

Programming Staff Note 61

From: R. C. Daley, S. Dunten, C. Garman
Date: January, 1966
SUBJ: File System Modules and Internal Calling Sequences.

Introduction

This memo is an attempt to document calls made between modules of the CTSS 7094 file system. User calls to the file system are described elsewhere: I/O operations and administrative and privileged calls are described in the CTSS manual (2nd Edition) and memo CC-241-2; supervisory and control calls are described in Programming Staff Note 47.

The file system is currently composed of 14 BSS modules: 13 FAP assemblies and one ALGOL (AED) compilation, forming seven functional sections of the file system, (see Fig. 1):

1. File coordinator (FCOR); interpret all calls to file system, and check for validity, restrictions, memory protection violations, etc.
2. Search module (SEAR, SRCH, STIK, PUSH); file directory search and maintenance.
3. Buffer control module (BUCK); determine sequence of operations needed to perform and check I/O transmissions.
4. I/O control module (I/OCM); coordinate I/O synchronization requests; start/stop I/O for any or all users.
5. Queue manipulation module (QMAN) (ALGOL); maintain I/O queues for individual devices.
6. Strategy modules (DDST, TPSM); initiate single-record I/O transmissions, service I/O queues and priorities.
7. I/O adapters (DDAP, TPAP); software interface between strategy modules and data-channels; perform single record I/O operations; service data-channel traps; I/O error checking and corrections.

One BSS module is not mentioned above: TMAN, track management module, which maintains the track usage tables for the disk/drum strategy module, (see Fig. 2).

File System Conventions

The file system operates in 7-index register mode throughout; in calls between file system modules, the contents of index registers 1, 2, and 3 must be preserved, index 4 is used for subroutine linkage, and the contents of index registers 5, 6, and 7 should be assumed to be destroyed.

In internal calls within the file system, the operation EFA (assembles as MOP) is used to indicate that address and tag of this parameter should be used together to form the effective address of the desired parameter. A standard subroutine (GETEFA) is provided to compute the effective address if required. The operation PAR (assembles as TKH) is used to indicate that machine addresses may be found in both address and decrement of the word; the tag field is ignored, but should be 0.

For subroutine calls which include EFA parameters in the calling sequence, the specified index register may not be 5, 6, or 7; if the contents of registers 1, 2, or 3 has been changed before a call to GETEFA, its value must be restored if the GETEFA subroutine is to return the proper effective address.

In references to variable names of the form "VARf" the lower-case "f" stands for a device number with values between 1 and FMAX (FMAX-3 for current implementation), i. e., names VAR1, VAR2 and VAR3 may occur.

The following sections describe calls between modules of the file system; the data bases and record formats are explained in the appendix.

Section:	Calls described:
1	File coordinator to search module.
2	" " to I/O control.
3	" " and search, to IUCM.
4	FCOR to strategy modules.
5	FCOR, buffer control to strategy modules.
6	I/O control to strategy modules.
7	Strategy modules to IOCM (incl. QMAN).
8	Miscellaneous device-oriented calls.

Appendix Data bases used within the file system.

SECTION IFCOR To SEAR

When the file coordinator wishes to interrogate or modify entries in a file directory, it does so by calls to the search module.

SEARCH - used to search for a specific file name in a specific directory.

```
TSX SEARCH,4
EFA UFD
EFA FILNAM
EFA INTO
PAR NOFILE,,UFDERR.
```

UFD - is the first of two consecutive locations containing NAME1 NAME2 of the file directory to be searched.

FILNAM - is the first of two consecutive locations containing the file name desired in UFD.

INTO - is the origin of a 9-word vector into which the desired information is read: NAME1 NAME2 of the file directory, followed by the 7 words from the directory entry.

NOFILE - error return if file not found in specified directory.

UFDERR - I/O error searching UFD entries.

UPDFIL - used when file coordinator wishes to modify information contained in a file directory entry.

```
TSX UPDFIL,4
EFA UFD
EFA FILNAM
EFA FROM
PAR NOFILE,,UFDERR
```

(In this and subsequent calls, variables previously defined in SEARCH will be assumed unless otherwise specified.)

FROM - origin of 9-word vector of information to be re-written into UFD entry for FILNAM.

ADDFIL - used when file coordinator wishes to add new files to a given file directory.

```
TSX ADDFIL,4
EFA UFD
EFA FROM
PAR NOROOM,,UFDERR
```

FROM - 9 words containing file directory entry to be added to UFD.

NOROOM - no room in secondary storage to place file entry.

REMOVE - used when file coordinator wishes to remove an entry from a file directory.

```
TSX REMOVE,4
EFA UFD
EFA FILNAM
PAR NOFILE,,UFDERR
```

REWRIT - used to insure that all changes to a file directory have been properly recorded on secondary storage.

```
TSX REWRIT,4
EFA UFD
PAR NOFILE,,UFDERR
```

GETQU - obtain record quota and usage for a specific device from a specified file directory.

```
TSX GETQU,4
EFA DIR
EFA DEVICE
PAR QUMERR,,UFDERR
SLW QUOTA
```

DEVICE - number of device for which quota is desired.

QUMERR - machine or system error reading specified U.F.D. (or U.F.D. not found).

The allotment and usage for the device is returned in the accumulator in the form.

```
PZE ALLOTD,,USED
```

ALLOTD - is the record quota for this device.

USED - is the number of records currently used on this device.

SETQU - update the current allotments and usage for a device in a file directory.

```
CAL QUOTA
TSX SETQU,4
EFA UFD
EFA DEVICE
PAR QUMERR,,UFDERR
```

QUOTA - is a word of the form PZE ALLOTD,,USED as above.

SECTION IIFile Coordinator To I/O Control Module

The I/O control module serves as a traffic cop to coordinate user I/O requests with supervisor usage control requests, by permitting or forbidding removal of queued I/O from any or all user's queues, and checking for completion of I/O where necessary.

FCOR to IOCM

IOBEGN - allow I/O for blocked users to proceed, inform strategy modules that I/O may be attempted for user.

TSX IOBEGN,4
PAR DUSER

DUSER - is the location of an integer specifying which user's I/O is to be allowed to proceed. If DUSER=0, I/O for all users not otherwise blocked is allowed to proceed.

IOHALT - allow no more I/O for specified user to be initiated, wait for all pending I/O for user to complete.

TSX IOHALT,4
PAR DUSER

DUSER - is defined as for IOBEGN. If DUSER=0, I/O for all users is to be blocked, in preparation for a device requiring service un-interrupted by other devices' memory requests.

DRAIN - wait for all pending I/O in any queues to terminate.

TSX DRAIN,4

Return is made when all strategy modules have finished processing all queued requests.

IORST - restart file system I/O after machine catastrophe.

TSX IORST,4

Return after informing strategy modules that machine reset has occurred, and that pending traps may have been lost. (Called by IORSET)

SECTION IIIBUFFER CONTROL MODULE

In any call to OPEN, RDFILE, RDWAIT, WRFILE, WRWAIT, FCHECK, FWAIT or CLOSE, the file coordinator will only be responsible for determining the validity of the call. If the call proves to be valid, the file coordinator will pass the call to the buffer control module, along with a pointer to the necessary information in the Active File Status Table. The following calls form the interface between the file coordinator and the buffer control module.

1. BOPEN - opens a file for subsequent reading and/or writing (called by OPEN).

```
TSX BOPEN,4      open a file
EFA AFINTY      .. pointer to file entry
PAR PRIOR,,ERROR .. file I/O priority
```

AFINTY, defines the effective address which points to the active file entry in the Active File Status Table. PRIOR is the location of an integer from 0-7 which defines the I/O priority of this file for as long as it remains an active file. BOPEN will be responsible for initializing that portion of the Active File Status Table for which the buffer control module is responsible.

2. BASIGN - assigns a buffer to a file which has been previously opened for reading and/or writing (called by BUFFER).

```
TSX BASIGN,4
EFA AFINTY
PAR Y,,ERROR      C(Y)= PZE  BUFADR
```

BUFADR is the first location of the buffer to be used as necessary when reading or writing the specified file. BASIGN will only store the buffer address in the Active File Status Table and return.

3. BREAD - reads from a file which has been previously opened for reading (called by RDFILE and RDWAIT).

```
TSX BREAD,4
EFA AFINTY
PAR MEMORY,,BUFFER
PAR RELADR,,EOFRTN
PAR Y,,QWAIT      C(Y) = PZE  LOC,,NWORDS
PAR ERROR,,PVIOL
PAR NOBUF
```

MEMORY is the location of an integer which specifies which memory unit is to be read into (1=A, 2=B). BUFFER is the location of an integer specifying in which memory the buffer resides. RELADR is the location of an integer specifying the address within the file at which reading is to begin. If C(RELADR) is zero, reading will begin at the word following the last word read from this file. NWORDS is the number of words to be transmitted from the file beginning with the address LOC. If an attempt is made to read beyond the end of the file, control will be returned to the location EOFRTN. When this occurs, the number of words that will actually be read will be returned in the AC. If an error has occurred during a previous operation on this file, the read request will be ignored and control will be returned to location ERROR. If a previous I/O request involving this file has not been completed or the Strategy Module cannot completely accept the current I/O request, the request will be ignored and control will be returned to location QWAIT.

4. BWRITE - writes into a file which has been previously opened for writing (called by WRFILE and WRWAIT).

```

TSX BWRITE,4
EFA AFENTY
PAR MEMORY,,BUFFER
PAR RELADR,,EOFRTN
PAR Y,,QWAIT      C(Y)= PZE LOC,,NWORDS
PAR ERROR,,PVIOL
PAR NOBUF

```

RELADR is the location of an integer specifying the address within the file at which writing is to begin. If C(RELADR) is zero, writing will begin at the word following the last word written into the file. NWORDS will be transmitted to the file starting with location LOC. If an attempt is made to write through the end of the file, control will be returned to the location EOFRTN. At this time, the number of words which will actually be written (up to the end of file) will be returned in the AC. If a user wishes to append information to the end of a file, the write operation must begin at the address following the last word in the file.

After a normal return from BWRITE, the AC will contain the number of records, if any, that have been appended to the file.

5. BTRUNC - truncates a file which has been opened for writing (called by TRFILE).

```

TSX BTRUNC,4
EFA AFENTY
PAR MEMORY,,BUFFER
PAR RELADR,,EOFRTN
PAR ERROR,,QWAIT
PAR PVIOL,,NOBUF

```

The file will be truncated before the relative address specified by RELADR.

6. BCHECK - checks to see if the previous I/O operation on this file has been completed (called by FCHECK, FWAIT, RDWAIT and WRWAIT).

```
TSX BCHECK,4
EFA AFENTY
PAR MEMORY,,BUFFER
PAR ERROR,,FINISH
PAR PVIOL
```

If the previous I/O operation has been completed, the buffer control module will finish any related tasks and return to location FINISH. A normal return will indicate that the specified operation is still in progress.

7. BSAVE - finish any I/O operation on the specified file, in order to change buffers or insure integrity of a file (called by BUFFER, WRFILE, WRWAIT).

```
TSX BSAVE,4
EFA AFENTY
PAR MEMORY,,BUFFER
PAR ERROR,,QWAIT
PAR PVIOL
```

If at this time all related I/O operations have been completed, the buffer control module will complete any of its related tasks. If it is necessary to initiate any new I/O at this time, the I/O should be started and control returned to QWAIT. A normal return from BSAVE will indicate that all modifications to the file have been completed, so that the file may be considered complete on its device, and buffers re-assigned if necessary.

8. BCLOSE - finish any I/O operation on the specified file so that the file may be returned to inactive status (called by CLOSE).

```
TSX BCLOSE,4
EFA AFENTY
PAR MEMORY,,BUFFER
PAR ERROR,,QWAIT
PAR PVIOL
```

Usage is same as for BSAVE, except a normal return from BCLOSE will indicate that all modifications to the file have been completed and the file may be safely removed from active status.

In all of the above calls to the buffer control module the error return PVIOL is used whenever an I/O request is found to violate the memory bounds set for this user. NOBUF is used whenever the requested operation requires a buffer and no buffer has been provided.

The buffer control module will initiate and control all I/O operations by giving the appropriate calls to the specified strategy module. All possible error conditions should be checked before any new I/O is initiated. Whenever possible, reading and writing should be done directly in and out of the user's memory. The buffer should only be used when a partial record is involved. For example, assume the record size is 10 and the user wishes to read 53 words from the beginning of the file. The first 3 records will be read directly into the user's memory. The sixth record will be read into the buffer associated with that file. The remaining 3 words will be copied from the buffer when the user calls FCHECK or attempts to initiate new I/O involving this file. If the user now wishes to read the next 53 words from the file, the first 7 words may be copied directly from the buffer. The buffer control module will also provide a label for every record it reads, writes or rewrites. This label will consist of a word which contains the record sequence number in the address portion. This word (or label) will be recorded as the first word of every record in the file. In addition, the label of the last record in the file will contain in the decrement the number of words in this record.

SECTION IV

FCOR to SMF

- 1) SINITf - inform strategy modules that the file system is being initialized.

```
TSX SINITf,4
PAR ENABLE
```

ENABLE is a location containing the address of the common enable word for all I/O adapters. Strategy modules are to perform any necessary initialization, including calls to their adapters for proper trap referrals and enable location (called by IOINLT).

- 2) SCRAPf - inform strategy module that an entry in users active file status table is being reset.

```
TSX SCRAPf,4
EFA AFENTY
PAR QWAIT
```

AFENTY is the location of the user's active file entry under consideration. If the file is not complete for any reason, RCOUNT should be set to 0; if, however, the call cannot be accepted at this time, return is made to QWAIT.

CHAPTER VFCOR and BUCM to SMFStrategy Modules:

The following calls form the interface between the File Coordinator or the Buffer Control Module and the Strategy Module for the device "f".

1. OPENf initializes a file for subsequent reading and/or writing (called by BOPEN).

```
TSX  OPENf,4
EFA  AFENY
PAR  ERROR
```

2. QTESTf checks to see if there is room in the specified queue to process the specified number of requests (called by BREAD, BWRITE, BCLOSE, BTRUNC).

```
TSX  QTESTf,4
EFA  AFENY
PAR  REQCT,,FULRTN
PAR  QWAIT
```

REQCT is the location of an integer which specifies the number of requests needed, or if negative, whether a request for (partial) deletion of a file can be honored. If the queue specified by the file I/O priority cannot accept this number of requests, control will be returned to the location FULRTN.

3. READf reads from a file starting from a specific record in the file (called by BREAD).

```
TSX  READf,4
EFA  AFENY
PAR  LABEL,,IOLIST
PAR  QWAIT
```

LABEL is the location of a word containing the record number of the first record to be read. This LABEL must match the label of the record to be read. If successive records are to be read with a single call, the record labels must be in ascending order and sequenced by ones. IOLIST is the location of a list of I/O commands in the following form.

```
IOLIST ION      ,,N
        IOP      A,M,N
        IOD
```

ION (PON) is used to skip N words in the record and proceed to the next command in the list. IOP (PIW) reads or writes N words starting from the location A in the memory unit specified by M (1=memory A, 2=memory B). After completion, IOP will proceed to the next command in the list. IOD (PZE) is used to terminate the list.

4. REWRTf rewrites successive records in a file starting with the record specified by the address of LABEL (called by BWRITE, BCLOSE).

```
TSX  REWRTf,4
EFA  AFENTY
PAR  LABEL,,IOLIST
PAR  QWAIT
```

The record labels will be verified and incremented in the same manner as with READf.

5. WRITEf appends successive records to a file (called by BWRITE, BCLOSE).

```
TSX  WRITEf,4
EFA  AFENTY
PAR  LABEL,,IOLIST
PAR  QWAIT
```

The contents of LABEL will be recorded as the record label of the first record. Successive records labels will be sequenced by ones starting with C(LABEL)+1.

NOTE: When calling either WRITEf or REWRTf, the decrement of LABEL will specify the word count of the last record to be written. If the decrement of LABEL is zero, the Strategy Module will assume that this record will be followed by another and will provide for any necessary chaining.

6. DFILEf deletes successive records from a file starting with the record specified by the address of RECNUM (called by DELFIL, BTRUNC).

```
TSX  DFILEf,4
EFA  AFENTY
PAR  RECNUM,,QWAIT
```

If the Strategy Module cannot accept this call at the present time, control will be returned to location QWAIT. Delete requests will automatically be placed in the lowest priority queue. Once a delete request has been accepted by the Strategy Module, the corresponding entry in the Active File Status Table is no longer required.

7. CLOSEf deletes any pending I/O requests for this file from its queue and corrects the file pointer in the Active File Status Table if necessary (called by BCLOSE).

TSX CLOSEf,4
EFA AFENTY

The Strategy Module will maintain priority queues for all I/O requests and will supply the necessary calls to the appropriate I/O Adapter to execute the requests. The interface between the Strategy Modules and their I/O Adapters is defined by the nature of the I/O device and cannot be specified by a single set of calls.

CHAPTER VII/O CONTROL MODULE TO STRATEGY MODULES

When the I/O control module receives synchronization requests from the file coordinator it may modify tables of its own, followed by calls to the strategy modules to insure synchronization where necessary.

IOHLTf - called when I/O for a particular user is to be stopped.

```

    TSX  IOHLTf,4
    PAR  PSTAT
    ...
    PSTAT PZE  AFST
  
```

PSTAT - contains the address of the base of the Active File Status Table (AFST) for the user whose I/O is to be blocked. The strategy module returns to the I/O control module when all I/O that has been removed from this user's queues has been completed, and all necessary information has been updated into the AFST. (IOHALT)

DRAINf - called when all I/O queues are to be emptied, or all I/O is to be stopped.

```

    TSX  DRAINf,4
  
```

The strategy module returns when its I/O strategy section has no I/O waiting for initiation or completion. (IOHALT, DRAIN)

IOBGNf - called when I/O for some user has been unblocked.

```

    TSX  IOBGNf,4
  
```

Strategy modules are to take any necessary measures needed to restart I/O. (IOBEGN)

IORSTf - called when machine catastrophe has occurred with the possibility that I/O traps have been lost.

```

    TSX  IORSTf,4
  
```

The strategy module returns to the I/O control module after making calls to adapter module to take appropriate action.

CHAPTER VIISTRATEGY MODULES TO IOCM

The strategy modules make three different types of calls to the I/O control module: utility calls used during I/O requests from the buffer control module, requests for queue operations, and requests to search queues and extract I/O operations.

SSETUP - since the only parameter passed to the strategy modules by I/O calls from BUCM is the active file entry (AFENTY) this subroutine is provided to determine the AFST to which it belongs.

```

... entry from BUCM (save XR's, etc.)
CLA 1,4
STO TEMP
TSX SSETUP,4
PAR TEMP
SIW ASSOC

```

TEMP - temporary storage location.

ASSOC (AC) contains:

```
PZE AFENTY,,AFST
```

AFENTY - base of active file entry involved in call.

AFST - base of active file status table containing AFENTY.

CNTIO - used by strategy modules to determine length of IOLIST and number of records involved in transmission.

```

TSX CNTIO,4
PAR PIOLST,,NWDSPR
SIW WRDCNT
STQ REGCNT

```

PIOLST - contains the address received in the IOLIST parameter of call from BUCM.

WRDCNT (AC) - is the number of words in IOLIST (including the IOD word).

REGCNT (MQ) - is the total number of records to be transmitted by IOLIST (rounded - up to nearest integer).

NWDSPR - location of an integer containing the number of words per record for this device.

Queue manipulation calls (all calls must be made with traps disabled either by explicit 'ENB =0' when normally enabled, or during I/O traps.)

QUINT - initialize queues for this device belonging to this user if necessary.

TSX QUINT,4
EPA QUEUEf

QUEUEf - address of word containing queue 'key' for this user's queues on this device.

QTST - test whether desired number, 'n', of worst-case queue entries are available.

TSX QTST,4
EPA QUEUEf
PAR NUM,,ERROR

NUM - contains an integer, 'n', corresponding to number of queue entries which may need to be made later.

ERROR - return location if space for desired number of queue entries is not available.

QGET - obtain a queue entry ("bead") to hold a particular I/O request, and insert it into the queues according to its priority.

TSX QGET,4
EPA AFST
EPA AFENTY
PAR PRI,,QF
PAR SIZE,,ERROR
SLW QBASE

PRI - contains an integer, which is the I/O priority established for this file.

QF - contains an integer, the location relative to AFST of the user's queue pointer, (i.e., 'PZE QUEUEf').

SIZE - contains the integer size of the queue entry desired on this call (label + I/O list + ?).

ERROR - return if space not available for queue entry of length SIZE.

QBASE (AC) - contains address of 0-th element of queue entry.

IOTASK of AFENTY - will have been incremented by 1 on successful return.

QDEL - return a queue entry to "available space" after it has been processed fully.

TSX QDEL,4
EPA QUEUEf
EPA AFENTY
EPA QENTRY

TSX .REFRD,4 .REFRR, .TREFR
PAR SUBR

SUBR - will be called when an I/O completion trap occurs, or when no I/O is in process and queue entries have been made, by a call of the form

TSX SUBR,4
PAR FLAGS

FLAGS - will contain indications of parity or device errors, and may contain a device address. (Check adapter specifications for details.)

.IORFC - "force" a call to the strategy module if no I/O is currently in process.

TSX .IORFC,4 .FORCE, .TFFRC
* PAR CUNIT

CUNIT - contains PZE cC000(8) where 'cc' is the channel to be re-started.

.IORST - reset adapter for incomplete operation on device, make error call to SUBR.

TSX .IORST .DRSET .TFRST
* PAR CUNIT

QENTRY is the address of the 0-th element location of the queue entry being returned.

FINISW of QENTRY - must have been set non-zero.

IOTASK of AFENTY - will be decremented by one.

QSCRIP - called when all active queues for a file are to be removed, either on closing a file or on fatal error in processing previous queue requests.

```
TSX QSCRIP,4
EFA AFST
EFA AFENTY
PAR QF
```

All queues pertaining to the file AFENTY will be returned to available queue space.

IOTASK of AFENTY - is set to 0.

When the I/O strategy section of a strategy module wishes to set up and initiate I/O activity through its adapter, it calls the following entries in the I/O control module to search the user and queue lists for active I/O queue entries which have not been blocked. (All queue entries with 'FINISW' set non-zero are considered inactive.)

FNDUSR - initialize search and return first active queue entry of some user (based on user priority).

NXTUSR - terminate search of current user's queues, proceed to search queues for next user in user priority.

NXBEAD - obtain next active entry from this user's queues. (Current queue entry may have been by-passed for some reason: e.g. wrong channel, or serious error in this file.)

```
TSX FNDUSR,4
PAR NOUSR,,STOPIO
PAR QF
STQ IOSTAT
SLW QBASE
TSX NXTUSR,4
PAR NOUSR
TSX NXBEAD,4
PAR NOBEAD
STQ IOSTAT
SLW QBASE
PAR NOBEAD
SLW QBASE
```

NOUSR - user list has been searched to exhaustion: if strategy module has low-priority I/O to keep it busy (e.g. unchaining records of deleted files) it may do so.

STOPIO - all new I/O is being blocked due to other higher-speed I/O devices requesting service.

NOBEAD - no further active queue entries for this user; (normally followed by call to NXTUSR).

IOSTAT (MO) - contains

PZE AFST

base of active file status table for current user.

QBASE (AC) - contains

PZE QENTRY,,AFENTRY

base of queue-entry and of active-file entry to which it pertains.

PROLST - process one record of I/O from queue entry and perform relocation and protection checks upon it.

'NWDSPR' (or fewer) words of I/O transmission will be removed from the I/O list in the queue entry, and assembled for the strategy module's call to the I/O adapter. The appropriate modifications will be made to the queue's I/O list, and all word-counts are zero, FINISW will be set non-zero in the queue entry. (At this point the last I/O transmission for this file will not have taken place, and a previous request may still be in process subject to I/O errors, so the queue entry should not be removed by a call to QDEL until after the completion of this I/O transmission.

TSE PROLST,4
EFA LIST
PAQ QBASE,,IOSTAT
PAE QLST
PAL NWDSPR,,FMVERR

QBASE,IOSTAT - as returned from FNDUSR, etc.

LIST - location where I/O transmission list for this record is to be assembled.

QLST - contains an integer, the location relative to QENTRY of the beginning of the I/O transmission list.

FMVERR - error return for protection error detected while processing list.

VIII-1

This section describes calls to the track management module calls specific to tape-strategy module, and controls calls to a hypothetical I/O adapter.

The file co-ordinator calls initialization and updating entries of track management module.

TD/DRUMS - initialize drum usage tables
TD/DISKS - initialize disk usage tables
UPDRUM - re-write drum usage tables
UPDISK - rewrite disk usage tables

TSX tentry,4

(All calls identical.)

The disk-drum strategy module calls the track management module to obtain (release) records when writing (deleting) files. (Calls made with traps disabled.)

GETTRK - obtains a record from usage tables.
DELTRK - release a record, return it to usage tables.

TSX ~~xxx~~TRK,4
PAR RECADR
PAR ERROR
SLW NEWREC

NEWREC (AC) - contains BCD record address (MNTTR) of record obtained.

RECADR - may contain

PZE DEVICE

to obtain an initial record address for a file on DEVICE, or

BCI 1,MNTTR

the record address of the current record for a file which is being extended or deleted.

ERROR - AC contains error code:
3 - no more records available
2 - illegal call.
(should not occur for DELTRK).

FCOR - TFSM - when a user wishes to mount a tape for subsequent reading or writing, the File Coordinator will reflect the following call to the Tape Strategy Module.

TSX MNTTAP,4
EFA IOBASE
PAR UNTTNO, CHANNEL

PAR Y,,QWAIT C(Y)= PZE MESSAG,,LENGTH

CHANNO - is the logical channel no. (1=channel A, 2 = channel B). If CHANNO is zero, the Tape Strategy Module will pick a channel for the user. UNITNO is the logical tape no. which the user wishes to assign to this tape. MESSAG→MESSAG+LENGTH-1 is a BCD message of up to 20 words. This message will be printed to the operator to help him identify the desired tape. The Tape Strategy Module will attempt to mount only one tape at a time. If a tape mount operation is already in progress, the Tape Strategy Module will return control to the location QWAIT. If no tapes are currently available in the tape pool specified by the left half of URCODE, the strategy module will return control to location ERROR.

To dismount a tape, the File Coordinator will generate the following call to the Tape Strategy Module.

TSX UMTTAP,4
EFA IOBASE
PAR UNITNO
PAR Y,,ERROR C(Y)= PZE MESSAG,,LENGTH

If the tape specified by UNITNO is currently in use, the strategy module will return control to location ERROR.

To unmount all tapes assigned to a particular user, the following call will be reflected to the tape strategy module.

TSX UMTALL,4
EFA IOBASE

All tapes, if any, that are currently assigned to this user will be dismounted with an appropriate comment to the operator.

To verify the label of the tape just mounted, the File Coordinator will reflect the following call.

TSX VERTAP,4
EFA IOBASE
PAR UNITNO
PAR Y,,MEM C(Y)= PZE LABEL,,LENGTH
PAR ERR1,,ERR2
PAR ERR3,,ERR4

LABEL is a BCD label to be found in the memory specified by MEM. The maximum LENGTH of LABEL is 4 words. If the tape specified by UNITNO is still in the process of being mounted, or the label has not yet been read, control will be returned to location QWAIT. The remaining parameters are locations of error returns and have the following meanings.

ENR1 Label does not match, try again up to 5 times
ERR2 Label is unreadable
ERR3 Specified tape does not exist
ERR4 Tape cannot be mounted, operations difficulties

VIII-3

To initialize a new tape for subsequent processing by the File System, the File Coordinator will reflect the following call.

```
TSX  LBLTAP, 4
EPA  IOBASE
PAR  UNITNO,,QWAIT
PAR  Y,,MEM
PAR  ERR1,,ERR2
PAR  ERR3
```

The parameters UNITNO, QWAIT, Y and MEM have the same meaning as in VERTAP. The remaining parameters are locations of error returns and have the following meanings.

```
ERR1  Label cannot be written (bad tape)
ERR2  Specified tape does not exist
ERR3  Tape cannot be mounted, operations difficulties
```

When the operator responds to a request from the Tape Strategy Module, the following call will be issued by the clock section of CTSS.

```
TSX  TAPKEY,4
PAR  CODE
```

This call is given during a trap and therefore all traps are disabled. The contents of CODE is an integer from 1-3 having one of the following meanings:

```
Unload tape with physical address in decs.
Previous request has been preformed
Previous request is illegal (no such tape etc.)
Cannot comply, operations difficulties
```

The following paragraphs describe sample calls to a hypothetical I/O adapter. The permissible names will be listed on the right of the call; an optional argument, marked by '*', may occur as noted where a specific channel number is necessary for the reference.

.INITD - initialize device adapter.

```
TSX  .INITD,4      .DINIT,.TINIT
PAR  ENABLE
```

ENABLE - contains:

```
PZE  ENBWD
```

the location of the system common enable word

.REFRD - establish trap reference subroutine for I/O strategy section of strategy module.

APPENDIX

Data Bases and Miscellaneous Data-Formats

This section contains descriptions of data-bases used in various portions of the file system, record- and file-label formats, and interpretation of 'POINTR' for disk files.

- A) Active File Status Table (AFST):
One for each of DUSERS
- B) Queues (QUEUEf):
One for each of FMAX devices in each AFST
- C) Active File Entry (AFENTRY):
One or more per AFST
- D) Record- and file-label format
- E) Interpretation of POINTR for files on disk/drum
- F) Tables within file-coordinator:
UTABLE, USRTBL, PRIORT
- G) Deletion algorithms in Disk/Drum Strategy Module

APPENDIX A-1

The Active File Status Table. (AFST) contains information pertaining to users' activity with respect to the file system, error information, pointers to users' I/O queues, and information pertaining to each user's active files.

Names in parenthesis refer to File-Coordinator entries directly involved with setting and modifying the information contained

Format information for the various entries appears after the table.

<u>Definition</u>	<u>Name</u>	<u>Usage</u>
0	PROBNO	NAME1 of user's U.F.D. (ATTACH)
1	PROGNO	NAME2 of user's U.F.D. (ATTACH)
2	AUTHNO	User's "author" number in binary
3	URCODE	User's restriction code
4	LIMITS	User's memory limits
5	RELOC	User's relocation
6	FPRIOR	User's file priority (SETPRI)
7	ERRILC	User's ILC on call with error
10	ERNAME	Name of file-system entry involved in last error
11	ERCODE	Error code of last error
12	IOCODE	I/O error code if any of last error
13	ENAME1	NAME1, NAME2 of file involved in last error
14	ENAME2	
15	ERCALL	Location within file-coordinator where error occurred
16	QUEUE1	Pointers to user's I/O queues (USTAT)
24	QUEUE7	
25	AFENTRY	First active file entry for user

Symbol

PROBNO, PROGNO, ERNAME1
ENAME1, ENAME2 }

AUTHNO
URCODE
LIMITS
RELOC
FPRIOR
ERRILC
ERRCODE
IOCODE
ERCALL
QUEUE F

Format

BCD (36 bits)

Binary 0 → 2¹⁸ - 1
Binary bits (See AG.7.03)
PZE LOWADR,, HIADR
PZE RELCON
Integer 1 → 7
PZE ADR,, MEMORY
Integer 1 → 10₁₀
Integer 1 → 7
PZE ADR
PZE Qf,, Qlf

APPENDIX B

Structure of I/O Queues

The I/O Queues (QUEUEF) are used to hold I/O operation descriptions from the time they are entered by strategy modules until they can be acted upon and removed during I/O traps.

All information contained within the queues is (or should be) relocatable, either with respect to the queues themselves, the user's active file status table and active file entries, or the user's core image.

Manipulation of queue entries is done only by the queue management module (QMAN): inserting and removing queue entries, and maintaining free storage within queue structures, while searching is done by FNDUSR, et al.

Each queue block is defined by a pointer in an AFST of the form:

QUEUEF PZE Qf, QLF

"Qf" is the (absolute) location of the 0th location of a block of queue storage of length 'QLf'. A queue entry consists of a 2-word header and a variable length data block:

0	3	18	21
QNUM	QLNG	FINISW	NEXTQ
	FUNCT		PAFST

A free storage block contains a one-word header:

0	3	18	21
	FREEP		FRELNG

NEXTQ - pointer to next queue entry.

FREEP - pointer to next free-storage block.

(Pointers are relative to origin of entire queue block; 0 means no successor entry).

QLNG - length of queue entry (Header (2) + data (0 - 7))

FRELNG - length of free-storage block including header word.

QNUM - queuing priority for this entry (equals PRIOR from AFENTY): entries are inserted in ascending order by QNUM, and in time sequence for entry for same value of QNUM (integer 1 - 7).

FUNCT - code for I/O operation connected with this queue entry; set by strategy modules after return from QGET.

PAFST - positive offset relative to base of user's AFST to determine base of AFENY.

FINISW - boolean switch set when usefulness of this entry is ended. Set to FALSE originally by QGKT; set to TRUE by PROLST upon exhaustion of I/O list or by strategy module if entry does not contain an I/O transmission list.

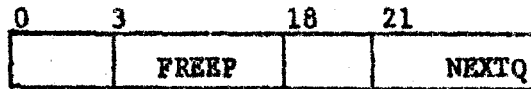
Note: FNDUSR et al., will by-pass all queue entries containing FINISW with value TRUE; QDEL will not delete a queue entry whose FINISW is FALSE.

QNUM and FINISW are 3-bit quantities in prefix and tag, respectively; all others are 15-bit quantities in address or decrement as shown.

QWORDS - is defined as the first word in the data section of a queue entry if QLNG is not 2.

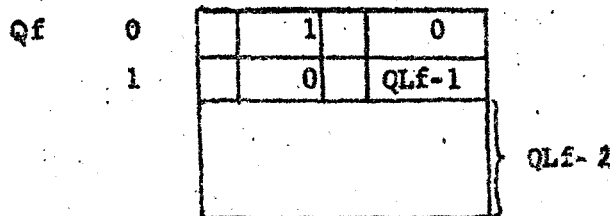
The variable data section may be of length 0 if FUNCT is a specific request to position at the beginning of a file for instance; or, more typically, contains LABEL and the entire contents of the I/O list (including IOD) for an I/O transmission request.

The first word of the entire queue block contains a "starter" word:



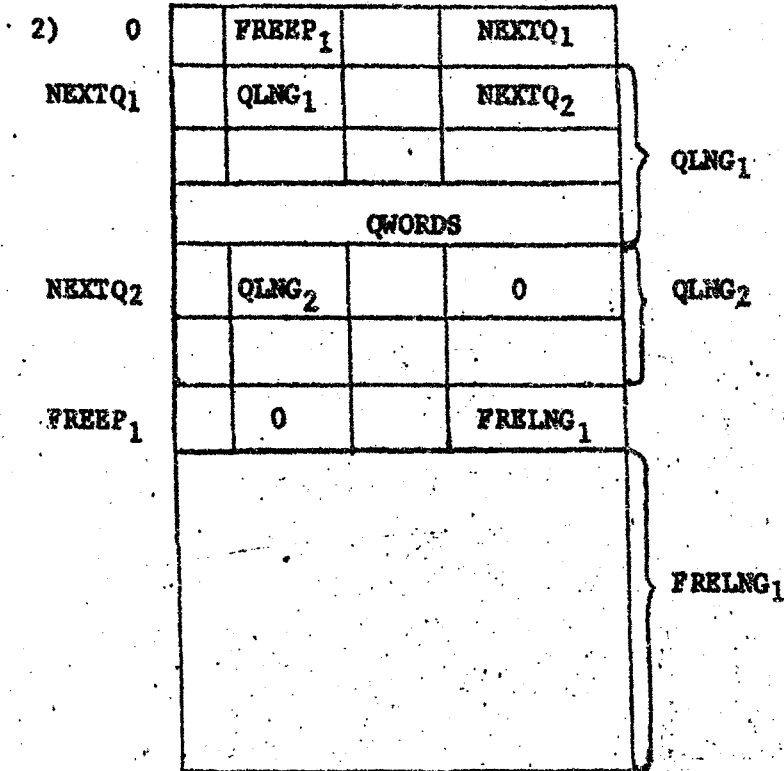
Examples:

- 1) A queue block which has just been initialized, or contains no active entries, will appear like this



Word 0 indicates no active queue entries (NEXTQ = 0) and a free-storage block in word 1.

Word 1 indicates no further free blocks (FREEP = 0) and that this block is of length 'QLf - 1', i.e., this word and the rest of the block are "free" to accept appropriate queue entries.



APPENDIX C

Active File Entry (AFENTRY)

All information concerning the status of an active file is passed through an entry in the Active File Status Table. The format of this entry is described below. The numbers in "()" indicate the number of binary bits assigned to the function. The character "-" is used to indicate an unused bit position.

- 0. RPROBN(36)
- 1. RPROGN(36)
- 2. RNAME1(36)
- 3. RNAME2(36)
- 4. RMODE(8), (unused)(28)
- 5. FPROBN(36) (set by File Coordinator)
- 6. FPROGN(36) (set by File Coordinator)
- 7. FNAME1(36) (set by File Coordinator)
- 10. FNAME2(36) (set by File Coordinator)
- 11. DAYTIM(36) (set by File Coordinator)
- 12. DATELU(18), AUTHOR(18) (set by File Coordinator)
- 13. MODE(8), ILOCK(10), F(3), RCOUNT(15)
- 14. S(1), W(1), R(1), NORECS(15), P(3), LCOUNT(15)
- 15. POINTR(36)
- 16. ASGNSW(3), REDREC(15), ---, REDWRD(15)
- 17. ---, WRTREC(15), ---, WRTWRD(15)
- 20. -, CHNG(1), PRIME(1), BUFREC(15), ---, BUFADR(15)
- 21. ---, WINDEX(15), ---, DINDEK(15)
- 22. -, DR(1), DW(1), DCOUNT(15), ---, DADDRS(15)
- 23. PRIOR(3), IOTASK(15), EFLAG(3), CURREC(15)
- 24. SMIMP1(36)
- 25. SMIMP2(36)

The following list describes the functions of the variable listed above. The initials in parenthesis indicate which modules may modify that variable. FC is the File Coordinator, BCM is the Buffer Control Module and SM is the Strategy Module. The module whose initials are listed first will be the module that is responsible for initializing the variable.

- RPROBN - Reference problem number (BCD); 0 if entry inactive.
- RPROGN - " programmer " (BCD).
- RNAME1 " file name, Part 1.
- RNAME2 " file name, Part 2.
- RMODE " mode (effective mode for linked file references.)
- FPROBN - (FC), user problem number (bcd).
- FPROGN - (FC), user programmer number (bcd).
- FNAME1 - (FC), file name, part 1.
- FNAME2 - (FC), file name, part 2.
- DAYTIM - (FC), date and time file was created or last modified YEAR(9), MONTH(4), DAY(5), TIME(18).
- DATELU - (FC), date file was last used.
- AUTHOR - (FC), programmer no. of user who last modified this file (in binary).
- MODE - (FC) mode of file.

ILOCK- (FC), used in file interlock mechanism (Prog. Guide AD, 2)
 F- (FC), device (1-3) on which file resides.
 RCOUNT- (FC, SM), number of words per record for this device.
 S (FC), non-zero if file open when USAVE called.
 W- (FC), non-zero if file open for writing.
 R- (FC), non-zero if file open for reading.
 NORECS- (FC, BCM), number of records in this file.
 P- (FC), reserved for compatibility with future systems.
 LCOUNT- (FC, BCM), number of words in last record of file.
 POINTR- (FC, SM), pointer to beginning of file (interpreted by SM).
 ASGNSW- (SM), switch for extension status of file.
 REDREC- (BCM), record no. of record containing the next word to be
 read from the file.
 REDWRD- (BCM), address of word within REDREC to be read next.
 WRTREC- (BCM), record no. of record containing the next address
 to be written.
 WRTWRD- (BCM), address within WRTREC to be written next.
 CHNG- (BCM), non-zero if contents of buffer differ from that
 of corresponding record.
 PRIME- (BCM), non-zero if contents of buffer represent a complete
 file record.
 BUFREC- (BCM), record no. of record contained in the buffer.
 BUFADR- (BCM), address of file buffer.
 WINDEX- (BCM), no. of words written into the buffer (CHNG=1 and
 PRIME=1).
 DINDEX- (BCM), index within buffer of words to be transmitted to
 or from user's memory before initiating new I/O for
 this file.
 DR- (BCM), non-zero when necessary to move words from buffer
 to user's memory before initiating additional I/O
 for this file.
 DW- (BCM), non-zero when necessary to move words from user's
 memory to buffer in order to complete the previous
 I/O request for this file.
 DCOUNT- (BCM), no. of words to move if DR or DW is non-zero
 DADDRS- (BCM), address in user's memory to start moving to or from
 when DR or DW is non-zero.
 PRIOR- (BCM), file I/O priority (0-7).
 IOTASK- (SM), no. of sub-tasks to be completed to complete previous
 I/O request for this file.
 EFLAG- (SM), non-zero if error during previous request (PERROR=1,
 FERROR=2).
 CURREC- (SM), record currently in process.
 SMIMP1 (SM), temporary for strategy module use.
 SMIMP2 (SM), " " " " "

APPENDIX D

- I. **Disk/Drum Record Label Format:** Each record on 1302 disk/7320 drum consists of 435 words, 3 words of record label, followed by 432 words of user data.

The 3 word label is interpreted as follows:

1	BACKP	
2	FOREP	
3	LCOUNT	RECNO

1. BACKP Backward chain pointer (BCD)
2. FOREP Forward chain pointer (BCD)
3. Sequence verification

The chain pointers contain the BCD record address of the preceding (following) record. Since BCD 0 is represented as octal 12, a binary 0 in a pointer indicates no predecessor (successor) record, i.e. BACKP is 0 for the first record of the file, and FOREP is 0 for the last record.

RECNO is the binary sequence number of the file, 1 → NORECS, sequenced by ones.

LCOUNT is the count of the number of words in the last record of the file; the field is zero in all except the last record; where its value may be 1 → RCOUNT.

If FOREP is zero, LCOUNT must be non-zero, and vice-versa.

- II. **Tape Record Label Format:** Since magnetic tape is by nature a sequential access medium (as opposed to random access), tape record labels are only 1 word long; followed by 432 words of data:

LCOUNT	RECNO
--------	-------

RECNO - sequential record numbers, 1 → NORECS, sequenced by ones.

LCOUNT 0 in all records except last, where it indicates the number of words in the record (1→ 432).

- III. **Tape File Format:** A detailed description of tape file label formats is contained in CTSS Programmer's Guide 2nd Edition, section AG.5.05.

A file of a tape will contain the following information.

```

TAPE FILE LABEL
(EOF)
Records of file
Last record (LCOUNT ≠ 0)
(EOF)
End-of-file Label
(EOF)
    
```

APPENDIX E

For files maintained on 7320 drum/1302 disk (F = 1 or 2) the item POINTER in the AFENTY and the U.F.D. entry for the file contains the addresses of the first and last records of the file:

POINTER ⁰ ¹⁸
LAST FIRST

Each 18-bit half represents the binary conversion of the 6-character BCD record address in the following format:

R L TTTT

- R - Logical record number (0 for drum, 0 or 1 for disk).
- L - Logical module number (4 for drum, 0 - 3 for disk).
- TTTT - Track address (0000-0399 for drum, 0000 - 9999 for disk).

Within the Disk/Drum Strategy Module, and in the chain words preceding the data in each record, the unpacked format (after binary-to-BCD conversion) is:

L TTTT R

i.e., rotated one character position to the left, with all 00₈ characters replaced by 12₈. (from even-parity representation of character 0).

Within the Disk/Drum Adapter and in the record-address written on each track, the BCD representation is rotated one character position further to the left:

TTTT R L

and is used to verify that the record area being read or written is correct.

When disk/drum errors occur, an online message is printed,

DISK/DRUM ^{READ} _{WRITE} ERROR, OPN= PP AM TTTT RL

SINSE DATA = (12 octal digits)

- A - Physical access mechanism.
- M - Physical module.

The current correspondence between L, A, and M is: (Jan. 1965)

L	A	M	
0	0	0	}
1	1	0	
2	0	1	
3	1	1	}
4	0	2	
			Disk
			Drum

PP is the 7631 File control operation order (IBM 7094, A22-6703-1, p. 87)

82	DVSR	Verify single record
83	DWRF	Write format
85	DUCY	Verify, cylinder operation
86	DWEC	Write check
88	DVTA	Verify track with addresses
89	DVHA	Verify home address.

SENSE DATA is decoded according to table in A22-6703-1 on p. 80.

APPENDIX F

Three tables are kept within the file-coordinator with entries for external reference: UTABLE, USRTBL, and PRIORT. They are referenced currently only by the file coordinator and IOCM; this section deals with their contents and significance.

UTABLE - location of AFST for current user

PZE AFSTLC,,AFSTCN

AFSTLC base or origin of user's AFST

AFSTCN length of AFST block.

USRTBL - vector of length DUSERS+1 containing AFST pointers for file system users. (User 0 = file system)

Format is same as in UTABLE.

PRIORT - sorted priority table for all users except file system:

PRIORT PZE USERNO,,UPRIOR

..... DUSERS in length

PZE 0 End of table

USERNO user's number (as at SETUSR, etc.)

UPRIOR user's priority (set by SETUSR)

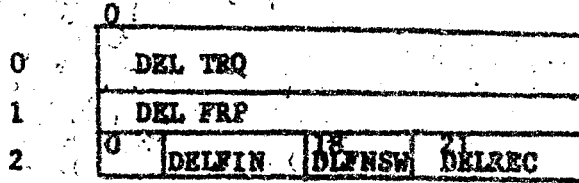
The table is maintained sorted on the basis of user priority, smaller value of UPRIOR indicating higher I/O priority.

APPENDIX G

Deletion Algorithm for Disk/Drum Files

When a disk/drum file is deleted or truncated, the process of searching the chain pointers of the file (unchaining), which is necessary for proper re-use of its records, is deferred until such time as the strategy module has no other I/O to process (NOUSR return from FNDUSR/NXTUSR).

Information about files to unchain is kept in a ring buffer in DDST, each entry containing:



DELTRQ - BCD record address of record to be read; binary zero if slot is empty.

DELFRP - BCD record address expected in FOREP of current record (normally = 0 for fresh entry)

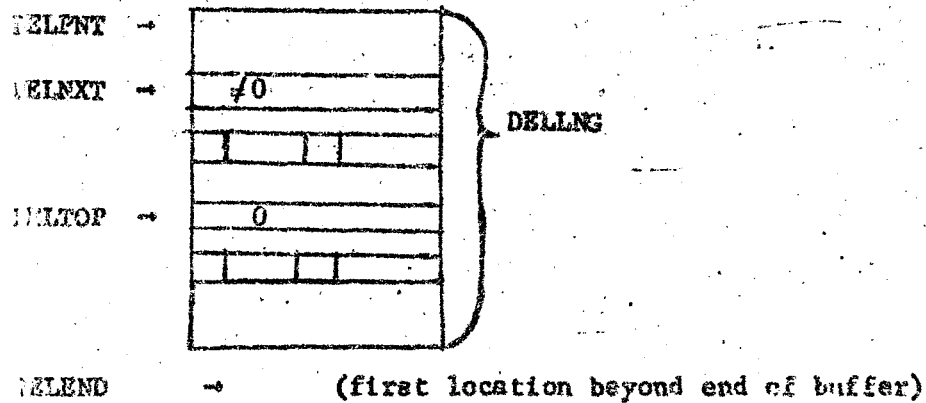
DELREC - record sequence number expected.

DELFIN - record sequence number from which delete begins.

DELNSW - FINISW for deletion entries: set for termination of truncation.

NOTE: Deletion is done by reading the file from the last record towards the beginning, following the BACKP chain, and subtracting one from DELREC for each record read, until DELREC matches DELFIN. DELFRP is used to verify that adjacent records point to each other.

The ring buffer contains room for a number of deletion entries (currently 20). It is manipulated by 2 reference pointers: beginning of buffer (DELPNT) and end of buffer (DELEND); and 2 working pointers: next entry (DELTOP) and active delete (DELNXT).



When a request to delete is made, DELTRQ of DELTOP is checked: if zero the request is honored, the entry made and DELTOP incremented to next entry (if increment moves it beyond DELEND, it is reset to DELTOP); if non-zero the entire ring buffer has active requests, and control will return to QWAIT.

When the I/O strategy section of DDST finds no I/O to set up, it checks DELTRQ of DELNXT and if it is non-zero proceeds to setup for reading the next record of the file to allow its re-use. If DELTRQ of DELNXT is zero (DELNXT should also equal DELTOP at this time), all active delete requests have been satisfied and the strategy module becomes inactive.

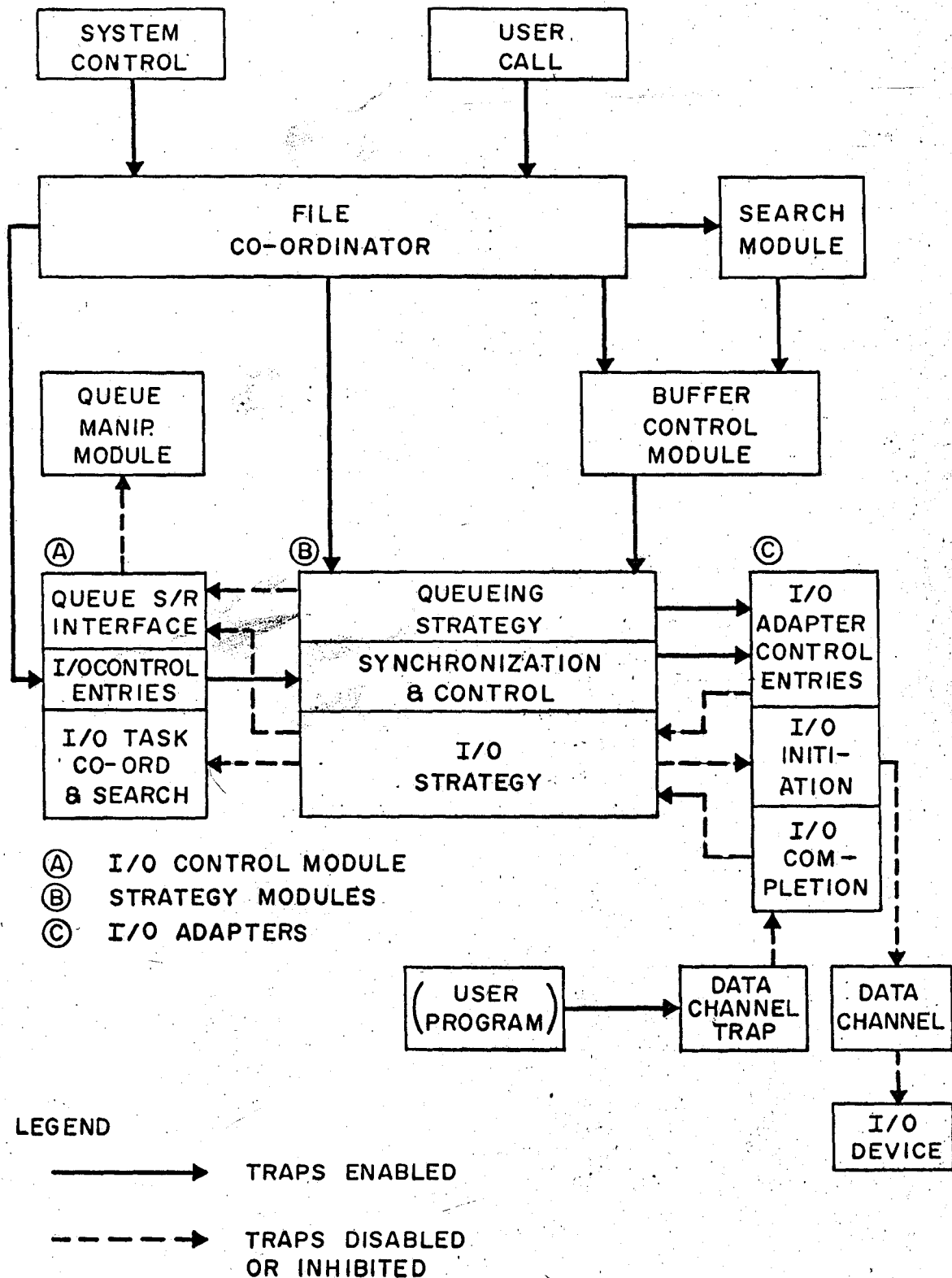


FIGURE 1 SIMPLIFIED FLOW DIAGRAM FOR FILE I/O SYSTEM (7094)

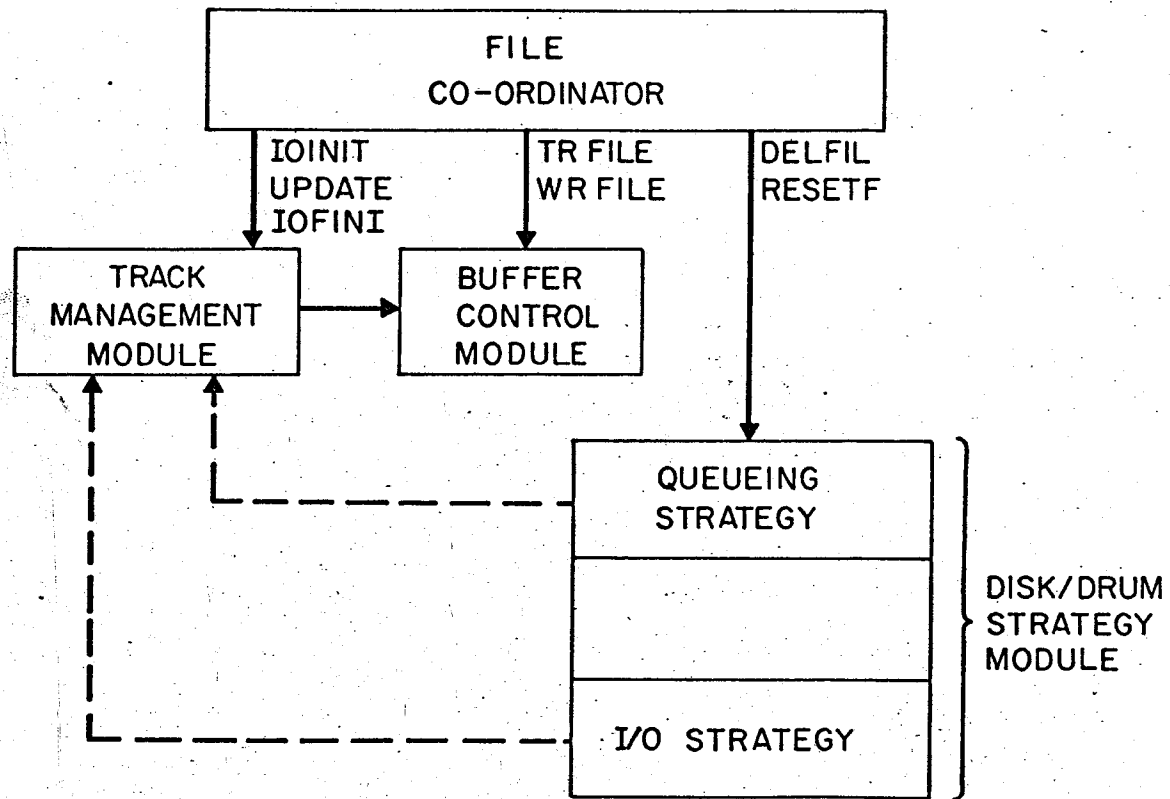


FIGURE 2 FLOW DIAGRAM FOR CALLS AFFECTING TRACK MANAGEMENT MODULE