Programming Staff Note 32 Computation Center

September 15, 1964

[illegible]

FROM: M.J. Bailey and R.C. Daley

SUBJ: A CTSS Secondary Storage Back-up and File Retrieval Scheme

This scheme, unlike earlier suggestions, is based on the following considerations and assumptions:

(a) That the processing of incremental dump tapes on the 1401 for the 7094 system is undesirable and would in any case be very restricted in scope due to the size and nature of the 1401.

(b) That the post-processing of incremental dump tapes on the main computer system (e.g. as background in the current system) would take up very little less of the total system time than would the periodic complete dumping of all system and (current) user files in secondary storage.

(c) That a complete dump of current secondary storage files could be done during time-sharing, without undesirable restrictions on the use of a user's files while it is taking place.

Throughout this paper the word "week" will mean a period of M 24-hour days (where M is probably 7 or greater), and the word "day" will mean a period of N hours (where N probably is either 24 or some factor of 24).

1.1 Back-up Operations Required

(i) For each "day", form a New File Tape (NFT) of one or more reels, in duplicate, automatically while CTSS is operating — in general dumping at some time after a user has logged out.

An NFT will contain a copy of the latest version of all files newly modified or created within that day. It might in some cases also contain redundant earlier versions of such files, due to a user's files having been dumped while he was logged in, an event which will be necessary prior to an automatic logout.

++ [illegible]

An NFT will always be terminated automatically, by the system, at the end of each N-hour day and at automatic logout time. In both cases this would involve first ensuring that all new files for the day or partial day, respectively, have been dumped; rewinding the last reel; and then suspending dumping until the tape operator has mounted tapes for the next day. Each reel should be labelled with two times defining the period during which the contained files were created.

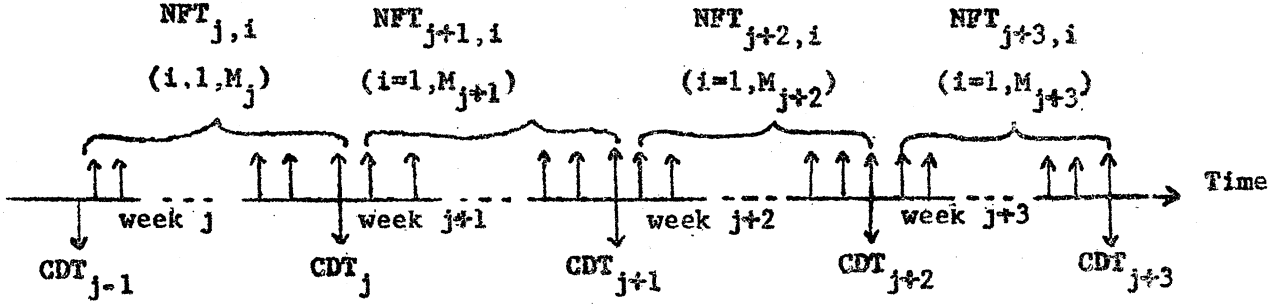
(ii) At the beginning of first day in each M-day week make a CDT -- a complete dump of current system and user files. This would also be done automatically while CTSS is in operation (in duplicate) without impairing the availability of a user's files. It would be done before the initiation of the first NFT of the week.

'Current' user files would be those <u>used</u> at least once during the last p weeks, where p is probably 3 or 4. That is to say, files liable to deletion by the administrator due to p consecutive weeks non-usage at this time or soon after, will not be copied on to the complete dump tape.

The material on the CDT would be ordered as follows: first, the system files, next those user files used at least once during the last i days, for all users, followed by those last used during the next most recent i days, and so on until all files used in the last (pxm) days are dumped. A printed summary of the contents of each reel would be automatically provided by the dump program.

1.2 Reload and Retrieval Tape Situation on a Given Day.

The following diagram shows the creation of CDT's and NFT's for the 4 week period week j through week (j+3):



(M_j, M_{j+1}, etc. are the numbers of NFT created each week and each is ≥ M. They are > M when automatic logouts have occured).

There will exist at any time during the Ith day of the Jth week the following tapes:

<u>NFT</u>: (a) An NFT, in duplicate, for each automatic logout so far on this day, if any.

(b) An NFT, in duplicate, for each of the previous (I-1) days in week J, and for each automatic logout made on these days.

(c) At least one NFT (single copy) for each day of the q weeks before week J. (q = 6?).

[Tapes in items (a) and (b) would be used for a reload in week J, while those in item (c) would be used for retrieval by date of creation.]

<u>CDT</u>: (d) A complete dump tape CDT_{J-1}, in duplicate, containing a copy of the state of the secondary storage at the beginning of week J.

except for those files unused since week (J-4) (if p, as given in 1.1(ii) above is 3). This tape would be used for a reload in week J, and for retrieval.

(e) CDT_{J-2}, CDT_{J-3}, etc. for each previous week for some years past, for general back-up and retrieval purposes. These tapes could be kept in duplicate for some most recent portion of the time past, but in single copies for the rest.

<u>Note</u>: If OF_x denotes the files last used in week x, CDT_x the files contained on CDT_x, and $NFT_{x,y}$ the files contained on $NFT_{x,y}$ then the following equation holds:

$$CDT_k = CDT_{k-1} + \sum_{i=1}^{M_k} NFT_{k,i} - OF_{k-p}$$

where M_k is the number of NFT created in week k and p is as stated in 1.1 (ii) above.

1.3 Procedure for a Complete Re-loading after Disaster in Day I of Week J.

(i) Reload onto the secondary storage all NFT written thus far in week J(i.e. items (a) and (b) in 1.2 above). If an NFT <u>and</u> its duplicate is unreadable, some files written this week would inevitably be lost.

(ii) Reload CDT_{J-1}, starting up CTSS after all the system files are in, then continue the reloading of users' secondary storage files simultaneous with the operation of CTSS. Due to the order of files on tape (see 1.1 (ii) above) each user's files will come back to him in groups, with the most recently used group first, then some time later the next most recently used group, and so on.

Files would not, in general, be reloaded into the same physical locations as they previously occupied .

As users start re-using the system their new files would not be dumped until all reloading was completed.

If CDT_{J-1} <u>and</u> its duplicate were unreadable, either <u>CDT_{J-2}</u> plus the $NFT_{J-1,i}$ $(i=1, M_{J-1})$ may be loaded instead, <u>or</u> offending portions of CDT_{J-1} ignored and the constituent files retrieved <u>at</u> their user's request after reloading was complete.

1.4 Retrieval

With reference to the diagram given in 1.2 above, the following statements summarize the file retrieval situation:

(a) Files last created or modified in week j can be found on CDT_j, possibly CDT$_{j+1}$, and possibly also CDT$_{j+2}$ (if p is 3 as described in 1.1 (ii)), i.e. unless they were deleted in some way during week j, (j + 1) or (j + 2).

(b) Files last created or modified in week j, day i, can also be found on NFT$_i$ of week j, if it is still being retained (see 1.2 (c) above).

(c) Files deleted due to non-use since week j can be found on CDT$_j$, CDT$_{j+1}$, and CDT$_{j+2}$ (if p = 3),or found according to time and date of creation/last modification as in (a) or (b).

(d) Files destroyed or deleted in week j + 1 can be found on CDT$_j$ if they were last made in week j or earlier, or otherwise may be found as in (a) or (b).

Since each reel of the CDT's and NFT's being saved would be accompanied by a written indication or summary of its contents according to the time of the last use or creation of its constituent files, the operator can always select the appropriate reel if the user provides him with the necessary information. In such a case only one reel need be searched for the file to be retrieved.

1. Data-Cells

It can easily be seen that the use of data-cells such as those used on the IBM 2321 Data Cell Drive instead of tapes would not, in any way, invalidate the scheme described here. Moreover, certain advantages would accrue from improvements which could be made due to the random accessibility of data-cells. A scheme which took full advantage of the data-cells might differ from the above in the points given below, assuming data-cells with the specifications of the IBM 2321 (Model 1). (It should perhaps be noted that the IBM 2321 transmission rate is approximately twice as slow, and its capacity twice as large as that of 2400 feet of 800 c.p.i. 512-word/record tape on an IBM 729 VI tape drive).

(a) A set of New Files for a given day (NFT, above) would occupy less space because the NFT would now contain the latest version only of each new file, since earlier versions made that day would be overwritten.

(b) The termination of a set of New Files at the end of a day would not often involve remounting a new data-cell (only when all strips were full). The new files for the next "day" should perhaps begin on the next higher addressed strip. (1 cell = 200 strips; 1 strip = 33,000 words).

(c) Complete Dumps (CDT, above) need no longer necessarily be ordered in a special way, except when the number of data-cells needed for a complete dump exceeds the number available on all drives (1 drive has 10 data cells). See item (e) below.

(d) The data-cell equivalents of the NFT and the CDT should contain, in the locations of lowest address, say, a directory of contents for each of the one or more cells making up the CDT or NFT.

(e) Reloading, given the acceptance of (a) through (d), above, would then proceed roughly as follows:

1. Reload the system files from the appropriate data cell; then start up CTSS.

2. If n data-cells are available over all drives, and m cells were needed for a complete dump, then if m > n, load the most recently used files from (m-n) particular cells.

3. Mount the remaining n (or less) data cells containing user files to be reloaded. Begin reloading from them (perhaps slowly).

4. The first time a user requests a file which is not yet back on disk but is on one of the mounted data-cells, transfer it onto the disk from the data cell.

(f) Retrieval of any kind would merely involve a scan through a data cells' directory to find the address of a file, once the operator has the correct cell on the machine. No file searching would be needed.

3. <u>Dumping and Editing</u>

Dumping of new files onto the NFT should be possible in one of two modes in order to balance the two objectives of (a) dumping as little as possible by waiting until users have logged out, and (b) avoiding lengthy automatic logouts.

"Complete" dumping, New File dumping, and disk reloading, should be given a guaranteed percentage of all CTSS time in order for the back-up scheme to work properly, and this percentage should be a variable parameter of the system. (These three system service routines would never run simultaneously with each other). why not dumping & reloading?

Other than these two points, since detailed considerations of the actual method of dumping need not effect an overall back-up and retrievel scheme, the method of dumping will not be discussed here and will henceforth be treated as a separate subject where possible.

This same comment applies also to disk editing (from the console or otherwise), administration of the secondary storage, and the details of file retrieval.