

NEW FACILITIES AVAILABLE FROM COMMAND PROCESSOR 6/6/71

ASK USER

an operation available to a subprocess. Opercl name CP:ASKU .
no parameters. expects return authorization capability count of 1.
Types a double quote on tty. (no preceding cr?) Expects user to
type a ~~XXXXX~~ name of an object using command processor standard
nameing conventions. Uses the command processor standard scan list
for implicit scan list. ~~XXXXXXXX~~ Reports all errors encountered directly
to the user tty. This includes panics. (Name typed at tty ends
with cr, of course.) Returns the named object to the calling
subprocess.

FIND

An operation available to a subprocess. Opercl name CP:FIND .
~~XXXXXXXX~~ 2 parameters. First is a capability parameter, expects to
find a scan list. Second is a block data. Expects to find a text ~~XXX~~
~~XXXX~~ nme in same form as a line passed to tty line collector. (first
word of the block data is a character count, succeeding words ascii
text. If the character count ~~XXXXX~~ exceeds the characters available in
the words passed, garbage may result. A cr is not needed at the end of the
text.) An object is looked up using command processor standard nameing
conventions, ~~XXXXXXXXXXXXXXXX~~ The scan list passed on the call is used as
the implicit scan list. ~~XXXXXXXXXXXX~~ Most errors returned to the ~~XXX~~ calling
subprocess in confused ~~XXXX~~ manner, but as errors. (I hope all errors are
returned to the calling subprocess.) If an object is successfully found,
it is returned to the calling subprocess.

Expects return authorization capability count of at least one.

maximum block data count is ?

SPACE

An operation available to a subprocess. Opercl name CP:SPAC .

One block data parameter. ~~XX~~ Expects to find 60 bit integer space requests in the same form as the space command in services. The words contain requests for space in the order; Swapped ecs, Fixed ecs, Not slots, Temporary disk. A zero is treated as no request for change, a minus 1 is treated as a request to reduce to nominal. A block data count of up to 4 may be used, routine truncates trailing space types. i.e. a block count of 1 is for swapped ecs only. The parameters are examined last first.

Not slots and Temporary disk space should be not used for now. Temporary disk space may cause system crashes.

Errors are returned to the calling subprocess in a confusing manner, but as errors. The one the subprocess may be most interested in is 'already have space of equal or higher priority' . All of these errors have the command processor error class (20B) and an error number listed in ~~XXXXX~~ CMMDP. I am not quite sure how i will publish these numbers.

A subprocess is expected to proceed as follows:

- I) have small initial map entries so as to keep swapped space needs below nominal. We hope to reduce nominal once this is being used.
- II) Large fluctuations in swapped size should be handled ~~XXX~~ by going nominal, then to the new size. Of course, map entries have to be zeroed during this.
- III) Large subprocesses, such as Scope, which have a command ~~XXXXXXXX~~ mode should go small while processing the next command. The line collector may eventually have a call that will freturn if no line is immediatly available, then these subprocesses could go small only if they have to wait for a line.

ABILITY PARAMETERS PASSED ON INITIAL CALL

If the bottom 18 bits ~~XX~~ of the top 30 bits of the word address zero in the subprocess descriptor is non zero, It is assumed that this ~~XX~~ subprocess expects capability parameters. All objects placed in the ~~XXXXXXXX~~ scanlist for the subprocess at creation time will be shifted to higher indexes by the amount in those 18 bits. The subprocess call operation constructed for the subprocess will have one block capability parameter with count equal to ~~XXXXXX~~ the value of those 18 bits.

The command processor ~~XXXXXXXXXXXXXX~~ will interpret the parameters on the command line not as bead ~~XXXXX~~ style parameters, but as objects named by the standard naming conventions. It will use ~~XXXXX~~ its normal scanlist with full permanent directory capability. The resulting objects will be passed in the block capability parameter. It will accept any number of parameters up to the size of the block capability parameter specification, or up to an internal limit of about 4.