Processes receive processor time for the execution of their images on an event-driven, pre-emptive priority basis. Each time an event such as an I/O interrupt occurs, the system services the event, then passes control to the highest priority process ready to execute (even if the process that had control before the event could continue to execute). Process can be assigned base priorities in the range of 0 to 31, with 4 (as a typical default) the norm time-sharing users and applications that are not time critical. The system automatically adjust priorities whose bases are in the range of 0 to 15 to favor I/O-bound process. Interactive users do not have to wait long periods while computation bound processes tie up the processor. Real-time process can be assigned higher priorities to ensure that they receive processor time whenever they are ready to execute. The system does not adjust priorities whose bases are in the range of 16 to 31, nor does it raise above 15, priorities whose bases are in the range of 0 to 15.

Processes can be started in the following ways:

* **User log-in** - When an interactive use logs in, the system creates a process for the user. The process provides an environment in which the user can communicate with the system to name images to be executed and to perform other activities.

* **Exsplicit creation** - A user can specify the creation of a new process. In creating a process, the user names an image that will be executed when the process starts.

* **Batch job** - A user can name images to be executed and other activities to be performed and can submit this information to the system as a batch job either from a terminal or a card reader.

The system queues batch jobs for execution. The user can regulate the number of queues and the number of streams per queue (that is, the number of batch jobs in the queue that can execute concurrently). Time-sharing users can enhance performance by restricting interactive use of the system to work that requires immediate responses (such as editing), by submitting as batch jobs, work that requires lengthy execution times, and by limiting batch streams to a number that is reasonable for the system configuration.

Different batch job queues may have different attributes such as the maximum CPU time permitted, working set size, and priority. Facilities are provided for started and stopping queues, as well as the jobs in a queue. Jobs in a given queue may be removed or held until a particular time.