Draft for approval
Published: 05/10/66

Identification

The Elementary File System (EFS)
D. Levinson

Purpose

The purpose of this paper is to describe an Elementary File System for the 645 Simulation System for use chiefly by EPL object programs. The program is basically a GMAP program and associates with the 645 simulator but does not, in the main, run under it, but GECOS.

Abstract of EFS System

1. Motivation

As stated above, the basic purpose of EFS is to provide EPL object programs with an I/O capability. It is anticipated that it will be used for checking out system programs which involve I/O such as editing routines. Of course, only the non-I/O parts of such programs can be checked out, but these require data files to operate on and can be provided by EFS.

2. Functions

Basic functions, as required by EPL are random and/or sequential I/O on logical record files. Thus, there will be an Open, Close, Read and Write with appropriate options. Options include temporary (life of process) permanent and foreign (from or to 7094 CTSS system) files, and 2 on-line print options, one for printing as I/O occurs, the other for bulk printing entire file at once when closed.

3. Structure

Figure 1 shows the overall structure. The EPL user writes one of the EPL I/O statements. The EPL compiler translates the statement into a call or sequence of calls to its object time subroutines. The object time subroutines interpret the calls and when actual I/O is required give one of the EFS calls. The EFS call interpretor, in turn, calls the 635-COBOL-Exec Driver (the Exec is an existing I/O system whose use will be justified later). The driver in turn calls GEFRC.

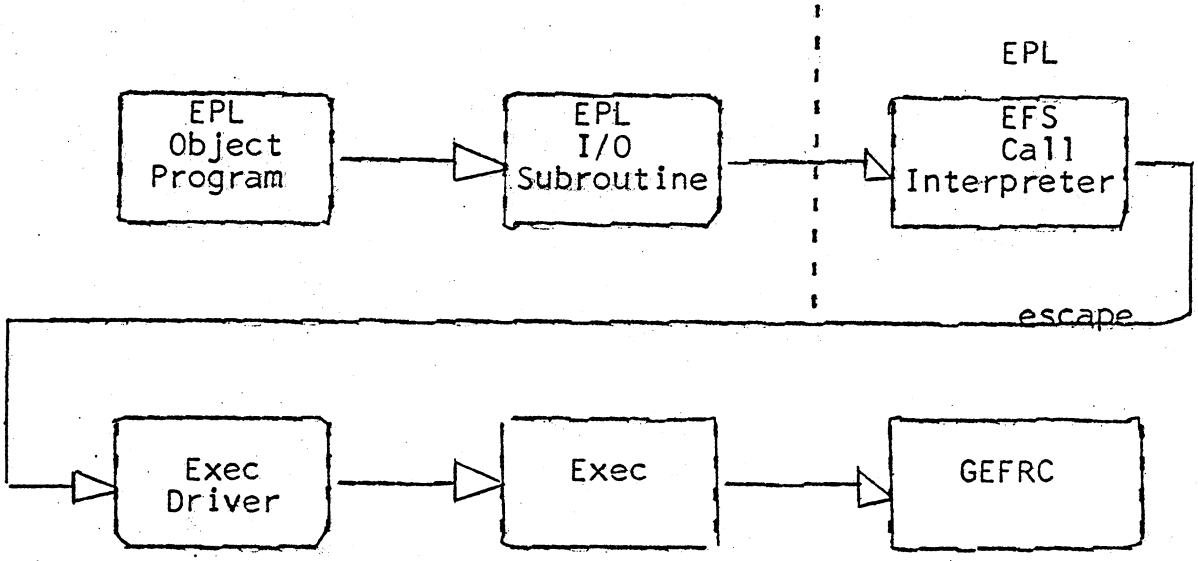

FIGURE 1

Functional Description of EFS

1. Design Objectives

The primary objective of EFS is to provide EPL with an I/O capability as soon as it can use it. Within this constraint the usual goals of efficiency. It is hoped, but not a design objective, that EFS will be the nucleus for a future simulator of the Multics file system. As a by-product EFS will provide an I/O capability to BSA, and may be used by 635 programs.

2. EPL Requirements

As stated above EPL needs to do random or sequential I/O on a given input file. The files are structured by EPL requirement, into logical records which may be read or written partially or completely with a given call. The size and number of records of a given file are arbitrary. The number of open files for a given process will be fixed at 15±5, a restriction dicatated by storage limitations. EPL also requires 3 types of files, temporary, permanent, and foreign. Temporary files live for the term of the process, permanent files until a request to delete, and foreign files whose source or destination is the 7094 CTSS system. Finally, 2 types of on-line print options are required as follows:

If the print option is given with the open statement, then the data transmitted by each call is converted from ASC-EPL to the GE character set and printed on-line with the 635 at the time of the request. If the print option is given with the close statement, then the entire file is converted and printed on-line with the 635 at the time of the close.

When given with the input statement the temporary, permanent or foreign option specifies the source of the file to EFS. When given with the close statement, they specify the destination. The options are given independently, but only one, of course, is valid for input. For output any or all are valid and specify multiple destinations.

3. Calling Sequences

By agreement with the EPL project, the following set of calls fulfill the requirements specified in 2.

```
OPEN      File Name                 character string ( )varying
          input or output           bit string (36)
                                      bit1(0 input
                                           (1 output
          Source (if input)         bit string (36)
          disposal (if output)      bit 1 (0 not foreign
                                          (1 foreign
                                    bit 2 (0 not save
                                          (1 save
                                    bit 3 (0 no print
                                          (1 print
          Status                  bit string (36)
                                    bit 1 (0 if no error
                                          (1 if error
```

NOTES:

1. "foreign" on input means file is to be read from 6.36 input tape. "foreign" on output means, file is to be written onto 6.36 output tape.

2. "save" on output means file is to be stored in 6.36 permanent file storage. "save" on input means file is to be retrieved from 6.36 permanent file storage.

3. Temporary files whose duration is the duration of the process in which they are created are obtained by default, i.e., not specifying either "foreign" or "save".

```
CLOSE     file-name          character string ( ) varying
          disposal           bit string (36)
                               bit 1 (0 not foreign
                                     (1 foreign
                               bit 2 (0 not save
                                     (1 save
                               bit 3 (0 no print
                                     (1 print
          status             bit string (36)
                               bit 1 (0 no error
                                     (1 error
```

NOTES:

1) "foreign" is redundant if also specified in open, and error if file was input; otherwise, the named file will be written onto 6.36 output tape.

2) "save" is redundant if also specified in open, error if open foreign, otherwise, temporary file will be stored in 6.36 permanent file system.

3) "print" is redundant if also specified in open (unless file is open for output and appended to), error if open foreign and not save (in which case must be output), otherwise, file is printed on line with 635.

4) Print on input or output means strings of elements specified in reads and writes will also be printed on line.

5) The foreign, save, print options may be given in all combinations except for input in which case the only restriction is that foreign and save may not be specified simultaneously.

READ	file-name	same as previous document
	location	
	record number	
	offset	
	number chars requested	
	status	bit string (36)
		bit 1 (0
		(1 EOF
		bit 2 (0
		(1 EOR
		bit 3 (0
		(1 char. exhaust
		bit 4 (0
		(1 any error
	number chars. transmitted	
WRITE	same as READ	
	except no record number	
	and status bit 1=1	
	means out of bounds and	
	bit 2 and bit 3 are meaningless.	
	Also, number of characters	
	transmitted is meaningless	