

## XXIX. COMPUTER RESEARCH\*

### Academic and Research Staff

Prof. J. B. Dennis  
N. Kerllenevich  
Rachel J. Levy

### RESEARCH OBJECTIVES

The purpose of this group, which is operated jointly by the Research Laboratory of Electronics, the Electronic Systems Laboratory, and the Department of Electrical Engineering, M. I. T., is fourfold:

1. To provide a flexible and readily accessible computation facility oriented toward the Laboratory's research goals.
2. To provide an educational facility where students may learn the principles of automatic computation, and undergraduate and graduate thesis projects may be carried out.
3. To develop computation techniques, especially methods that make program design and construction more convenient, and allow easier communication with the machine.
4. To investigate new principles for organizing multiprogrammed computer systems.

J. B. Dennis

### STATUS OF RESEARCH

#### 1. Computing Equipment

The Computer Research Group operates two general-purpose machines. The TX-0 computer was built by Lincoln Laboratory, M. I. T., as an experimental machine, and was turned over to the Department of Electrical Engineering in 1958. The Digital Equipment Corporation donated one of their PDP-1 machines in 1961. The PDP-1 has become the central processor of a time-shared multiuser system<sup>1</sup> implemented by the group. Our activities have included modifications to the two computer systems to improve their performance, as well as the programming of translators, editors, debugging programs, and executive routines suited to the on-line nature of the use of the two systems.

#### 2. Development of an Improved Multiuser System

Experience obtained from operation of the PDP-1 time-sharing system, together with experience at Project MAC, has taught us much about desirable properties of time-shared computer systems, or, in more general terms, multiprogrammed computer

---

\* This work is supported principally by the Joint Services Electronics Programs (U. S. Army, U. S. Navy, and U. S. Air Force) under Contract DA 36-039-AMC-03200(E); and in part by the National Science Foundation (Grant GP-2495) and the National Aeronautics and Space Administration (Grant NsG-496).

systems. Features of a multiprogrammed computer system (MCS) are generally evident in the services that an object program may obtain from the executive program by giving commands that we call meta-instructions. From work done at Project MAC, Dennis and Van Horn<sup>2</sup> have published a set of meta-instructions and defined their meaning in terms of a program structure felt to be particularly suitable for an MCS. The principal ideas in this paper concern parallel processing, relations among computations, and the organization and manipulation of directories of files, etc., retained by the system on behalf of its users. Plans have been made to implement many of these ideas in the next incarnation of our multiuser system. The new system involves hardware additions and modifications, as well as the preparation of a new executive system. Some of the more important aspects of the system design are outlined here.

a. File memory. A file memory system is being added to the PDP-1 installation in the form of miniature tape units (DecTape), so that a user may retrieve programs and data by command from his console. Consistent with the informal way in which the system is operated, a user will ordinarily bring a tape containing his programs with him and mount it on a free unit at the beginning of a session of use, and take the (updated) tape with him when he leaves.

b. Parallel programming. We have found the notion of process to be more fundamental to the design of an MCS than the concept of program. A process is the sequential execution of the steps of a procedure. One process may by means of a fork meta-instruction create a second process which is thought of as running concurrently with the first. The use of a join meta-instruction allows a process to test whether two or more asynchronous activities have been completed. Execution of a quit meta-instruction terminates a process. The fork, join, and quit meta-instructions are basic primitive operations of parallel programming. A particularly efficient implementation of these primitives through a combination of hardware and software techniques is a central feature of the new design. In the new system the use of these primitives to control concurrent input/output activity replaces the interrupt feature that serves this function on essentially all present machines.

c. Computations. We use the term computation to mean a collection of processes that collectively accomplish a computing task. Each computation runs on behalf of a user of the system and has an associated sphere of protection that determines what files, services or directories the processes of the computation are allowed to access or use. In our revised system the sphere of protection is established by a 64-entry list, called a capability list,<sup>2</sup> through which the processes of the computation execute input/output operations, or gain access to directories, files, and other services provided by the executive system. We permit one computation to run a second computation under its

surveillance. This is a valuable asset in program debugging: It allows a programming system running as an object computation to monitor a second object computation under test without the danger of being disabled if a process should run wild or address incorrectly.

d. Directory Structure. The system design includes a hierarchical directory structure. Each item in a directory associates an alphanumeric name with a pointer to a file, an input/output function, an entry point giving access to some system service, or a directory giving further associations of names with pointers. Processes may reference items in a subtree of the directory lattice by specifying a downward path from a directory that is accessible through the capability list of the computation. The root directory of a user of the system is always accessible through the capability list of a computation operating on his behalf. By means of meta-instructions similar to those defined elsewhere,<sup>2</sup> processes are able to share files in a rather flexible way.

J. B. Dennis, N. Kerllenevich, L. J. Rotenberg

#### References

1. J. B. Dennis, "A Multiuser Computation Facility for Education and Research," *Communs. ACM*, Vol. 7, No. 9 (September 1964)
2. J. B. Dennis and E. C. Van Horn, "Programming Semantics for Multi-Programmed Computations," Technical Report MAC-TR-23, Project MAC, Massachusetts Institute of Technology, December 1965 (to be published in *Communs. ACM*, March 1966).

