RELIABLE SYSTEMS

To the best of our knowledge, no manufacturer explicitly offers unreliable computers for sale, but over the past ten to fifteen years some have advertised, and sold, computers which they claim to be "reliable". This is an exceedingly unusual – perhaps unique – example of understatement in computer advertising, for what they mean is "*completely* reliable". More specifically, they mean that the computers in question will not be affected by what we have called internal accidents; they will always be available for use, so that they can support a full-time, uninterrupted, demand for computer services.

Such continuous service has become a requirement of many organisations as their dependence on computer systems for data processing, communications, and coordination has extended from local use to a world-wide scale. The reliable service has been achieved in the hardware by means such as the incorporation of error-checking mechanisms in the circuitry and the provision of redundant machinery at all levels, so that in case of a fault developing in any component it is always possible to provide a working replacement instantly.

The idea has been around for a long time. In 1967, Algirdas Avizienis published an article on the design of "fault-tolerant computers", and he has recently reviewed[REQ20] the development since then. He states that there is no reason why modern microprocessors should not be contructed with very high levels of reliability, but that the Pentium Pro processor, despite apparently having the best provision for reliability among comparable chips, "falls short of the checking provided in the 1960s by the IBM System/360 and its contemporaries".

The reliable processor's emergency action is automatic, and does not affect any software, but operating system cooperation might be necessary in cases where some hardware devices become unusable, and stored data are concerned. The most obvious example of operating system involvement is in the disc store, where removing a device necessarily affects the pattern of data storage, and replacing it after repair usually requires significant reorganisation of data to restore the required level of redundancy in stored data.

Reliability, in this sense, need not extend over the complete system to be useful. If the integrity of your disc store is more important than completely uninterrupted service, it is reasonable to provide redundant disc storage without necessarily going to the expense of redundant processors and other system components. In systems of this sort, all files are stored several times on different physical storage devices, so that destruction of any device can never destroy all copies of any file. In effect, the system has a permanent backup copy which is always completely up to date. Various manufacturers provide for such safe disc storage under names such as shadowing, imaging, or mirroring.

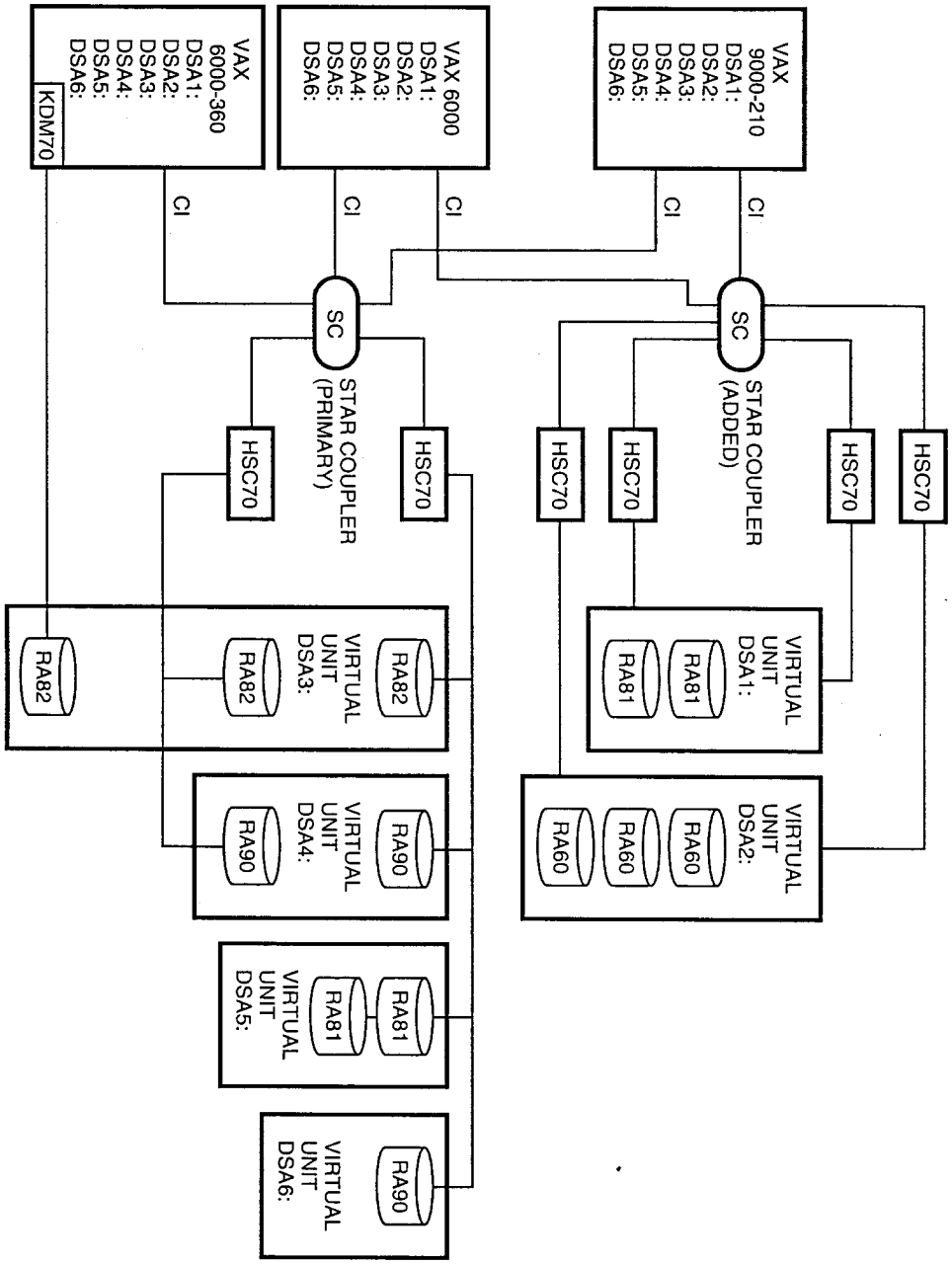
The diagram shows an example of such a system[REQ10], in which redundancy is incorporated at many levels. The details are not very important for our purposes (if you're interested, look up the reference), but notice that three processors (on the left) are connected through two levels of device management hardware to six "virtual units". Each of the virtual units is composed of one or (usually) more real disc units, all of which contain the same data.

To cope with such backing store organisation, the operating system must know on which hardware units copies of each file must be stored, and which units are at present active, and, when a unit comes back into service after maintenance or repair, the operating system must return it to its proper state by copying files as required from other units.

REFERENCES.

REQ10 : S.H. Davis : "Design of VMS volume shadowing phase II – host-based shadowing", *Digital Technical Journal* **3#3**, 7 (Summer, 1991)

REQ20 : A. Avizienis : "Towards systematic design of fault-tolerant systems", *IEEE Computer* **30#4**, 51-58 (April, 1997).



QUESTIONS.

Consider the reliable system in the diagram. Can you find ways in which each of the computers can store files so that they are safe from any single machine failure ? - or from any two simultaneous machine failures ? How can the operating system find out about any such failure, and how it should react ?
