* **Swapping** - If too many processes exist to fit in physical memory, the system stores some,of the processes on disk. The system monitors process status changes and ensures that the highest priority processes ready to execute are in physical memory. In addition, the system will attempt not to swap an executable process out of memory, until the process has accomplished work for a minimum (configuration-specified) period of time.

Paging and swapping should be high overhead activities as long as concurrent processes do not exceed a reasonable number for physical memory, secondary storage, and the types of processing being performed. The in-memory cache for released code and data, the technique of paging each process against itself, and the assurance that ready-to execute processes are in memory combine to minimize overhead.

If desired, processes can exercise control over memory management. A real-time process, for example, could inhibit the paging and/or swapping of critical code and data.

The system maintains structures in physical memory for sharing code and data among processes, for synchronizing image execution among processes, and for communicating (sending messages) among processes. User processes create and control the structures by requesting appropriate services of the system. Nonshared structures and shared structures can be protected against access by other processes.

The system minimizes operating system overhead by taking advantage of the following VAX hardware features:

* **Context switching and queue instructions** - To schedule processes quickly and efficiently

* **Software interrupt delivery mechanism** - To minimize the processor time required to return from performing a service

* **Asynchronous system trap delivery mechanism and queue instructions** - To minimize the processor time required for I/O processing

* **Interrupt priority levels** - To improve I/O response time