

1 November 1973

To: Systems Group
From: James Harp
Re: IPC Runtime Routine: User Documentation

Refer to Operation System Notes (OS Note 8/31, sec. 10) for specifications of the interprocess communication runtime routine. The routines are:

Sending message:

BEGIN\MESSAGE
SEND\SCALAR
SEND\DESCRIPTOR
SEND\VECTOR
SEND\STRING
SEND\CAPABILITY
SEND\MESSAGE

Receiving message:

AWAIT\MESSAGE
RECEIVE\SCALAR
RECEIVE\DESCRIPTOR
RECEIVE\VECTOR
RECEIVE\STRING
RECEIVE\CAPABILITY

2

BEGIN\MESSAGE: Signifies beginning of a new message.

BEGIN\MESSAGE ()

Description:

BEGIN\MESSAGE informs the runtime that a new message is about to be assembled and sent.

Fail Action:

None.

Example:

BEGIN\MESSAGE ()

Date: 10/12/73

Author: James Harp

SEND\SCALAR: Places a scalar in message.

SEND\SCALAR (SCALAR S)

Description:

SEND\SCALAR appends the argument scalar to the message.

Fail Action:

SEND\SCALAR fails if the scalar will not fit in the message.

Example:

SEND\SCALAR (S)

Date: 10/2/73

Author: James Harp

4

SEND\DESCRIPTOR: Places a descriptor in message.

SEND\DESCRIPTOR (DESCRIPTOR DS)

Description:

SEND\DESCRIPTOR appends the two-word argument descriptor to the message.

Fail Action:

SEND\DESCRIPTOR fails if the two-word argument descriptor will not fit in the message.

Example:

SEND\DESCRIPTOR (DS)

Date: 10/2/73

Author: James Harp

SEND\VECTOR: Places a vector in message.

SEND\VECTOR (VECTOR V)

Description:

SEND\VECTOR appends the entire argument vector to the message.

Fail Action:

SEND\VECTOR fails if the vector will not fit in the message.

Example:

SEND\VECTOR (V)

Date: 10/11/73

Author: James Harp

SEND\STRING: Places a string in message.

SEND\STRING (STRING S)

Description:

SEND\STRING appends the contents of the string of scalars (as specified by the selector) to the message.

Fail Action:

SEND\STRING fails if the contents of the string of scalars will not fit in the message.

Example:

SEND\STRING (S)

Date: 10/12/73

Author: James Harp

SEND\CAPABILITY: Places a capability in message.

SEND\CAPABILITY (SCALAR INDEX)

Description:

SEND\CAPABILITY causes the capability which is contained at the specified index (at the time the SEND\MESSAGE call is made) to be appended to the message. Capabilities are appended in a separate queue from other type of items.

Fail Action:

SEND\CAPABILITY fails if the capability index will not fit in the message.

Example:

SEND\CAPABILITY (EXTERNAL\SEGMENT)

Date: 10/12/73

Author: James Harp

SEND\MESSAGE: Sends assembled message.

SEND\MESSAGE (SCALAR PORT\CAPABILITY)

Description:

SEND\MESSAGE causes the assembled message to be sent to the specified destination port.

Fail Action:

None.

Example:

SEND\MESSAGE (PORT\CAPABILITY)

Date: 10/12/73

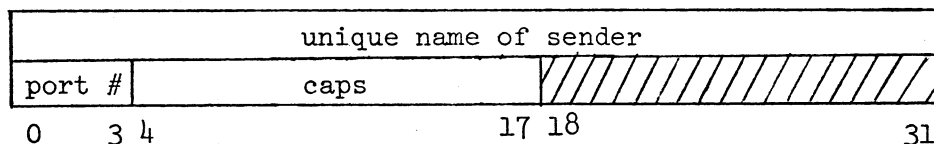
Author: James Harp

AWAIT\MESSAGE: Awaits first message to any specified port.

AWAIT\MESSAGE (SCALAR PORT\MASK) RETURNING DESCRIPTOR

Description:

AWAIT\MESSAGE awaits the first message which is sent to any of the ports specified in the mask. Furthermore, it returns as value a two word "header" for the message which is formatted as:



where "port" specifies which port the message was received on; "caps" specifies how many capabilities came in the message.

Fail Action:

None.

Example:

AWAIT\MESSAGE (PORT\MASK)

Date: 10/12/73

Author: James Harp

RECEIVE\SCALAR: Gets a scalar from message.

RECEIVE\SCALAR () RETURNING SCALAR

Description:

RECEIVE\SCALAR removes a scalar from the message and returns it as value.

Fail Action:

RECEIVE\SCALAR fails if there is not a scalar in the message.

Example:

S ← RECEIVE\SCALAR ()

Date: 10/2/73

Author: James Harp

RECEIVE\DESCRIPTOR: Gets a descriptor from message.

RECEIVE\DESCRIPTOR () RETURNING DESCRIPTOR

Description:

RECEIVE\DESCRIPTOR removes a two-word descriptor from the message and returns it as value.

Fail Action:

RECEIVE\DESCRIPTOR fails if there is not a descriptor in the message.

Example:

D ←RECEIVE\DESCRIPTOR ()

Date: 10/2/73

Author: James Harp

RECEIVE\VECTOR: Gets a vector from message.

RECEIVE\VECTOR () RETURNING VECTOR

Description:

RECEIVE\VECTOR interprets the next item on the message as a vector and returns a vector descriptor which locates the vector. The data located by the vector descriptor is valid up to the point that a new message is received.

Fail Action:

RECEIVE\VECTOR fails if there is not a vector in the message.

Example:

V ←RECEIVE\VECTOR ()

Date: 10/11/73

Author: James Harp

RECEIVE\STRING: Gets a string from message.

RECEIVE\STRING (STRING S)

Description:

RECEIVE\STRING interprets the next item as a string, and the argument string is set up to locate the sent string. This data is also valid only until the next message is received.

Fail Action:

RECEIVE\STRING fails if there is not a string in the message.

Example:

RECEIVE\STRING (S)

Date: 10/12/73

Author: James Harp

RECEIVE\CAPABILITY: Gets a capability from message.

RECEIVE\CAPABILITY (SCALAR INDEX)

Description:

RECEIVE\CAPABILITY copies the next unreceived capability which was sent with the message to the location in the process capability segment specified by the scalar argument index.

Fail Action:

RECEIVE\CAPABILITY fails if there is not a capability in the message.

Example:

RECEIVE\CAPABILITY (EXTERNAL\SEGMENT)

Date: 10/12/73

Author: James Harp