FAILURE RECOVERY

Dr. R. M. Needham

Mathematical Laboratory,
University of Cambridge,
Corn Exchange Street,
Cambridge, CB2 3QG.

Abstract:

The problems of maintaining the integrity of a multi-access filing system and the teaching of this subject at postgraduate level are discussed.

Rapporteurs:

Mr. M. J. Elphick
Mr. L. Waller

The teaching at Cambridge in this area is at postgraduate level only. The aim is to give the student a feeling for the range of possible contingencies he should consider and plan for, and to set some standards for the performance a system ought to provide. The fact that the Cambridge system suffers from some hardware reliability problems is something of an advantage here!

Dr. Needham dealt solely with the integrity of the filing system, as the problems of processor reliability tend to be of parochial interest only. The general considerations should apply to any filing system in which the unit of information stored is typically smaller than the physical disk size. The Cambridge system uses a large fixed disk, and currently holds files for some 300 users, with a total of about 8000 distinct files (of which 50% do not exceed 4K characters in size). Their content is variable, but irrelevant to the integrity problem. Users may create, modify, manipulate and delete files from on-line terminals, and the result is a high activity of file usage.

The need is for an 'integrity and recovery system' which the users will trust implicitly; without this confidence, users are tempted to 'do it themselves', e.g. by maintaining their own back-up copies of files on tape. The system should also be capable of adjustment in response to variations in system parameters, e.g. the rate of disk failures.

The standard solution is to copy new and updated files to magnetic tape at the earliest opportunity. However, since a file will often be updated several times in a console session, this would lead to an excessive outflow of information. In practice, a special job is run automatically at a specified time interval to dump copies of new or updated files onto magnetic tape. This interval, which was 20 minutes a year ago, is currently three hours (and could probably be increased still further); it is, however, shorter than physical requirements dictate, in order to increase the users' confidence. The daily outflow of information from the disk to tape storage is about eight million characters --- it is only as low as this as a result of the requirement that the user must explicitly specify that he wants his file preserved. If he does not do this, then no copy is taken and the file may be lost in the event of disk failure.

The next problem is how to organise the information dumped on tape so as to make the recovery process as straightforward and cheap as possible. The file system has a three-level structure of Master Directory, user directories, and user files. Files are controlled and accessed by the use

of the appropriate user directory, which is in turn accessed through the Master Directory; hence, loss or corruption of information can occur at any of these three levels:

1. The Master Directory may be lost. Recovery from this is a major operation, requiring from two to three hours to complete; happily this situation occurs less than once a year. It is reasonable to expect that the system itself should detect any errors at this level, and it would also seem reasonable to allow the system to initiate the recovery process, should the fault be discovered by the system. In practice, the latter proved to be too drastic in the case of loss of the master directory. It is usually worthwhile for a systems programmer to attempt to 'fiddle his way out' for a few minutes at least.

2. A user directory may be damaged. Again, the system can detect this, as may the particular user affected. To check all user directories takes about two minutes, and is done on every 'cold' or 'warm' start, as this can include partial checking of the actual files (e.g. that the disk areas pointed to by entries in the user directory do appear to contain valid file records, and that file sizes are correct).

3. A user may detect damage to one of his files, and the system should let him initiate the recovery process from the terminal. Experience has taught the need to build flexibility into the system, and avoid the need to reconstruct the whole file structure after every failure. Thus, the loss and subsequent restoration of a user directory should not affect other users, and damage to one of a user's files should leave him able to continue useful work with his remaining files. However, not all users are equal; and recovery from any damage to the directory and files 'owned' by the system library must clearly take priority.

The speaker went on to discuss the process of recovery. There is a need for a quick minor recovery operation, while major failures (which occur much less frequently) can be left to a more lengthy process. A compromise must be made between the effort expended on the organisation of the information stored on tape, and the amount of tape searching required if errors do occur. Repeated sorting of the tape copies may place too heavy a load on

the system, and affect the general performance.

The Cambridge system maintains a number of magnetic tapes, which are used cyclically. After filling one such tape, copies of all directories, and those files of which no copy exists (or whose only copy is about to be overwritten) are written to the next in the cycle and, as a result, the contents of this 'Dump System' grow. Although some information becomes obsolete, a vast amount of semi-permanent information would circulate over a long period were no further action taken. One solution would be to increase vastly the number of tapes in the cycle, and hence increase the amount of tape to be scanned during recovery. An alternative approach, adopted at Cambridge, is to maintain a number of secondary 'Archive' tapes, which are grouped into eight separate cycles with four tapes per cycle (each cycle being associated with a group of users). The Archive System operates similarly to the primary Dump cycle; however a complete cycle of the former takes four weeks. The net result (as the user sees it) is that within three hours of the creation of a file which requires preservation, a copy will reach the primary Dump tapes and will then move to the Archive cycle allocated to the user after a week, at which time it will be dropped from the Dump cycle. From there it may be retrieved in about 90 minutes indefinitely or for a period of up to one month even if the user deletes it from his directory. Retrieval may be initiated by the user by means of a recovery command, which changes the file status recorded in the User Directory. A special job runs every hour or so, looking for such file-retrieval requests. Checking and retrieval is done by reference to directories and disk maps. Should the user attempt to use a file requiring retrieval, the system will inform him of this and refuse to proceed further with that job. This is an important point --- some operating system have been designed which merely 'hang up' in this situation, awaiting the recovery of the file.

It is psychologically desirable to take greater care of users' files than they would themselves, and accordingly the primary dump tapes are duplicated. At one time so were the archive tapes, but this proved too much of a good thing! At present, around 60 tapes are in use. The system is entirely automatic in operation, unless a catastrophe of unusual dimensions occurs, and has proved capable of dealing with a large number of mixed failures.

[A more extensive description of the Cambridge system may be found in (Fraser, 1969)].

Turning to the teaching of this topic, the speaker remarked that there was not much literature on the subject. To ask students to read the coding would be 'a perversion of all we are trying to achieve', but a number of important points could be got across. It is possible to derive 'quasi-equations', e.g. relating the number of tapes used to the degree of security ensured and to the expected amount of tape searching required for recovery. However, it can take a long time to give the necessary background knowledge of the system, and it is fortunate that at Cambridge the teaching of this topic can be deferred until students have had experience of using the system. It is still difficult to give more than a scattered and partial understanding of the problems, but they can be encouraged to expect such systems to work in a smooth and consistent way.

It is important to instil into these future software writers the right attitude towards users of the systems they will construct. They will not have the opportunity of thinking this out while working in a manufacturer's team, where the pressure to improve programs will only come as a delayed feedback from the eventual users. They must be encouraged to write programs which are robust, well thought out and which cater for all likely contingencies. This is done at Cambridge by getting students to add small command programs to the system (this can be done quite easily), which must cope with emergencies such as pulling out the console plug.

In conclusion, Dr. Needham confessed that he didn't see how to convey a general understanding of failure recovery in a wider context without 'dragging students through the mire of some real system'. One hoped to find some reasonably instructive mire through which to drag them.

DISCUSSION

In the short discussion which followed, Professor Coffman said that he felt that the issue of giving students background experience with a working system was one of degree; some motivation was necessary, but this could be carried too far, without corresponding benefit. Dr. Needham agreed that one had to be careful here - there were whole areas of the Cambridge multiple-access system which were completely unrewarding to study (and were not covered in this course). Asked by Professor Page if he could estimate the amount of effort required from students in absorbing background material before teaching could begin, Dr. Needham remarked that one would not ask students to memorise this material --- it was

essentially non-examinable. One merely wanted them to get some feel for what the problems are.

REFERENCES

Fraser, A.G. (1969): 'Integrity of a mass storage filing system', Computer Journal, Vol. 12, pp. 1-5.