5. RANDOM FILES

5.1 Description

It has already been stated that a 'random' file comprises records that can be required at random – there is no way of predetermining the next record to be accessed from the last record accessed. The system needs some kind of reference to each record so that the record can be found and read when the user requires it. For this purpose, a 'logical record number' is used that identifies the record's position relative to the beginning of the file. This implies that the file is normally created as a *sequential file with sequence determined* by the value of one item in the record. The last few digits of an account number, for example, could be used as a logical record number as long as there is a direct relationship between the value of that part of the field and the position of the record relative to the beginning of the file (account number 0132, must be the 132nd record in the file).
An example sequence of events could be as follows:

1. A file is created using the CRF utility (see manual M08, The Utilities Reference Manual or the TOSSUT utility in M11 DOS6800 Reference Manual) to contain records about accounts.
2. Account records are written to the file *sequentially* in account-number order.
3. When the system is on-line the keyboard operator inputs an account number.
4. The program decodes the account number to produce a 'logical record number'.
5. The 'random read' instruction is executed and the Data Management routine takes the logical record number and calculates the physical position of the record on the disk. (It knows the address of the first sector in the file and the blocking factor so it can find the relevant sector by dividing the logical record number by the blocking factor).
6. The record is read directly (and placed under exclusive access) and made available to the task.

Note that when the file is assigned, the CRN is always set to zero. After subsequent accesses, the CRN is set to the last record that was accessed.

It is possible that records do not contain an item that is increasing by one for each record in the file. This does not matter as long as there is a direct numerical relationship between the item and the record's position in the file.
If a file is required that contains no numerical relationship between records then the programmer has the choice of inventing a new field in the record that can contain a logical record number, or better, to use the *indexed* random method described in chapter 6.

5.2 Creating a Random File

The file can only be created on a TOSS formatted disk, i.e. one that has been created by the utility Create Volume (CRV). The creation of the file must be performed in two stages, firstly, the file space is set up by the utility program Create File (CRF); see the Utilities Reference Manual, M08, or the TOSSUT utility in M11 DOS6800 Reference Manual for a detailed description, then the actual records must be written to the file.

1st Stage

The sequence of operations for CRF is as follows:

1. Call CRF utility under the TOSS utilities Monitor, or as a subroutine if required by the application, or via the TOSSUT utility under DOS
2. CRF requests a number of parameters. Most parameters can be given as required, but for a random file, two parameters are obligatoiry. To 'File Organisation' give 'S', and to 'Number of index files' give '0'.
3.. CRF searches the volume(s) for free extents large enough to hold the stated file size.
4. The file is created with the required number of records, all of which contain space characters.
   Each record is set to 'FREE' status. The LRN is set to zero for this file.

2nd Stage

It is possible to write records with random write instructions at this point but on a subsequent read instruction the LRN will still be zero and an error will be returned stating End-of-File. The actual records must be written to the file using the sequential write instruction so that the LRN will then be set to the last used record in the file. The operations for this second stage depend upon the source of the records. If the records already exist, for example a bank putting its account records onto the computer, then the records could be copied into the 'empty' file on the disk. If it is a new system, then the account numbers could be written to predetermined positions on each record. Each account record could then have the rest of the information set up at the time account numbers are allocated to customers.

Remember when setting up the records that the logical record number will be related to the record's position on the file. The records must be written, therefore, in the order determined by the relationship between the record key to be used and the logical record number. (Account Numbers allocated as 1, 2, 3, 4, etc. should be set up as the 1st, 2nd, 3rd, 4th, etc. records in the ifle. If account number are allocated by tens, i.e. 10, 20, 30, 40 etc. these will still be logical record numbers 1, 2, 3, 4 etc. so provision must be made in the using task to reduce the input key to the relevant logical record number.)

The file can now be used as a random file, see the instructions in the following paragraphs, or as a sequential file, see Chapter 2. If another task is going to use this file as an indexed random file, then this file should be created according to the instructions in Chapter 4. After set up, the file can be used for random, as well as indexed random, processes.

5.3 Instructions

A definitive description of these instructions is contained in the relavant language reference manuals, either Assembler Programmer's Reference Manual, M06, or CREDIT Programmer's Reference Manual, M04.
These instructions are, briefly:

ASSIGNING THE FILE

When the file is 'assigned', the filename is linked to a file code declared in the same task. If the file is to be used only by that task it must be assigned with TC = 1. If the file is to be accessible by more than one task, it must be assigned as a common file, with TC = 0.
After successful assignment, the file is available to the assigning task for random read, random write and random delete, the record being accessed by its logical record number, see section 3.2.

RANDOM READ

This instruction will allow the task to access the record by its logical record number, and to put the record under exclusive access. The CRN will be updated to the last record that was accessed.

RANDOM WRITE

The record to be written is specified by its logical record number and the CRN is updated to this record after a successful write. If the record on the file is 'free' it will be set to 'used' but if it is already 'used' it can only be written if the record is under exclusive access. This implies that if an item in a record is being changed, the record must be read by random read first. Exclusive access is released after a random write.

RANDOM DELETE

This instruction can only be used on a record, specified by its logical record number, that is under exclusive access. It sets the status character to'free' then releases exclusive access. Note that in CREDIT, this instruction is effected through the Data Set Control 1 statement (DSC1).

CLOSING THE FILE

A 'close file' instruction is used to indicate that the file is no longer required by that task. The LRN is updated and saved on the volume. The close action applies only to the task that issues it, and the file is still available to other tasks that are using it at that time.