Published: 08/31/66

Identification

Random Linear I/O
J. F. Ossanna, V. A. Vyssotsky, G. G. Ziegler

Purpose

The Multics I/O system provides capability for random linear I/O. This section describes the I/O system calls for performing random linear I/O.

Random Linear Frames

An existing linear frame may be attached to a process (or a new linear frame may be created and attached to a process) as a random frame by a call to the I/O system. A frame which already exists when it is so attached may have previously been attached to processes as either a sequential or a random frame or both (although not both at the same time). When attached as a random frame, a linear frame may be regarded as a set of elements each identified by an element number. An element number is represented in I/O calls as a 35 bit signed integer, whose value must be non-negative. Any two distinct elements of a linear frame have different element numbers. If the file is non-empty, there is a first element, which is element 1, and there is a last element, element number E. To every element of the frame there corresponds exactly one element number, and that number e is in the range $1 \leq e \leq E$. A linear frame has no gaps.

The I/O system calls for random linear I/O are formally identical to those described in section BF.1.12 for sequential linear I/O. They will be described in terms of the differences between their action on a random frame and their action on a sequential frame. One outstanding difference is that the argument elemno of the <u>read</u>, <u>write</u> and <u>seek</u> calls is interpreted differently. In each of these calls, elemno is interpreted as follows. If 1 ≤ elemno ≤ M, where M is the declared maximum element number of the frame, then elemno is the number of the element to be read, written or found. If elemno = 0, then the element involved is the current element. If elemno < 0 or if elemno > M, the call is rejected, and the current element number becomes undefined. Another significant difference is that the normal mode of a random linear frame is the <u>replacement</u> mode as opposed to the <u>truncation</u> mode for sequential linear frames (see section BF.1.04 for a discussion of modes).

The Write Call

The general form of the random write call is the same as that of the sequential write call:

```
     call write(name,elemno,workspace,nelem[,status])
```

The arguments and actions of random write are identical to those of sequential write, except for the difference in interpretation of elemno described above.

The Read Call

The general form of the random read call is the same as that of the sequential read call:

```
     call read(name,elemno,workspace,nelem[,nelmt[,status]])
```

The arguments and actions of random read are identical to those of sequential read, except for the difference in interpretation of elemno described above.

The Tell Call

The general form of the tell call for a random linear frame is the same as for a sequential linear frame:

```
          call tell(name,elemno[,status])
```

The arguments and actions of the tell call are exactly the same for random frames and for sequential frames.

The Seek Call

The general form of the random seek call is the same as that of the sequential seek call:

```
          call seek(name,elemno[,status])
```

The arguments and actions of random seek are identical to those of sequential seek, except for the difference in interpretation of elemno described above.

The Breaks Call

The general form of the breaks call for random linear frames is the same as for sequential linear frames:

```
     call breaks(name,breakptr,nbrks[,status])
```

The arguments and actions of the breaks call are exactly the same for random frames as for sequential frames.

The Delete Call

The general form of the random delete call is the same as that of the sequential delete call:

```
call delete(name,elemno[,status])
```

The arguments and actions of random delete are identical to those of sequential delete, except for the difference in interpretation of elemno described above. The action of random delete is exactly the same as that of a random write with nelem = 0.

The First and Tail Calls

The general form of the first and tail calls for random frames is the same as for sequential frames:

```
call first(name[,status])
call tail(name[,status])
```

The arguments and actions of the first and tail calls for random frames are the same as for sequential frames.