Definitions:
 Process
      More precisely, an internal process is the execution of one or more interruptable programs in a given storage area. An internal process is identified by a unique process name. Thus other processes need not be aware of the actual location of an internal process in the store.
   external process
      the i/o of a given document identified by a unique process name.

   Peripheral device

   multiprogramming system as a set of independent, parallel processes identified by names

Communication
   Send message (receiver, message, &buffer)
   wait message (&sender, &message, &buffer) //blocking
   send answer (result, answer, buffer)
   wait answer (&result, &answer, buffer) //blocking

   advantage: secure, control by os

   Usage: inter-process communication, synchronization, system calls
        real-time synchronization using a clock external process
        disk I/O using a file external process

Disk:
   internal process creates an external process offering information of a file
   send message about reading/writing, and wait answer
   no file system introduced in this paper, but any file system can be implemented inside an external process

CPU resource management:
   processes can be dynamically created; very different from THE!
   creation: assign name, assign storage area
   start
   stop
   remove
   How to implement different scheduling algorithms?

Memory resource management:
   how to enforce memory protection among processes?
     (no virtual memory management described in the paper, but is open to VM mgmt extension)
     child process' storage has to be part of the parent
     tagged memory
     if we want to implement virtual memory, the parent process can do that

Process hierarchy
   Each parent process can implements its own resource management policies, and serves as an "OS" to all its children processes
   very similar to "library OS", "Unikernel", "Exokernel" ideas later

Comparing with THE:
   More reliable
   More extensible

More scalable
Worse performance

Questions:
1. How to implement scheduling other than round robin?
2. Page 4: is the queue served first in first out or in any order?
3. is the concept of parent process spawning children process different from fork?
4. is external process idea still around? is it just file descriptor?
5. are nucleus messages really better than THE's semaphores?