INTERNAL CHANGES

SUBPROC:

new subroutine: dostk
entry pt: dostk
cm resident

Action:

Decrements the stack pointer, checks for empty stack error, and sets up registers for a call on enviorn (but does not call enviorn itself.)

Input

b7	return link

Output

x2	word 0 of current stack entry. End of path is right justified.
a2	Abs cm addr of current stack entry
a6	abs adr of P.Stack
b2	new current subprocess
b6	end of path subprocess

Registers used: a1/x1;a2/x2;b2;b4;b5;b6

The following ECS actions were modified to use dostk:

Return with parameters
Special return
Return

CAPAB:

error modifier conventions

Calls on CAPAB subroutines (getcap,putcap,arbcap) should have b3 set to the value of error modifier. New additions to CAPAB should branch to error code with b3 set in this manner also. ie., CAPAB uses the following code for processing an mot error:

```
ERRMOT  SX1     B3
        LX1     18
        SX6     E.MISCE
        SX7     E.CLMOT
        BX7     X7+X1
        EQ      E.ERROR
```

ECSACT,XTEXT

Entries in the system jump table are defined using symbols defined in ECSACT,XTEXT. Their definition appeared as follows:

```
J.SYMBOL    bss     1    .. define J.SYMBOL as cm resident
J.SYMBOL    bss     2    .. define J.SYMBOL as ecs resident
```

The new method for symbol definition is as follows:

```
            EACT    SYMBOL    .. define J.SYMBOL AS ecs
                              resident
            CACT    SYMBOL    .. define J.SYMBOL as cm
                              resident
```

SYMBOL must be 6 (six) characters or fewer.

New operations should meet the following requirement (because it leads to nice documentation):

The entry point name should be 5 (five) or fewer characters. thus we can have:

```
            EACT    SYMBOL
            ECSCODE J.SYMBOL,SYMBOL
C.SYMBOL    SET     LAST+1
```

for jump table symbol definition, ecs initialization, and CPNAMES,XTEXT definition where SYMBOL is the entry point name