. Thus most application programs which use Thoth can be portable, if not machine independent.

Most previous work on software portability has focused on problems of porting programs over different operating systems as well as hardware. To our knowledge, this is the first time an entire system has been ported. Our experience indicates that this approach is practical both in the cost of porting the system and its time and space performance.

The design of Thoth strives for more than portability. A second design goal is to provide a system in which programs can be structured using many small concurrent processes. Thus we have aimed for efficient interprocess communication to make this structuring technique attractive. We have also provided safe dynamic process creation and destruction.

A third design goal is that the system meet the demands of real-time applications. To meet this goal, the system guarantees that the worst-case time for response to certain external events (interrupt requests) is bounded by a small constant.

A fourth design goal is that the system be adaptable to a wide range of real-time applications. A range of system configurations is possible: A stand-alone application program can use a stripped version of the Thoth kernel which supports dynamic memory allocation and interprocess communication. Such a configuration requires less than 2000 16-bit words of memory. Larger configurations can support process destruction, a device-independent input-output system, a tree-structured file system, and multiple teams of processes. (A *team* is a set of processes which share the same logical address space and therefore can share data.)

Thoth is implemented in a high-level language called Eh (a descendant of BCPL) and a small amount of assembly language. The major job in porting the system seems to be in redesigning the code generation parts of the compiler.

Since it appears impractical to design system software to be portable over all computers (even over all existing machines), we have aimed at making Thoth portable over a subset of machines. Machines in the set can be characterized by a set of properties such as: a word must be at least 16 bits in length, a pointer to a word must fit into a word, etc. Roughly, this set of machines includes most modern minicomputers. It is important that many machines which do not yet exist will be included in it.

A number of application programs have been written using Thoth. In addition to software development tools, communications and real-time control programs have been written. All of these programs require few if any changes when ported to new hardware. Some of these programs have been developed by inexperienced programmers who were not planning on porting their program. Hence, it seems to take less skill to write portable software in this system than using conventional techniques. However, existing software written for other systems is incompatible with Thoth and usually difficult to port to the Thoth system.

Although, at the time of this writing, we have limited experience with porting the system to new hardware, we feel that Thoth has been highly successful in terms of our original objectives. Among other things, it has partially demonstrated the feasibility of building a portable operating system for a specified class of machines.