

RANDOM FILE OPERATING SYSTEM

SERIES F

SPECIFICATIONS

TYMSHARE, INC.  
Technical Division  
925 E. Meadow Drive  
Palo Alto, California

April 25, 1969

COMPANY CONFIDENTIAL AND PROPRIETARY

The Program Logic, Code, Designs, and Techniques  
Set Forth on This Document, and all Information  
Relating Thereto, are and Shall Remain the Property  
of TYMSHARE, INC., a Corporation.

IN PROGRESS DEVELOPMENT  
SPECIFICATIONS

I. OPERATING SYSTEMS

Series F

A. Random File Version

General Description

The major change in this version will be the addition of large, random access files. The current sequential access files will be compatible. Files will be declare-able as append only. A new disk mapping concept will be implemented. File access privileges will be increased.

Features

1. Random Access File Manipulation

Opening, reading, writing and closing sequential files is accomplished in the same manner as in previous systems. Additional SYSPOPS are available for manipulation of random files.

a. File Parameters

F=File number returned by the open BRS.

CP=Cursor position. The address in the file character space of the next character, word or block to be written or read. Lowest possible address is 0 and the highest 14B6-1.

CPTOP=Location of highest CP position plus 1 which data has been written. It can be compared to end-of-file.

b. User Privileges

The following privileges, controllable during opening of a file by use of SYSPOPS, are enabled or disabled by the EXEC.

P1 User may set file to read mode.

P2 User may set file to write mode.

P3 User may erase information from a file.

P4 Not used.

## Random File Operating System Specifications

P5 Not used.

P6 User may effectively execute the Set Cursor Position SYSPOP (SCP).

P7 Not used.

P8 Not used.

c. Opening Files for Input

Files may be opened for input using BRS 62 or BRS 64. The CP will be set at zero. Privileges P1 and P6 only will be enabled, thus a file opened for input cannot be put into a write mode.

d. Opening Files for Output

Files may be opened for output using BRS 63 or BRS 65. The desired privileges are controlled by the high order bits (1-8) in the A register. If none of these bits are on, the file is considered sequential, the old data is deleted, and the EXEC will enable privileges P1, P2, P3 and P6. If any of the privileged bits are on the old data is retained, and these privileges are enabled accordingly. Bit 1 corresponds to P8, Bit 2 to P7, etc.

e. Closing Files

The same BRS's that close sequential files are used to close random files.

f. File SYSPOPS

F=File number.

WIO F Increments the value of CP by three.  
Traps if CP is not a multiple of three.

CIO F Increments value of CP by one.

BIO F Increments value of CP by three times the number of words transferred. Traps if CP is not a multiple of three.

The above three SYSPOPS require the same arguments as in the past and are used in the same

## Random File Operating System Specifications

manner. There will be five new SYSPOPS for manipulating random files. They are:

1. RCP Read cursor position

Calling sequence: RCP F  
Normal Return

F=File Number

Description: Returns the current value of the CP in register A, with the sign bit set if in a write mode and returns an eight-bit privilege mark in bits 1-8 of X. Bit eight corresponds to P1, bit seven to P2, etc. Sign bit and bits 9-23 of register X are unchanged.

Registers effected: A, X

2. SCP Set cursor position

Calling sequence: LDA N  
SCP F  
Exception Return  
Normal Return

N=New value of the CP. If the sign bit is on, the write mode will be set, otherwise read mode is set.

F=File number

Exception return A is unchanged  
X contains error number  
X=1 cannot set mode specified by sign bit of A.  
X=2 argument greater than 1436-1.

Normal return Registers unchanged.

Registers effected: A, X

3. PCE Erase and set cursor position

Calling sequence: LDA N  
PCE F  
Exception Return  
Normal Return

N=New value of CP

F=File number

## Random File Operating System Specifications

Description: This SYSPOP increases the value of the CP from the current value to the values specified by bits 1-23 of register A. Bit zero of A is ignored. All characters from the old value of the CP to the new value of CP-1 are erased. The physical file size may decrease.

Exception return: A is unchanged  
 X contains error numbers  
 X=1 File not in write mode  
 X=2 P3 is disabled  
 X=3 Argument in A less than current CP.  
 X=4 Argument in A greater than CPTOP.

Normal return A,B,X are unchanged. The erase is completed. If the argument is less than CPTOP the new value of the CP is equal to the argument.

If the argument is equal to CPTOP the following occurs.

- a. Erase is completed from the original value of the CP to CPTOP-1.
- b. The new value of the CP is equal to its original value.
- c. CPTOP is decreased to equal the original value of the CP. This is the only way to decrease CPTOP.

4. RSP Read size parameters

Calling sequence RSP F  
 Normal Return

F=File number

Descriptions: This SYSPOP returns the file parameters in the A,B, and X registers as follows:

## Random File Operating System Specifications

A=Value of CPTOP  
B=Current physical size in characters  
X=Remaining number of characters in  
the file size quantum.

### 5. SSP Set Size Parameters

Calling Sequence LDA N  
SSP F  
Normal Return

N=Number of Characters  
F=File Number

Description: This is a privileged SYSPOP which can only be executed by forks with at least subsystem status and will trap if a fork without status attempts execution. The file size quantum will be set equal to the number of characters in the A register. However, this is limited to 1000000B characters.

Registers effected: A

### 2. Large Files

The file structure in this version of the operating system will allow a maximum size file of 3,145,727 characters. This size includes all overhead. The amount of actual data in the maximum size file will be roughly 96 percent of this. However, since we have nearly reached the capacity of our discs, the first release of this version will contain artificially smaller upper bounds on file size. This will help protect our systems from running out of disc space in the middle of a day's operation (3,145,727 characters use up 4,095 D bits). The details on how these artificial bounds will be implemented has yet to be determined.

### 3. Append Only Files

Files may be declared APPEND ONLY by the owner of a file directory through the use of the private write option in the EXEC DECLARE command as follows:

```
-DECLARE /FILE/ Lf  
PRIVATE  
WRITE ACCESS?A Cr
```

## Random File Operating System Specifications

A line feed may be used as a confirming character if the user wishes to set further options allowed by the DECLARE command. Any time an append only file is opened for output the old data is retained and the current output is appended.

## 4. Disc Mapping

A completely new mapping scheme will be used with this operating system. Any one of four areas that can be mapped will be mapped on startup. The mapping program MAP will make a preliminary survey of the disc and will determine which of the four areas contains the most free data blocks. This area will then be mapped unless the operator chooses to select one of the other areas. To make this selection, the operator types a line feed after calling the mapping program.

Example:

```

-/MAP/ Cr
MAP AREA 2

```

or

```

-/MAP/ Cr Lf
ENTER MAP AREA 3 Cr
MAP AREA 3

```

Area 2 will be mapped.

Area 3 will be mapped.

The map will be built in approximately twelve minutes. The disc reorder routine, BEEP, will be made obsolete by the new MAP.

## 5. New Line Printer Logic

It will no longer be possible to output directly to the printer. Output must first be put on a file then to the printer using a new printer program that will be supplied. The printer must be assigned to the user. It cannot be assigned concurrently with a magnetic tape unit.

## 6. New Specifications for FDC Command

If a directory is declared with the FDC command, it will have the following attributes:

a. SHAREABLE?Y (Normally not shareable.)

All users in the account may use GFD to access the files in this directory. Their access is subject to the private controls on the files set

## Random File Operating System Specifications

by the owner or the Account Supervisor with the DECLARE command. Normally, the users sharing this directory will not be able to use the DECLARE command to change the private controls on any of the files (see CONTROLS, below). All users may also access the files in this directory by using the parentheses notation.

## b. LISTABLE?N (Normally listable)

Files in this directory will be accessible to all users in this account through GFD or the parentheses notation. However, the FILES and DIRECTORY commands will not be available.

## c. CONTROLS?Y (Normally No)

All users in this account may access this directory and may also use the DECLARE command to change the private controls on any file. (See SHAREABLE, above.)

## d. NEW FILES?Y (Normally No)

Normally, only the owner of the directory will be able to add new files. If this declaration is made, then any user may add new files through GFD or the parentheses notation.

Only the owner or the Account Supervisor may use FDC command to change the directory controls.

## 7. Remote Command Files

The specification for files declared REMOTE has been changed. A remote command file will have private access to files in the directory in which it resides, even if it is executed from outside the directory. Files referenced without the parentheses notation will be taken as files in the calling directory. Files in the directory in which the remote command file resides must be referred to by the parentheses notation.

Example: A remote, public, command file called P1 is stored in directory Z5A. It is being executed from directory B. It reads data from a file in B called D1 and it creates and writes on a file, R1, in A. Directory A is a normal directory that is not

## Random File Operating System Specifications

shareable and will not accept new files. However, the remote declaration allows P1 to add a new file to A, although P1 can be executed from any other directory. In P1 the above files will be referred to as:

D1 and (Z5A)R1

## 8. Restored GFD Privileges for Account Supervisors

With the GFD command, the Account Supervisor will be able to use any file directory in his account with the same privileges as if he logs into it. He will be able to:

- a. Write on any file that is not private, read only.
- b. DECLARE
- c. REMOVE
- d. RENAME
- e. INIT
- f. FDC

However, he will not be able to add new files unless the directory is set, or he sets it, to accept new files.

## 9. Additional File Access Privileges for Account Supervisors

All files in all directories in his account will be public to the Account Supervisor through the parentheses notation unless they have been declared otherwise. He will be able to use:

COPY  
DECLARE  
DELETE  
RENAME

on any files in another directory by using the parentheses notation. The RENAME command can be used to move files from one directory to another.

## 10. Restored Privileges for Operations

When operator status is set, the user will have the

Random File Operating System Specifications

same file privileges as the Account Supervisor,  
(see above) except the privileges will extend to  
all accounts.

## RANDOM FILES

### File Size Restrictions

File data is stored on the disc in 1400B character blocks and the cursor position associated with the beginning of any data block is a multiple of 1400B. The monitor does not write data blocks that are all zero; hence, a file that covers a large range of cursor positions may grab less disc space than a small file that is densely packed with information. Each file has a minimum overhead of 1400B characters to map every 40B blocks of file data. The details of the mapping overhead are complicated and are a function of the position of the data in the file and the amount of zero data.

Each file that is opened for output is assigned a physical file size quantum to limit the amount of additional disc space that the user may grab before closing the file. This quantum, 1,364,000B characters is refreshed every time the file is opened regardless of the current size. This quantum can be changed by privileged execution of the SSP SYSPOP (see below). 1,364,000B characters is equal to 63D pages. The file quantum reserve is depleted by writing both data and data mapping tables (index blocks).

### New File SYSPOPs

SSP F set size parameters. OP Code = 26B

This privileged SYSPOP can only be executed by programs with the highest status (EXEC status). The file size quantum is set equal to the number of characters specified in A. The EXEC will not issue this SYSPOP except in conjunction with special large files created in conjunction with the EXEC DECLARE command.

RSP F read size parameters. OP Code = 25B

A is set to CPTOP  
B is set to the total current physical size.  
X is set to the remaining file size quantum.

The PCE SYSPOP has been changed.

PCE F erase and set cursor position.

This SYSPOP should only be executed when the file is in write mode. PCE increases the cursor position from the current value to the value specified in bits 1-23 of A. Bit zero of A is ignored. Let CP1 be the current cursor position; let CP2 be the final CP. All the characters from CP1 to CP2-1 are set to zero. The physical file size may decrease.

## Random Files (cont., page 2)

Error Return No Skip, X=error number

X=1 File not in output mode  
X=2 P3 is disabled  
X=3 Argument in A is less than current CP  
X=4 Argument in A is greater than CPTOP

Normal Return, One Skip. A,B, and X unchanged.

The erase was completed. If the argument in A is less than CPTOP, the new CP is equal to CP2. If the argument is equal to CPTOP then:

- a. The erase is from CP1 to CPTOP-1.
- b. The file is positioned to CP1.
- c. After processes (a) and (b) are completed, CPTOP is decreased to equal CP1. This is the only way to decrease CPTOP.

### By the Way

When the file quantum runs out, I cause an instruction trap. You can avoid this unruly condition by watching the file size quantum via the RSP SYSPOP.

### Random File EXEC Changes -- by Verne Van Vlear

1. EXEC file BRS's for sequential files will work as they did in the past.
2. If a file is opened for input (using BRS 62 or 64), the random file cursor position may be set as desired. P1 and P6 are enabled.
3. No special options are available in opening new files.
4. In order to open random files for output (update), either the BRS 63 or 65 may be used. The desired options are controlled by the high order eight bits in the A register. If none of these bits are on, the file is considered to be sequential and the old data is deleted and P1, P2, P3 and P6 are enabled. If any of the privileged bits are on, the old data is not deleted and the privileges are determined by the user, not the EXEC. P8 corresponds to bit 0 of A; P1 is bit 7.

David Gardner  
2/18/69

DG:bm