Published: 08/31/66

Identification

Frames, Items, and the Current Item Number
J. F. Ossanna, V. A. Vyssotsky, G. G. Ziegler

Purpose

This section specifies the way in which external data is logically subdivided into items by the Multics I/O system, and the rules governing the value of the current item number.

Data Frames

Such I/O operations as read and write deal with data aggregates and media which are external to the procedure doing reading and writing. A read call must specify (explicitly or implicitly) where the data is to be read from, and a write call must specify where the data is to be put. In Multics, read and write calls (and certain other calls) specify a place in the external world by a two component address. The first component is a character string, specifying a frame of data. A frame may be a logical place, or it may be a physical device. A frame may, for example, be a file in the file system, or it may be the data on a physical reel of tape, or it may be data entered from the keyboard of a typewriter console, or output printed on the console. A frame may even be a part of some other frame. For example, given a frame 'alpha' which is a file in the file system, the frame 'beta' may be an item within the frame 'alpha'. The association of character string names with such particular places as these is established by attach calls, q. v.

A frame of data is either linear or sectional. A linear frame is intrinsically a single sequence of bits. It has a length, which is the number of bits in the frame, and which may be zero. If the length of a linear frame is greater than zero, it has a first bit and a last bit. A linear frame has no intrinsic structure beyond that implied by the preceding three sentences, but a structure may be imposed on it to allow convenient manipulation of the data in the frame. In particular, referencing the data by bit number and number of bits would be inconvenient. To alleviate this problem, when a linear frame is attached to a process, an element size in bits

is declared. Read and write calls specify how much data is to be read or written in terms of number of elements, and the declared element size is used by the I/O system to determine which bits are implied by a request for m elements starting with element number n. It must be emphasized that the element size is not a property of the frame, nor of the data in the frame, but is rather a property of the way data is transmitted to and from the frame.

A sectional frame is a collection of data units called records, each record having a record number. Record numbers are positive integers, and any two distinct records in a frame have different record numbers. However, there is no requirement that a frame actually contain a data record for each record number; a frame may, for example, consist of one record, whose record number is 62. The division of a sectional frame into records is an inherent property of the data in the frame. For example, if sectional frame alpha is created and 100 records of 144 bits each are written into it, then the frame itself contains sufficient information to show that it contains 100 records of 144 bits each, and not 200 records of 72 bits each.

Data Items

We shall use the word item to denote an element of a linear frame or a record of a sectional frame. In the following discussion of items and item numbers, all statements are applicable both to elements of linear frames and to records of sectional frames, unless the contrary is explicitly stated.

In the two component external data address used in a read or write call, the second component is an item number. This item number determines where within the specified frame the reading or writing should be done. However, the place to be read or written may be either relative to the beginning of the frame (random accessing) or relative to the current item number (sequential accessing).

The Current Item Number

Every item in a frame of data has an item number, which is a positive integer. From the time a frame of data is attached to a process until the time it is detached, the frame has a current item number. There may or may not be a data item whose item number is the current item number; for example just after creation of a new frame there is a current item number, but no data in the frame,

and hence no item corresponding to the current item number. Whenever a call to the I/O system causes the current item number to change, the status return shows whether or not there is an item with the current item number.

A frame can become attached to a process in two ways. It may be attached as a result of an explicit attach call; alternatively, an item of a frame is implicitly attached as a frame if it becomes the current item of the containing frame, and the current item of the containing frame has been attached as a frame. In either case, when a frame is attached the value of the current item number for that frame is set to 1. A frame can become detached from a process in two ways. It may be detached as a result of an explicit detach call; alternatively, the frame is implicitly detached if it is attached as the current item of some other frame, and the current item of the containing frame changes. In either case, when a frame is detached it ceases to have a current item number.

While a frame remains attached to a process, the current item number may be changed by calls to read, write, seek, delete, first or tail. If bit 4 of the status return is set on return from an I/O call (i.e. if the call was rejected), the call did not change the current item number. In the following discussion we shall take this rule for granted; all statements about changes in the current item number are implicitly qualified by the observation that if bit 4 is set in the status return, then the current item number is unchanged.

Consider a read, write, seek or delete call for a frame which gives itemno as the argument for determining the item number. Suppose the frame has been attached as a sequential frame. A read, write, find or delete call increments the item number by itemno before reading, writing or deleting. Then, for a read or write call, the item number is further incremented before return. If the frame is sectional, the current item number is incremented by 1 (i.e., the current record number on return is one more than the number of the record just read or written). If the frame is linear, the current item number is incremented by the number of elements actually read or written (i.e. the current element number on return is one more than the number of the last element read or written).

Now suppose instead that the frame had been attached as a random frame. Then, if itemno > 0, the current item number is set to itemno before any reading, writing or deleting is done. If itemno = 0, the current item

number does not change before reading, writing or deleting. For a read or write call the item number is again incremented before return, exactly as in the case of a sequential frame.

A first call for either a random or a sequential frame sets the current item number to 1. A tail call for either a random or a sequential frame sets the current item number to N+1, where N is the item number of the highest numbered item in the frame.

The only circumstance in which a frame may fail to have a current item number is hardware, software or operator error (bit 15 of the status return set to 1). After such an error, the current item number may or may not be defined, and if it is defined, its value may not be given by the above rules. The only sure way to re-establish a correct current item number in this case is to detach the frame and successfully reattach it. However, in many cases of hardware error, the current item number will still be appropriately defined.

Status and Traps Related to Current Item Number

Several status return bits are related to the current item number. Let i be the current item number at the time of a call, let I be the current item number at the time of return from the call, and let N be the number of the last item in the frame at time of return. Then, for any read, write, seek, delete, first or tail call that does not cause bit 4 or bit 15 to be set:

Bit 55 will be 1 if and only if I=1 (i.e. beginning of frame).

Bit 49 will be 1 if and only if I > N (i.e. last data).

Bit 43 will be 1 if and only if I > N+1 (i.e. end of frame).

Bit 56 will be 1 if and only if i > N+1 (i.e. beyond end of frame).

I/O traps may be taken corresponding to these four conditions. The default action for them is: no trap on bits 55, 49 and 43; trap on bit 56 and abort if a read call, otherwise complete the request and return.