Network Working Group
Request for Comments: 121
NIC: 5833

M. Krilanovich
Computer Research Lab, UCSB

21 April 1971

NETWORK ON-LINE OPERATORS

PREFACE

The operators described in this document have been implemented within UCSB’s On-Line System and make the Network (via the NCP) accessible to On-Line System users.

A set of operators is provided to facilitate the use of the Network by On-Line System users. The operators are defined on LVL I on system ’NET’, and serve an an interface between the users and the Network Control Program (NCP), which supervises all Network operations at this site.

A concept fundamental to On-Line System Network operations is that of a completion code variable. Associated with each socket that is not in the closed state is a unique variable, called a completion code variable. This variable serves two purpose: it identifies the local socket referenced, and upon completion of the operation it contains a completion code to indicate the outcome of the operation to the user. It may be used at any time for any purpose other than Network operations. Its value at the beginning of a Network operation is immaterial rather it is the variable itself that is important. In all Network operations, whenever a completion code variable is called for, the only acceptable type variable is a LO storage location, either case 1 or case 2.

In those operations requiring a socket to be specified by its socket identifier, the following format is used: the site number of UCSB is always used for local sockets, and so this item is never specified. The site number of a foreign socket, if specified, must be a positive integer or a LO storage location, and must be less then 256 in value. If not specified, the site number of a foreign socket will default to the site number of UCSB. A socket number is specified as an integer, either unsigned, in which case it is assumed to be positive, or with an explicit minus sign, or a LO storage location, of any value. Negative socket numbers are represented internally in two’s complement form and therefore for the purpose of the gender of a socket, a socket number is even or odd according to whether its absolute value is even or odd, respectively.

In all operations involving the On-Line System Network operators, there are certain conventions concerning the format of the data sent and received. The data is grouped in ’messages’ consisting of three fields, op code, length, an text, in that order. The op code is one byte in length and is a code which indicates how the text field is to be interpreted. The length field is two bytes long, and gives the length, in bits, of the text field, which contains the actual data. (The op code and length fields together are termed a header.)

The following op codes are presently defined:

op code meaning

0 This op code is a NOP. No text field exists, and the contents of the length field are unpredictable. (This op code is used mainly as a delimiter.)

1 The text field contains EBCDIC characters, one character per byte. The On-Line System operators consider the characters as intended for display as soon as the text field has been received.

2 The text contains codes for keypushes, one byte per key. The On-Line operators consider the keys as intended for execution as soon as the text field has been received.

3 The same as for an op code of 2, except that the On-Line operators consider that the execution of the keys will be delayed until all data for that receive operation has been received.

The standard format of data sent or received by the On-Line System operators is a string of messages, with the last message indicated by a header with a NOP op code, called a trailer. These conventions are the default situation; any of them may be overridden by appropriate programming. Following is a description of each operator, its function and key sequence.

(1) OPEN

By invoking this operator, the user requests that the specified local socket be removed from the closed state and thus be prepared to participate in data transfer. There are three distinct operations that can be performed by this operator and these are described below:

(A) CONNECT

By invoking this operator, the user requests that the specified local socket be logically connected to the specified foreign socket, and that transfer of data be enabled between the two sockets. Processing at the terminal is suspended until the foreign sockets either accepts or rejects the connection attempt. This operation is valid only when the local socket is in the closed state.

The key sequence is as follows:

```
UP Q1 Q2,Q3[,Q4] RETURN
```

where

Q1 is the completion code variable to be associated with the local socket.
Q2 is the socket number of the local socket.
Q3 is the socket number of the foreign socket.
Q4 is the site number of the foreign socket.

The following completion codes are possible for this operation:

0, The operation was successful and the connection has been established
4, The specified completion code variable is already assigned to a socket; the operation was suppressed
8, The specified local socket is not in the closed state; the operation was suppressed
12, All communication paths to the specified foreign site are in use; the operator was suppressed
16, Local resources are insufficient to support another connection; the operation was suppressed
20, The connection attempt was rejected by the foreign process
28, The specified foreign site is invalid; the operation was suppressed
36, Either hardware at the foreign site is inoperative or the NCP’s counter-part at the foreign site does not exist or has failed
40, Local and foreign sockets are both either send or receive sockets; the operation was suppressed
44, By operator command, all Network operations were terminated; the socket will be closed
60, An NCP control transmission error occurred; the operation as suppressed

(B) LISTEN

This operation is a request for notification of any connection attempt directed toward the specified local socket. Processing at the terminal is suspended until such a call is received. This operation is valid only when the designated local socket is in the closed state.

The key sequence is as follows:

```
UP Q1 Q2 RETURN
```

where

Q1 is the completion code variable to be associated with the local socket.
Q2 is the socket number of the local socket.

The following completion codes are possible for this operation:

```
0, The operation was successful and a call has been received
4, The specified completion code variable is already
          assigned to a socket; the operation was suppressed
8, The specified local socket is not in the closed state;
          the operation was suppressed
12, Local resource are insufficient to support another
          connection; the operation was suppressed
44, By operator command, all Network operations were
          terminated; the socket will be closed
```

(C) ACCEPT

This operation accepts connection with the foreign socket whose call caused successful completion of a previous LISTEN operation by the specified local socket. After completion of this operation, data may be transferred to or from the local socket, depending on its gender. This operation is valid only when the last operation referencing the local socket as a LISTEN operation. Processing at the terminal is suspended until the operation is completed.

The key sequence is as follows:

```
UP Q RETURN
```

where

Q is the completion code variable associated with the local socket by a previous LISTEN operation.

The following completion codes are possible for this operation:

0, The operation as successful, and the connection is established
4, The specified local socket is in the closed state (the NCP may have received an abort notification from the foreign process); the operation was suppressed
8, The previous operation specifying the designated local socket was not a LISTEN; the operation was suppressed
12, All communication paths to the specified foreign site are in use; the socket has been returned to the closed state

(2) CLOSE

This operation is a request that the specified local socket be returned to the closed state. If the last operation involving this socket was a LISTEN, this operation refuses the foreign process connection attempt. If the last operation was a CONNECT, the attempt is aborted. If a connection is established, any data in transit form the local socket is allowed to reach the foreign socket and to be either received or flushed before the local socket is closed.

Processing at the terminal is suspended until the socket has been returned to the closed state.

The key sequence is as follows:

```
DWN Q
```

where

Q is the completion code variable associated with the local socket.

The following completion codes are possible for this operation:

0, The operation was successful, and the socket has been returned to the closed state
8, The specified completion code variable is not currently assigned to a socket; the operation was suppressed
12, The specified local socket is in the process of being closed the operation was suppressed, but the local socket will be closed
36, Either hardware at the foreign site is inoperative, or the NCP’s counterpart at the foreign site does not exist or has failed
44, By operator command, all Network operations were terminated; the socket will be closed
60, An NCP control transmission error occurred; the operation was aborted
64, A transmission error occurred the operation was aborted, but the socket will be closed

(3) SEND

This operation caused data to be sent to the foreign socket. Processing at the terminal is suspended until the data has been received by the foreign socket, or until it has been queued locally by the NCP.

The possible key sequence are as follows:

```
STORE  Q1 Q2 text Q2
STORE +Q1 Q2 text Q2
STORE -Q1 Q2 text Q2
STORE .Q1[X]Q3 RETURN
```

where

Q1 is the completion code variable associated with the local socket.
Q2 is any key except RESET.
’Text’ is a string of any length of any keys except Q2 and RESET.
Q3 is an unsigned integer. If it is preceded by ’X’, the

integer is assumed to be in hexadecimal.
[X]Q3 should have a value less then 256.

If Q1 is the first key after STORE, the keys comprising ’text’ are not executed but rather are sent as a one byte code per key, in one or more standard messages. They are preceded by a header with the op code which indicates keys to be executed as soon as they are received, and followed by a trailer. If a ’+’ precedes Q1, the headers and the text will be sent, but no trailer. If a ’-’ precedes Q1, only the text will be sent. If a ’.’ precedes Q1, [X]Q3 will be sent as a one byte number, with no header or trailer.

The following completion codes are possible for this operation:

0, The operation was successful, and the data has been sent
4, The specified local socket is not a send socket; the operation was suppressed
8, The specified completion code variable is not assigned to a socket; the operation was suppressed
12, A previous send operation is in progress ;the operation was suppressed
16, The connection is not fully open; the operation was suppressed
20, The foreign socket terminated the connection before completion of the send operation; not all the data was transmitted
36, Either hardware at the foreign site is inoperative, or the NCP’s counterpart at the foreign site does not exist or has failed
44, By operator command, all Network operations were terminated the socket will be closed
52, One or more interrupts were received from the foreign socket; the operation was suppressed
60, An NCP control transmission error occurred; the operation was aborted and the socket will be closed
64, A transmission error occurred; the operation was aborted and the socket will be closed

(4) RECEIVE

This operation causes data to be received from the foreign socket. Processing at the terminal is suspended until the data sent by the foreign socket has been received by the local socket and processed.

The possible key sequences are as follows:

```
LOAD  Q1
LOAD +Q1
LOAD -Q1 Q2, Q3 RETURN
LOAD .Q1 Q4 RETURN
```

where

Q1 is the completion code variable associated with the local socket.
Q2 is a positive integer, or a LO storage location, less then 256.
Q3 is a positive integer, or a LO storage location, less than 65536
Q4 is a positive integer less than 5.

If Q1 is the first key after LOAD, one or more standard messages will be received and executed. The data may or may not be executed, as soon as it is received, depending on the op code. Processing of manual keypushes is suspended until a trailer is received. If Q1 is preceded by a '+', the operation is as above, except that only one message, with no trailer, is received, and the operation is only performed if there is data queued for the local socket, ready to be received. ('LOAD +' is intended mainly for use with the Network On-Line Interface.) If Q1 is preceded by a '-', a string of text whose length, in bits, is Q3, is received and processed according to the op code Q2. If a '.' precedes Q1, Q4 bytes of data are read and placed, right justified, in the LO accumulator.

The following completion codes are possible for this operation:

0, The operation was successful and the data has been received
4, The specified local socket is not a receive socket; the operation was suppressed
8, The specified completion code variable is not assigned to a socket; the operation was suppressed
12, A previous receive operation is in progres; the operation was suppressed
16, The connection is not fully open; the operation was suppressed
20, The foreign socket terminated the connection before completion of the receive operation; data is unpredictable
36, Either hardware at the foreign site is inoperative, or the NCP's counterpart at the foreign site does not exist or has failed
44, By operator command, all Network operations were

```
         terminated; the socket will be closed
      52, One or more interrupts were received from the foreign
         socket; the operation was suppressed
      60, An NCP control transmission error occurred; the
         operation was suppressed and the socket will be closed
      200, There was no data queued for the local socket; the
         operation was suppressed (applicable only to 'LOAD +')
```

(5) CHECK

This operation causes the status of the specified local socket to be displayed. There is no completion code variable associated with this operation, since it is always successful and the socket is identified by its socket number. This operation is valid at any time, and with one exception, noted below, is always completed immediately.

The possible key sequences are as follows:

```
DISPLAY (Q RETURN...)...
DISPLAY (Q?)...
DISPLAY RETURN...
```

where

Q is the socket number of the local socket whose status is to be displayed.

Successive RETURN'S will display the status of local sockets with successively higher socket numbers, and once DISPLAY has been pushed, it need not be pushed for subsequent occurrences of Q.

If Q is followed by a '?', processing at the terminal will be suspended until the socket is not in the closed state, at which time the status will be displayed. If the socket is not closed at the time the '?' is pushed, response is as if the '?' were a RETURN.

If DISPLAY is followed immediately by a RETURN, the status of all non-closed sockets owned by the terminal is displayed, each preceded by its associated completion code variable. If this information is more than will fit on one tube, successive RETURN'S will display the remainder of the information.

The status has the following format:

'lclno' 'mnem' 'fgnsite' 'fgnno' 'deficit'

where

'lclno' is the socket number of the local socket.
'mnem' is a mnemonic of up to eight characters for the status.
'fgnsite' is the site number of the foreign socket, or blank.
'fgnno' is the socket number of the foreign socket, or blank.
'deficit' is the send/receive deficit in the form 'bytes.bits', or blank.

Following are the possible mnemonics, together with their interpretations:

'MNEM'	MEANING
OPEN	A connection is fully established. No SEND/RECEIVE operation is in progress. 'fgnsite' and 'fgnno' are the site number and the socket number of the connected socket. 'deficit' is the number of bits queued locally at the socket by the NCP and available to satisfy a future RECEIVE operation, or awaiting output as the result of a previous SEND operation.
LISTEN	A LISTEN has been issued.
CONNECT	A CONNECT has been issued. 'fngsite' and 'fgnno' are the site and socket numbers of the foreign socket.
DECISION	A LISTEN has been completed. 'fgnsite' and 'fgnno' are the site and socket numbers of the calling socket.
CALL(S)	One or more calls have been received for the local socket. No LISTEN or CONNECT has been issued.
I/O	A connection is fully established. A SEND/RECEIVE operation is in progress. 'fgnsite'

and ’fgnno’ are the site and socket numbers of the connected socket. ’deficit’ is the number of bits yet to be sent or received.

CLOSED — The socket is in the closed state.

<--DRAIN — The foreign socket is attempting to close the connection. The NCP has data yet to be read by the local socket. ’fgnsite’ and ’fgnno’ are the site and socket numbers of the connected socket. ’deficit’ is the number of bits yet to be received.

DRAINED — The foreign socket is attempting to close the connection. The NCP is awaiting arrival at the foreign site of data currently in transit. ’fgnsite’ and ’fgnno’ are the site and socket numbers of the connected socket.

CLOSING — The local socket has issued a CLOSE. The NCP is in the process of returning the local socket to the closed state. ’fgnsite’ and ’fgnno’ are the site and socket numbers of the connected socket.

DRAIN--> — The local socket has issued a CLOSE. The NCP is completing the last SEND operation before returning the local socket to the closed state. ’fgnsite’ and ’fgnno’ are the site and socket numbers of the connected socket. ’deficit’ is the number of bits the NCP has yet to transmit.

(6) IDENTIFY

This operation is used to identify a local socket by its completion code variable. The operation is valid at any time, and is always completed immediately. Since it is always successful, there are no completion codes for the operation, and the contents of the completion code variable are not changed.

The key sequence is:

```
ID Q...
```

where

Q is the completion code variable associated with the local socket.

The response is a display of Q, followed by 'IS UNDEFINED' if the socket is in the closed state, or the socket number associated with it if it is not closed. After ID has been pushed once, it need not be pushed again for subsequent occurrences of Q.

(7) PURGE

This operator is used to dissociate all completion code variables from sockets, and to issue a CLOSE to all sockets currently in use at the terminal. The operation is valid at any time, and is always completed immediately. Since it is always successful. there are no completion codes for the operation, and the contents of the completion code variables are not changed.

The key sequence is:

```
DEL RETURN
```

(8) SIGNAL

This operation is used to convey a signal to the foreign process. The significance of the signal is completely user-dependent. The effect is that the next time the foreign socket attempts to initiate a RECEIVE or SEND operation, the operation will be suppressed, and a completion code supplied indicating that a signal had been received. Processing at the terminal is suspended until the signal has been sent to the foreign NCP. This operation is valid only when the socket is fully open.

The key sequence is:

```
REFL Q
```

where

Q is the completion code variable associated with the local socket.

The following completion codes are possible for this operation:

0, The operation was successful, and the signal has been sent
4, The specified completion code variable is not assigned to a socket; the operation was suppressed
8, The connection is not fully open;the operation was suppressed

(9) CONVERSATIONAL CONTROLLER

This operator is intended primarily for use with the Network On-Line Interface, and is used to carry on a 'conversation' with the foreign process. Any data queued locally for the specified local receive socket is received and displayed as characters. Any keys pushed will be sent from the specified local send socket. The user has the option of specifying whether or not upper keyboard buttons are to be sent. In any case, the buttons ERASE, REPEAT, ENTER, PRED, and RESET, will not be sent, and pushing any of these, except RESET, will cause suspension of data reception until a button other than these four is pushed. Pushing RESET always terminates the operation.

The key sequence is as follows:

```
ARG [,] [-]Q1[-]Q2
```

where

Q1 is the completion code variable associated with the local send socket.
Q2 is the completion code variable associated with the local receive socket.

If ARG is followed by ',', both upper and lower keyboard buttons will be sent, and the operation will be terminated only by pushing RESET. If no ',' follows ARG, only lower keyboard buttons will be sent, and the operation is terminated by pushing RESET or any upper keyboard button. If a '-' precedes Q1, no headers will be sent, and if a '-' precedes Q2, no headers will be received. In any case, if headers are to be received, the operation is terminated when a trailer is received, and a corresponding trailer is sent if headers are to be sent.

The completion codes for the send socket are the same as for the SEND operation, and the completion codes for the receive socket are the same as for the RECEIVE operation. If a non-zero completion code is found, a message is displayed, either 'OUTPUT ERROR' or 'INPUT ERROR', depending on whether the send socket for the receive socket encountered the non-zero completion code, and the operation is terminated.

[This RFC was put into machine readable form for entry]
[into the online RFC archives by Gruss Gottfried 01/98]