Map (and subprocess)

A set of quadruples \(<FA, CMA, CNT, W>\)

where FA is the address of a word in file
CMA is an address in central memory
CNT is a positive integer (>0)
W is Boolean

Each subprocess has a map

With a subprocess is in the state called "running"

Then from time to time the following sequence of actions will take place:

1) a CPU or CM will be selected

i) swap-in

\(<FA, CMA, CNT, W>\)

For each quadruple in the map, CNT words will be copied from the file to central memory starting at FA to CMA. The quadruples will be treated in order.

ii) run

An XT pair will be placed in the central registers and the CPU will be allowed to run awhile

Then the XT pair will be copied out

iii) swap-out

For each quadruple \(<FA, CMA, CNT, W>\) being on the map copies from central memory to file

With \(W = \text{true}\) CNT words will be copied starting at CMA to FA. The quadruples will be treated in order.

Note: This set of actions is not an atomic act, not is easy seen in its sequence.
**Operation**

2 parts

A) list of pairs $\langle \text{type}, \text{mosn} \rangle$

b) Some kind of action description (will depend on type of operation)

To invoke an operation within a subprocess (The only way to invoke operations)

give a set $\langle z_0, \ldots, z_n \rangle$ of indices to a capability list $(C_1, \ldots, C_k)$

The following conditions must be satisfied:

- $\text{type}(C_{z_0}) \subseteq \text{operations}$  
- $\text{type}(C_{z_j}) = \text{type}_{z_j}$  
- $\text{mask}(z_{z_j}) \nleq \text{mask}_{z_j}$

\[ 1 \leq j \leq k \]

- a) 2 means any bit on on left must be on on right

- b) $\langle \text{type}_{z_j}, \text{mask}_{z_j} \rangle$ is $j$th item in list of pairs for Obj($C_{z_0}$)

Then the action

If the conditions are satisfied then the action will be taken
Capability:

a triplet <obj, transform>
Objects

files

processes

subprocesses

operations

capability bits

directory pages

allocation blocks (accounting blocks)