Subprocess descriptors

These objects are implemented as low-level disk files. They are used to direct the construction of subprocesses.

open

open merely guarantees that the descriptor will remain in ecs. It does not return a new capability.

create a subprocess

1st parameter of the creation call will be the subprocess descriptor. Subsequent parameters may be required for the construction. [This will depend on the subprocess]

This call will return a block of 1 or more objects. [The number depends on the subprocess]. The 1st object returned will be an operation for doing work on an initial call on the subprocess.
Create a subprocess descriptor

This operation requires the usual parameters for creation of a disk object that is to be represented by a low level disk file.

It also requires the presentation of a block of data which describes the subprocess, plus a block of additional parameters. [for the latter do we have sufficient ors facilities?]

The following will specify what information must be in the block of data, but not necessarily the exact form. That will be defined by the implementor.

The block of data will contain a number of sections to define various things related to the subprocess,
0) **Object descriptors**

A number of sections require the specification of particular objects. A uniform method is used for this specification. It is a word of data as follows:

<table>
<thead>
<tr>
<th>T</th>
<th>N3</th>
<th>N2</th>
<th>N1</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>5</td>
</tr>
</tbody>
</table>

(T = 1 indicates that the object is to be opened, certain objects will be opened regardless of T, i.e., files used in maps and the local C-list.)

**Class**

0. End marker for a list of objects.
1. Scratch object & N2 (See section on scratch objects).
2. A parameter N1 after subprocess descriptor in subprocess creation call.
3. Hard link, directory given as parameter N1.
4. Soft link, hard link to directory given as parameter N2.
5. Null object [used at least in C-lists]
note: 3 and 4 are supposed to mirror hard and soft links in directories, so if those ideas change, these specifications will change accordingly.

3) scratch objects

Certain objects are constructed new at the time. The subprocess is constructed. If these are disk objects they are placed in the process temporary directory with the scratch bit on. If the objects they should go into a list local to the process (?).

A list of descriptions of these objects is given as one section of the defining block of data.

Each object is described by a sequence of words.

42 18

N TYPE
<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>end of list marker</td>
</tr>
<tr>
<td>1</td>
<td>swapped file, N is shape</td>
</tr>
<tr>
<td>2</td>
<td>c-list, N is length</td>
</tr>
<tr>
<td>3</td>
<td>approved standard creator file</td>
</tr>
<tr>
<td>4</td>
<td>operation, N is number of wds belonging to object</td>
</tr>
</tbody>
</table>

Format of this description not yet defined.

Could consist of:

- seg of wds: 0 for datum
- option: list + type for object
- a class code, object description
II) Parameter description list

This section describes the parameters expected at subprocess creation time.
The list is preceded by a count of words in the list. The words in the list give the option bits and type for each parameter.

III) Basic subprocess

A list of items:

<table>
<thead>
<tr>
<th>Item</th>
<th>How given</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class code</td>
<td>object description</td>
</tr>
<tr>
<td>Class code of father</td>
<td>&quot;</td>
</tr>
<tr>
<td>Heap entries</td>
<td>an integer</td>
</tr>
<tr>
<td>Compiled maxsize</td>
<td>&quot;</td>
</tr>
<tr>
<td>Subprocess field length</td>
<td>&quot;</td>
</tr>
<tr>
<td>Entry point</td>
<td>&quot;</td>
</tr>
<tr>
<td>i-list</td>
<td>object description</td>
</tr>
</tbody>
</table>
III) Contents of map

A list of items formed as follows:

- object description [of file]
- file address
- in address
- count
- an item

(terminated by an object description of type 0, which
- file address ..., rm flag)

IV) Contents of c-list

List of object descriptors

V) Objects to return at subprocess creation time

List of object descriptors

List must be an operation to call (this subprocess)

VI) List of addresses of hard links

For the convenience of load & dump disk, a list of all

- links within this descriptor containing hard links

(constructed by system, not supplied in call)