

Workshop on basic design principles of operating systems

Chairman: Dr I.C. PYLE
AERE Harwell, England

The discussion was concerned with operating systems, their definition and programming. The usefulness of the idea of virtual machines was agreed, although the concept itself is not clearly specified. On the issue of programming language, the question was asked whether operating systems must be written in languages different from those used for general-purpose real-time or process control; and if so what are the essential differences?

Operating systems and virtual machines

L. Is an operating system the 'glue' which holds together the facilities that a user has at his disposal? Or is the operating system that which is performed by programs rather than carried out by the system operator? Different people seem to use the phrase for anything between these extremes.

B. Looked at from the outside, e.g. a process control application, we need to estimate the stability of a computer system with respect to changes in

- a) The environment which may affect the operating system such as new kinds of devices.
- b) The application programs to be executed, such as new control algorithms.

Regarding the hardware plus operating system as a virtual machine helps us to do this.

P. There are three different aspects of an operating system, which one should clearly distinguish in discussing the levels of a virtual machine:

- a) The facilities through which the user can use the hardware of the computer system more easily.
- b) The facilities through which the user can control interactions of his own tasks.
- c) The facilities through which the operating system can control the user program to share resources between several user programs.

B. An ESONE Working Group is engaged in trying

to define an interface between application programs and operating systems for use with CAMAC. This is guided by the concept of a virtual CAMAC controller and the virtual machine concept is found valuable in the work.

E. Is a real-time system only a small-scale batch system?

L. No, because the programs do not cooperate.

M. A language defines a machine. The definition of any programming language must imply a virtual machine, which can run programs written in the language.

Programming languages

M. The operating system is an extension of the hardware and has to be able to interact with it in very detailed ways which may not be appropriate for application programs. It should therefore be programmed in a different way.

J. The time factor is an important difference between the language chosen for operating systems and application programs. The operating system is called frequently in comparison with applications programs and therefore must be written in very efficient language.

X. This is not uniformly true for operating systems: parts are used with different frequencies.

L. Not only do we need to consider the basic operating system primitives, we need to realise that the important structure in a real-time system is a process rather than the program. We need programming languages in which the idea of a process is clearly expressed.

P. It is important to distinguish between a 'real-time language' which incorporates timing facilities, and a 'language for real time' which is suitable for use in real-time programming while not having any specifically real-time features. CORAL 66 is

one of the latter.

H. An application package should look like a virtual machine when it is finished, therefore you should have a common language with which to

construct them. A language for operating systems should have more privileged features so that the application programmer has a subset of the same language.