

MANAGERS ARE PEOPLE, TOO

An inevitable side-effect of having two or more people using the same system independently is the growth of certain activities which we can conveniently group together under the *management* heading. (We've included "independently" in that statement because it's obviously possible to get along without explicit management if the people can agree between themselves on some discipline to run the system, and in simple cases that might be practicable – but even then if the system is at all complicated it might be necessary to carry out some management functions such as partitioning discs or setting parameters governing access to common resources.)

Our notion of management for this purpose (we don't dare to claim that it amounts to a definition !) fits in well with the dustbin model : we include all the organisational tasks which have to be done to keep the system going, but which are not the responsibility of the people for whom the system is doing its primary job of getting work done. Here are some examples of management responsibilities, each of which can be significantly assisted by special software provided in the operating system.

Managing the system : To maintain and develop a computer system (which includes both hardware and software) it is necessary to keep track of the system's current state (to detect poor performance or breakdowns) and patterns of usage (to identify peak loads, long-term trends which might be useful guides to future development plans, etc. - see *CONFIGURING THE COMPUTER SYSTEM*). Generally, there should be facilities for inspecting and recording the state of any system component.

Software for intervention is also necessary. Starting and stopping the system are obvious requirements; means of monitoring and controlling the progress of individual processes might be provided, and might be useful in emergency tasks such as stopping rogue processes or resolving deadlocks.

Managing the people : We have already mentioned the user database; we need software to maintain and monitor these data. As new people come to use the system or people move away, the database must be changed. Such operations begin as manual tasks, but must be reflected in the state of the user database if they are to become effective; appropriate management software can make this task straightforward and (just as important) reliable.

As well as coping with changes to the population, the current users must be served; there must be software with which they can inspect their accounts and the states of their parts of the system, and means of setting such system parameters as are accessible to them.

Managing money : The accounting system is an important part of most commercial computer systems. It must be able to keep track of costs incurred at all levels in the system, and to bring these together under the appropriate headings so that accounts may be made up and properly verified. Audit trails must be kept in case of disputes.

If that sounds easy, it's deceptive. Consider what must be done to charge for processing : each microsecond of processor operation must be recorded, the system must know whose process is responsible for the activity, and the record must find its way back to the correct account in the user database. Depending on the accounting system used, the same might be required for each byte of memory, and each microsecond of use of communication channels. And so on. Effective accounting software must spread its tentacles throughout the whole of the system, and it can determine whether a system management algorithm is or is not usable.

Managing exceptions : "Exceptions" is a polite word for emergencies – there's an exception if something unexpected has happened, which usually means that something has gone wrong. (The term also includes events such as interrupts, which are slightly surprising but basically under control.) An exception might not

be fatal to the operation of the system, but typically there is some damage to be cleared up somewhere.

After a really unexpected exception, the important aim is to minimise the subsequent trauma to system and to people. If it's possible to keep the system going, it might be worth trying to find out what went wrong. Typically, some sort of system state analysis might be attempted, but in many cases not much can be done. More commonly, management procedures for handling serious faults are concentrated on saving such information as might be useful in restarting the system with as little loss of work as possible; we shall say more about such procedures in the next section.

QUESTIONS.

Consider our remarks on charging for processor use. How would you implement it ?
