NAME

rje − DQS-11B interface for remote job entry

DESCRIPTION

The **rje** interface defines a special file that looks like a concatenation of Binary Synchronous Communication (BSC) text blocks. This file may be both written to and read from, but not simultaneously. Data transfer with the two-point BSC discipline is strictly half-duplex.

The device can be opened by only one process at a time. It is expected that a process that successfully opens the DQS will spawn separate subprocesses to handle reading and writing. However, no distinction is made among the several processes that may have the DQS open. For example, reads within a message, even from a single block, may be executed by several processes in sequence. The overriding constraint is that a complete message must be read from or written to the DQS before any transfer of data in the opposite direction can begin. A process that tries to write while the DQS is reading, or vice versa, will be put to sleep until the transfer of the currently active message has been completed.

A complete message consists of one or more text blocks. A message being written to the DQS is terminated by a write of zero bytes, which causes an EOT to be transmitted. A message being read from the DQS is terminated by the reception of an EOT (which is not passed on to the reader, but is registered as a read of zero bytes). By convention, an EOT follows each block which ends in an ETX.

The length of a text block cannot exceed 512 bytes, including the line prefix and appendix. These two sequences, which must be present in blocks being written and will be passed on in blocks read, are constructed from the control bytes SOH, STX, ETB, ETX, DLE. The DQS itself will supply leading SYN bytes and trailing block check and pad bytes. The interface examines only the last byte of each text block received and so is unaware of the presence of headings or transparent text. The selection and interpretation of these features is the user's responsibility.

Line control functions, such as the alternating affirmative responses (ACK0 and ACK1), are automatically interspersed with text blocks as required by the line discipline. The interface handles the initial line bid and the EOT reset at the end of a transmission. A 3-second time-out is also respected. The interface will send TTD's and respond WACK's if its buffers are not serviced fast enough. When receiving, expiration of the time-out will cause the interface to abort the active message by sending EOT. When transmitting, the failure to send a block successfully after seven tries will cause the interface to terminate the active message prematurely. Such aborts cannot be appealed.

Reads on the DQS will return bytes from a single text block. If one read does not exhaust a text block, successive reads will return additional bytes from the same block. A returned count of zero indicates the end of a message. Until the remote station bids for the line, all reads will return zero bytes. The error bit will never be set by the interface itself. The DQS must be read to the end of a message before it will accept writes.

Writes to the DQS must consist of a single, entire text block. A write that specifies a count of zero bytes defines the end of a message. The count returned by a write call must be checked. A count of zero for the first write of a new message indicates that it was not possible to acquire the line. Otherwise, the DQS should return exactly the count specified in the write call. However, the error bit is set when a line error requires that the message be aborted. Notification of the error is not punctual, because data blocks are buffered for transmission. A write of zero bytes must be issued, or an error must occur, before the DQS will accept reads.

An open returns with the error bit set if the DQS is already open or not ready. The DQS should be opened in mode 2 to allow both reading and writing.

The DQS interface steals a number of buffers from UNIX (currently two) for the duration of each message. This number is specified at system generation time and may be tuned to influence overall system throughput.

FILES

/dev/rjei DQS11-B communicating with IBM 370

SEE ALSO

General Information−Binary Synchronous Communication, IBM Systems Reference Library #GA27-3004.
DQS11-A/B PDP-11 Communications Controller Option Description, Digital Equipment Corporation.