Published: 11/17/67

Identification

Resetting Asynchronous User Ionames
reset_user_in, reset_user_out
W. R. Strickler

Purpose

When it is determined that I/O data read ahead or written behind are, for some reason, no longer useful, a call to either resetread or resetwrite is necessary to discard the data (see BF.1.04). Procedure reset_user_in provides the I/O caller with the facility to discard read-ahead data for the ioname "user_input" and, should the discard fail, to inform the user by means of a message in the error segment. Procedure reset_user_out provides a similar facility for data written behind on ioname "user_output".

Usage

```
call reset_user_in;

call reset_user_out;
```

Implementation

Procedure reset_user_in:

1. Calls resetread (BF.1.04) for the ioname "user_input" with optional argument status to be returned. The status bit string of the affected read call should reflect the occurrence of the resetread call.

2. Calls check_iostatus (BY.4.03) for a description of the status of the read call.

3. Checks code, returned by check_iostatus, for failure of resetread; if reset failed, seterr (BY.11.01) is called to place the I/O status description in the user's error segment, and condition reset_user_in_err is signalled.

Procedure reset_user_out takes the same steps for discarding data written behind, with "resetread" replaced by "resetwrite", of course, and "user_input" replaced by "user_output".