Bell Telephone Laboratories, Incorporated PROGRAM APPLICATION INSTRUCTION

- 1 -

**COPYSEG(a)** 

#### NAME

copyseg - make a copy of a segment

SYNOPSIS

(copyseg = 21.) copyseg(segnum, &newid, msident)

int segnum; /\* index into PCB segment table \*/ int \*newid; /\* address of new segment ID \*/ int msident;/\* message identifier word \*/

## DESCRIPTION

Copyseg makes a copy of the segment specified by segnum in memory if possible. If the in-core copy has been successful, a value of 1 is returned in C. If the in-core copy was not successful, but a copy had to be created in the swap area, a value of 0 is returned in C. In this case a message has been sent to the memory manager to make a copy. An acknowledgement message will be returned to this process upon completion of the copy with a message identifier word *msident*. This message should be waited for by the caller. In both cases the segment id of the new segment is returned in *newid*. This system function is typically used to make a copy of an existing process.

# SEE ALSO

## DIAGNOSTICS

If the segment to be copied does not exist or if space could not be allocated for the new segment, a -1 is returned from C.

Tassembly language, r0 should point to a block of two words, the first word vinch is a flag in eschedular and the second word which is the address of the synchronization flag *flar* in the filter's address space. If the address of the synchronization flag is zero, the  $f_{c}$ -ward location is POB is used. The value of the scheduler flag is < 0 for *cylad*. = 0 for *criticla* and > 0 for  $r_{c}$ .

State event interrupts are tabilited while the kornel checks flag, potential timing problems between the "base line" and asynchronous event handler parts of a supervisor process can be resolved. The type of timing problem is illustrated by the buffered I/O in the UNIX supervisor second flags over The "base line" ende will set flag to one and initiate a buffer write then call cavard flags waiting (or the I/O to complete. If the I/O manages to complete before the "base line" completes a statice (or the I/O to complete. If the I/O manages to complete before the "base line" completes a statice (or the I/O to complete. If the I/O manages to complete before the "base line" completes a statice of the covar (preemption could occur), the event handler will mark the buffer for the I/O as done and clear flag. The covar flag to a state the covar flag to a state the covar process of the covar preventing the supervisor from the blocking for an event which has already occurred.

A value of I is remined from C.

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

# **COPYSEG(a)**