MULTICS TECHNICAL BULLETIN

To: Distribution

From: M. D. MacLaren

Subject: The New I/O System

Date: October 30, 1973

This bulletin updates the description of the new I/O system given in MSB-113. Except as explicitly noted to the contrary, the system described by these two bulletins is that which will be part of the first product release. The old I/O system will continue to be supported at MIT; indeed it will continue to be used by the system for some time.

CONTENTS

I. Additional Features (features not described in MSB-113)
II. Functional Changes (from system as described in MCB-113)
III. Corrections and Clarifications
IV. Installation (especially its immediate effect on users)

I. ADDITIONAL FEATURES

1. Stream Openings. The operations read and write are supported for stream openings. The operation read processes an entire line, transmitting characters up to a new line character or the end of the buffer. The new line character is not placed in the buffer; it and any characters that won't fit in the buffer are skipped. The operation write transmits the contents of the buffer and then a new line character.

2. PL/I Varying Strings. When a level-one varying string variable is written as a record, the current value of the variable is written without the length field. When a record is read into a level-one varying string, its value is set to the contents of the record and its current length is set to the record's length. Combined with (1) this permits using record I/O for terminals and unstructured files.

---

3. Line as Record. Attachment to a file may specify that an attempted sequential opening be interpreted as a stream opening, thus supporting read and write operations. In iox_$attach_file, this is done by specifying interp=2. In the attach command, the option record is used.

4. Destroy iocb. The call

```
call iox_$destroy_iocb (iocb_ptr, code);
```

informs the I/O system that an iocb will no longer be used. The pointer to it may no longer be used. The space occupied by the iocb is made available for use as another iocb. The argument variable iocb_ptr is set to null.

5. Multi-segment files. (possible feature in future - not part of first release). It is proposed that the storage system recognize multisegment files, and that hardcore entries be provided as the only means to add or delete segments in multi-segment files. This would insure basic consistency for multi-segment files and solve the current problems with write access.

6. Propagate. The entry iox_$propagate will do the following (in addition to its function as described in MSB-113).

6.1 Case: iocb.attach_descrip_ptr=null (i.e. The iocb is being detached). The entry sets iocb.open and iocb.detach to err_not_attached, and it sets all other operation entries in iocb to err_not_open. The pointers iocb.attach_data_ptr, iocb.open_descrip_ptr, and iocb.open_data_ptr are set to null.

6.2 Case: iocb.attach_descrip_ptr =null, iocb.open_descrip_ptr=null (the iocb is attached and closed). For all operations except open, control, and detach, iocb.op is set to err_not_open. Also, iocb.open_data_ptr is set to null.

6.3 Case: iocb.open_descrip_ptr^=null (the iocb is open). Propagate sets iocb.open and iocb.detach to err not_closed; and for all other operations, if iocb.op=err_not_open, then iocb.op is set to err_no_operation.

This change to propagate relieves the writer of an I/O attachment module (i.e. a DIM) of most of the work of maintaining the proper state for iocb entries. Also it insures that adding new entries to the iocb (e.g. to support asynchronous I/O) will not require changes to any existing modules. Note also that propagate has only one argument, iocb_ptr.

II. FUNCTIONAL CHANGES

1. The old file dim will remain available as file_ through ios. See section IV for restrictions on its use.

2. The truncate command will be left as is. A warning not to use it on record files will be added to the MPM.

3. Cross ring synonyms will not be supported in the first release.

4. Only nonstandard tapes will be supported in the first release.

5. Keeping file types in the branch will not be supported in the first release. The distinction between a sequential file and an indexed file is made using a code word at the beginning of the file.

6. The commands iomode and line_length are not part of the product. The new command ioc (io_op in MSB-113) may be used instead. A user-oriented command for setting terminal related characteristics may be provided in release two.

7. The subroutine get_at_entry_ is not part of the product.

III. CORRECTIONS-CLARIFICATIONS

1. The arguments rec_len and buff_len are fixed bin(21).

2. A PL/I external file constant is always associated with an iocb with the same name. Sysin and sysprint are attached as synonyms for user_input and user_output when appropriate

3. A Fortran logical unit nn is always associated with the iocb named filenn. The iocb's file05 and file06 are attached as synonyms for user_input and user_output when appropriate.

4. If PL/I or Fortran attaches an iocb, it detaches at close.

5. In the description of opening modes "keyed nonsequential" will be replaced by "direct". In the attach command and in the documentation, "append" will be replaced by "extend".

6. In an opening for keyed sequential update, write is supported.

7. More than one open iocb may be attached to the same file if all openings are for input.

8. The ring_bracket field of an iocb will be hidden from users and better terminology will be used to describe cross ring synonyms.

9. iox_$propagate has only one argument, iocb_ptr.

10. Read returns with status code = 0 if the record length is shorter than the buffer length.

11. Seek_key and read_key have another argument rec_len, which follows key in the argument list. Rec_len is set to the length of the record.

12. The command to perform i/o calls (new type) will be called "io_call", abbreviation "io". (It was called io_op in MSB-113.)

IV. INSTALLATION

The plan is to install the new I/O system some time in November. MPM documentation should be available in December. The immediate effects of the installation will be as follows.

1. The streams user_input, user_output, and error_output may only be attached through the new file DIM, one of the following existing DIMS

file_, tw_, discard_output_, mrdim_, ocdim_,
ntty_, absentee_, tek_, exec_com_,

or as a synonym for a stream attached in this manner. When one of these streams is attached to file_, the element size must be nine and the new-line character must be the single read delimiter. (These are the defaults for file_.)

2. The new file DIM features improved performance and will provide uniform support for record I/O in PL/I and Fortran (unformatted I/O) when the language i/o routines are converted. A stream may be attached through the new file DIM by the command

attach -streamname- -pn -pathname- -extend

The argument "-extend" is optional. It means that output to the file is to be added to the end of the file (normally the file is overwritten).

3. When a stream is attached through the new file DIM, the only ios_ calls allowed are to ios_$read, ios_$write, and ios_$detach. Note that the file_output command uses the new file DIM, and that PL/I and Fortran will use it in the future.

When PL/I and Fortran i/o is changed, the effects involving the i/o system will be as follows

4. PL/I I/O

4.1 Record files. Record files are read and written using the new file DIM. Files written using the old file manager (fm_) can no longer be read by PL/I. The new DIM supports two types of files: sequential and indexed. In sequential files, the records are stored contiguously in segments. In indexed files, the records are kept (logically) in key order.

4.2 Attachment. An external file constant is associated with the stream with the same name. An internal file constant is associated with a stream with a unique name. Opening a file in PL/I causes the associated stream to be attached (unless already attached). If no title option is given, and the file is not sysin or sysprint, the attachment is to the file whose pathname is the same as the PL/I file's name. When a title option is given, its meaning is the same as the command attach streamname title.

Thus to attach to file foo, the title should be "-pn foo".

4.3 Use of old file dim. If the stream associated with a PL/I file is attached through file_, the PL/I file can only be used for stream I/O.

5. Fortran I/O

5.1 Unformatted I/O. Unformatted I/O uses the new file DIM which reads and writes the new form of sequential file. (See 4.1). Old unformatted files can no longer be read by Fortran.

5.2 Use of old file dim. If the stream associated with a Fortran logical unit (e.g. file06) is attached through file_, the logical unit can only be used for formatted I/O, and backspace and rewind are not allowed.