Process gives work to

One Process may create work that must be completed by other Processes. This structure is essential in determining whether a system can get into a deadlock.

CONCERNS SATISFIED: Performance and capacity.

Process gets resources from

In systems with dynamic resource allocation, one Process may control the resources used by another, where the second must request and return those resources. Because a requesting Process may request resources from several controllers, each resource may have a distinct controlling Process.

CONCERNS SATISFIED: Performance and capacity.

Process shares resources with

Two Processes may share resources such as printers, memory, or ports. If two Processes share a resource, synchronization is necessary to prevent usage conflicts. There may be distinct relations for each resource.

CONCERNS SATISFIED: Performance and capacity.

Process contained in module

Every Process is controlled by a program and, as noted earlier, every program is contained in a module. Consequently, we can consider each Process to be contained in a module.

CONCERNS SATISFIED: Changeability.

Access Structures

The data in a system may be divided into segments with the property so that if a program has access to any data in a segment, it has access to all data in that segment. Note that to simplify the description, the decomposition should use maximally sized segments by adding the condition that if two segments are accessed by the same set of programs, those two segments should be combined. The data access structure has two kinds of components, programs and segments. This relation is entitled "has access to," and is a relation between programs and segments. A system is thought to be more secure if this structure minimizes the access rights of programs and is tightly enforced.

CONCERNS SATISFIED: Security.