Bell Telephone Laboratories, Incorporated PROGRAM APPLICATION INSTRUCTION

- 1 -

PA-1C600-01 Section 12 (a) Issue 1, 10/1/77 AT&TCo SPCS

SENDCPMSG(a)

SENDCPMSG(a)

NAME

sendcpmsg - send a capability message

SYNOPSIS

(sendcpmsg = 51.)
sendcpmsg(msgbuf)
int *msgbuf; /* pointer to message */

DESCRIPTION

Sendcpmsg sends a message from the current process to another process (kernel or supervisoruser type). The message to be sent starts at *msgbuf* and may be up to 112 words long. A message consists of a six word header defined by the following structure:

```
struct msghdr {
```

int	*mslink;	/* pointer to next input message */
int	msfrom;	/* sending process number */
int	msto;	/* receiving process number */
char	mssize;	/* message size in words */
char	mstype;	/* message type */
int	msident;	/* message identifier */
char	msstat;	/* message status word */
char	msseqnum;	/* message sequence number */

};

followed by three words of capability:

struct cp clist {

int	cpm_num;	/* capability entry in PCB */
int	cpm_owner;	/* capability owner */
int	cpm_cap;	/* capability */

};

and then the sender's data. The sender need only fill in the capability number cpm_num; the kernel sendcpmsg EMT then fills in the capability owner and the value of the capability in the other two words of the structure from the process PCB before sending the message onto the msto process. If the capability specified is bad or none is specified, the message is sent as a normal message. Otherwise the capability bit is set in the message header. It is the receiver's responsibility to validate the capability. The sender must also fill in the msto, mstype and mssize fields of the message. The mssize word is the size of the sender's message in words exclusive of the header. The *mstype* byte may be any number from 0 to 0376. The value of 0377 is reserved for acknowledgement messages. The sender may fill in *msident* in order to identify a particular acknowledgement message, as this word is never modified during the life of this message. The message is verified and copied into a kernel address space message buffer area. Here the msfrom word is filled in by the kernel as well as the message sequence number. The message is put on the input queue of messages for the msto process using the mslink word. A programmed interrupt request is enabled by sending a message event to the *msto* process. The message sequence number *msseqnum* is used only for debugging purposes. The *msstat* byte is filled in by the receiver of this message in its acknowledgement to this message. It contains the error code if non-zero. The value of -1 is reserved by the system for the case where the intended receiver process does not exist or is aborted abnormally.

If the input message queue for the receiver is overloaded or no message buffers exist in the kernel message buffer pool area, an error is passed back to the library routine which then road-blocks the process. Normally a value of 1 is returned from C.

Bell Telephone Laboratories, Incorporated PROGRAM APPLICATION INSTRUCTION

SENDCPMSG(a)

SEE ALSO

sndmsgfrom(a), getmsg(a), gettype(a), sendmsg(a), intro(c).

DIAGNOSTICS

If the message is too big, a bad EMT is indicated by a fault code of 10.

/* sending process number investores /* sending process number */ /* message size in words */ /* message rive */ /* stessage identifier */ /* message status word */ /* message somence mumber */

and the sender's data. The sender need only fa in the main mappe and more hads of the basis message. The maxime word is the size of the sender's message in words exclusive of the basis of the mayne byte may be any number from 0 to 0376. The value of 0377 is reserved far aciewledgement messages. The sender may fill in maken in order to identify a particular ackmaticipament message, as this word is never modified during the life of this message. The message is verified and copied into a kernel address space message bather area. Here the massage is verified and copied into a kernel address space message bather area. Here the message is used in by the kernel as well as the message sequence number. The message is the input gueue of messages for the main process using the maink word. A programmed intermet request is enabled by sending a message event to the maxim byte is filled in by the message of the main process using the maink word. A programmed intermet request is enabled by sending a message event to the maxim byte is filled in by the message as the reaction of the specified maximum byte is filled in by the message of the maximum for the main process. The maximize the intermet at on the input gueue of messages for the main process using the maink word. A programmed intermet request is enabled by sending a message event to the maximum byte is filled in by the message. It contains the strue code is an area for unther maximum filled in the second of the maximum byte is filled in by the maximum byte in the state of the spectral for the case where the intended retender to the case where the intended retender to the message of the maximum byte is filled in the tender of the state of the intended re-

ibe derive message queue for the receiver is everlosted or no message buffers exist in the randi message buffer pool area, an error is passed back to the fibrary routine which from road rests the bronese

(a) southan the assessment of a sectore (a).

incompany to the second of the second second by a facility of 10

SENDCPMSG(a)

Section 12 (a) Issue 1, 10/1/77 AT&TCo SPCS

PA-1C600-01

- 2 -