4 The System Nucleus

In the last chapter we developed the concepts and tools needed to build the paper operating system outlined in chapter 1. As mentioned there, the paper operating system will resemble an onion in which each layer provides a set of functions dependent only on the layers within it. At the centre of the onion are the facilities provided by the hardware of the machine itself. The onion layers can be regarded as implementing successive virtual machines, so that the onion as a whole implements the virtual machine required by the user.

The major interface between the basic machine hardware and the operating system is provided by the system nucleus, which is the innermost layer of the onion. The purpose of the nucleus is to provide an environment in which processes can exist; this implies handling interrupts, switching processors between processes, and implementing mechanisms for inter-process communication. Before describing these functions in detail we shall look at the essential hardware required to support the operating system we are trying to build.

4.1 ESSENTIAL HARDWARE FACILITIES

(1) Interrupt Mechanism

It was mentioned in chapter 2 that in order for I/O activities to be overlapped with central processing it must be possible to interrupt the running process when a peripheral transfer is complete. We therefore demand that our computer should provide an interrupt mechanism which at least saves the value of the program counter for the interrupted process and transfers control to a fixed location in memory. This location will be used as the start of a piece of program known as an interrupt routine, or interrupt handler, whose purpose is to determine the source of the interrupt and respond to it in an appropriate manner. We will describe the interrupt handler in section 4.4, and will discuss the various forms it can take according to the precise nature of the interrupt mechanism available.
(For the sake of completeness we mention that some computers, for example, the CDC CYBER, operate without an explicit interrupt mechanism. In such a computer one or more processors must be dedicated to monitoring the status of the I/O devices to detect when transfers are complete. In the case of the CYBER this function is performed by a number of so-called 'peripheral processors', while the CPU is relieved of all I/O handling. A restricted form of interrupt still exists, however, since the peripheral processors can force the CPU to jump to a different location.)

(2) Memory Protection

When several processes are running concurrently it is necessary to protect the memory used by one process from unauthorised access by another. The protection mechanisms which must be built into the memory addressing hardware are described in detail in the next chapter; for the present we take their existence for granted.

(3) Privileged Instruction Set

In order that concurrent processes cannot interfere with each other, part of the instruction set of the computer must be reserved for use by the operating system only. These privileged instructions perform such functions as

(a) enabling and disabling interrupts
(b) switching a processor between processes
(c) accessing registers used by the memory protection hardware
(d) performing input or output
(e) halting a central processor

To distinguish between times when privileged instructions are or are not allowed, most computers operate in one of two modes, usually known as supervisor mode and user mode. Privileged instructions can be used only in supervisor mode. The switch from user to supervisor mode is made automatically in any of the following circumstances.

(a) A user process calls on the operating system to execute some function requiring the use of a privileged instruction. Such a call is termed an extracode, or supervisor call.
(b) An interrupt occurs.
(c) An error condition occurs in a user process. The condition can be treated as an 'internal interrupt', and handled in the first instance by an interrupt routine.
(d) An attempt is made to execute a privileged instruction while in user mode. The attempt can be regarded as a particular kind of error, and handled as in (c) above.