Operating Systems

B. RANDELL, Editor

Implementation of the SHARER2 Time-Sharing System

M. C. HARRISON
New York University, New York*

A simple mechanism is described for the execution of part of a program with its own memory protection. This allows such a program to act as a suboperating system. An improved version of the SHARER time-sharing system using this feature is described.

KEY WORDS AND PHRASES: operating system, memory protection, time-sharing, multiprogramming, monitor, submonitor, suboperating system
CR CATEGORIES: 4.30, 4.31, 4.32

In a recent paper "SHARER, a Time Sharing System for the CDC 6600" by M. C. Harrison and J. T. Schwartz [1], the implementation of a time-sharing system as a subsystem of a standard nonconversational operating system (SCOPE) was described. Somewhat before the article appeared in print, a substantial improvement was made in the implementation which greatly simplified the design and reduced by an order of magnitude the difficulty of debugging such a subsystem.

In the original subsystem, the executive responsibility was shared between two programs executing in parallel on two processors. This gave rise to a number of programming problems concerned with simultaneous reference to tables in memory accessible to the two programs. Worse than this, one of the processors was not memory-protected and could (and sometimes did) destroy the background system when a bug was encountered in either program. Extensive checking of the potentially dangerous program would have been required to make the system secure.

To solve both these problems, the subsystem was modified to use a single executive program in the fast central processor, which is memory-protected and therefore cannot destroy the rest of the system. Those functions which this program could not do were built in to the standard SCOPE monitor with rigorous error-checking. One of these functions, which we refer to as XJD, has been found to be of use in a number of other situations, and is worth describing.

A program in a SCOPE job, which we will refer to as a "control" program, can issue the request XJD (p,t) where p is a pointer to a 16-word block of memory specifying a "controlled" program whose memory is wholly within the memory of the control program. On receiving such a request, the SCOPE monitor checks the parameters, and if they are legal, switches the central processor to the controlled program. It allows the controlled program to execute for up to t milliseconds, or until it makes an error or a request, when it simply switches the central processor back to the control program giving an indication of the cause of the controlled program's termination. If the controlled program made a request or an error, the control program can service it or pass it on to SCOPE if necessary.

The use of the XJD request is not limited to a particular job. All jobs running in the system can use it with no danger to each other or to the system. Note that if the control program is prepared to recognize and service an XJD request from one of its controlled programs, it is possible for the controlled program to act also as a control program, thus providing a hierarchy of programs or subsystems of effectively arbitrary depth.

The SHARER time-sharing system was rewritten in the summer of 1967 as a control program which uses XJD to execute user programs. This eliminated the need for the special peripheral program, except the teletype communication routine. The latter is implemented as a standard peripheral driver callable by a request in the normal way.

The resulting subsystem, SHARER2, is not distinguished from any other job in any way, and can be run undebugged during development with no danger to any other job running in the system. In fact, it is possible to run two or more such time-sharing subsystems simultaneously (communicating with different sets of terminals, of course).

The idea of a program being able to execute a part of itself is of course not new. It is available in the systems for the GE 645 and the SDS 940, which have much more elaborate addressing facilities than the CDC 6600. On the 6600 the attraction is the extreme simplicity of implementation.

This work was supported by the US Atomic Energy Commision under contract number AT(30-1)1480.

* Courant Institute of Mathematical Sciences.

REFERENCES

1. HARRISON, M. C., AND SCHWARTZ, J. T. SHARER, a time sharing system for the CDC 6600. *Comm ACM 10*, 10 (Oct. 1967), 659–665.
2. Papers on Multics. Proc. AFIPS 1965 Fall Joint Comput. Conf., Vol. 27, Pt. 1, Spartan Books, New York.
3. LAMPSON, B. W., LICHTENBERGER, W. W., AND PIRTLE, M. W. A user machine in a time-sharing system. *Proc. IEEE 54*, 12 (Dec. 1966), 1766–1774.