

1) Process control management

a. The process

A process is a program which is running. Thus, there is a lot of different process in the same running in a computer. So we must manage them to improve the efficiency.

b. Process control management:

The control and management of processes depend on the scheduler. Scheduling algorithms are managing processes in Central Processing Unit (CPU). The main goal of a scheduler is to reduce time of used of the CPU and shared it between all the processes which are waiting to be runned.

c. Scheduling algorithms

The five more used scheduling are:

- First Come First Out (FCFO or FIFO: First In First Out) which run each process by order of coming and make wait all the process which are not runned in a queue.
- Shortest Job First (SJF) which take the process with the smaller CPU Burst and run them before the longest.
- Fixed priority pre-emptive scheduling which order process to run in first by its priority.
- Round Robin (RR) which put in queue all processes by order of coming. Then, it passes a pre-determined time to each process until they are not finished.
- Multilevel feedback algorithm which we will develop in detail later.

All those scheduling algorithm are pre-empted scheduling algorithm.

d. Criteria of comparison

Thus, some scheduling algorithm are more efficient in function of the use. We define the efficiency thanks to some criteria like:

- CPU utilisation. CPU must be busy as possible
- Throughput. It compts the number of process finished in a unit of time.
- Waiting time. Time before the process is runned by the CPU.
- Turnaround time. It is the time necessary to the CPU to run a process (waiting time and burst time of the process).
- Response time. It is the time between a request and the beginning of the response.

So, we compare scheduling algorithms in a table:

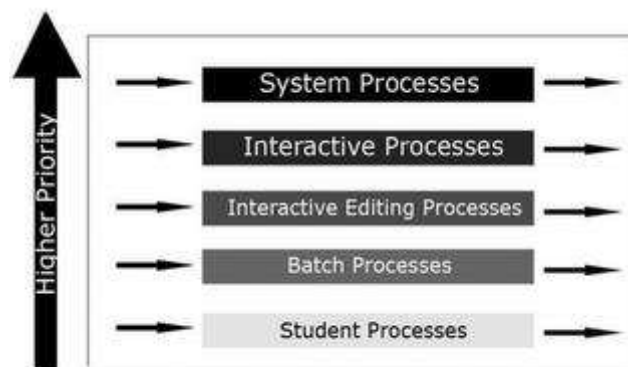
Scheduling algorithm	CPU utilisation	Throughput	Turnaround time	Response time	Starvation free
FCFS	low	Low	high	Low	Yes
SJB	Medium	High	Medium	Medium	No
Priority	Medium	Low	High	High	No
RR	high	medium	medium	High	Yes
Multilevel feedback queue	High	High	Medium	Medium	Yes

2) Process Management in Windows Seven

Microsoft's OS use multilevel feedback queue scheduling algorithm.

a. How does it work?

As we can understand in the name, there are several ready queues in this algorithm. Each is standing for some processes and depends on the priority of those processes. Thus, we have the highest priority with a queue for the System Processes and a lowest priority queue for the Student Processes. We can notice that each ready queue is using its own scheduling algorithm.



b. How a process can change of queue?

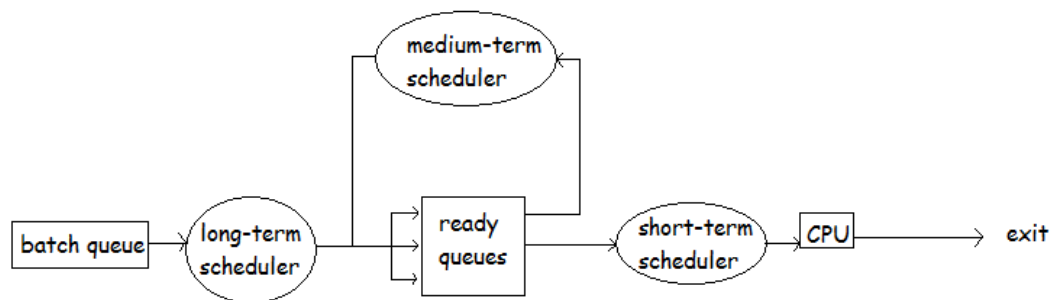
Before to be allowed in a queue, each process have to pass by the long-term scheduler. This scheduler is going to admit processes in a ready queue. It is looking for the concurrency of the process in the different queues in real time to be sure that the processes will have enough time to be runned.

Then, the medium-term scheduler is going to swap out a process from the ready queues because the process is unactive for a time, its priority is too low, it is paging fault frequently or it is taking too much memory. This sheduler will swap back, or

swap in, those processes as soon as enough memory is available or the process is unblocked.

Finally, the short-term scheduler is going to decide in the ready queues which process is going to be executed by the CPU. It is activated after a signal and it has to have done its decision before the end of the time slice.

We can summarise it in the scheme beside:



3) Conclusion

We have seen that in an operating system the number of processes and their different natures have led to a way of management on different levels. Each level is a ready queue with a level of priority. Then each queue has its own scheduling algorithm to manage process in function of their nature. Moreover, to avoid starvation or process which are not runned, the queues were implemented with schedulers to manage processes for coming in the ready queues, inside ready queues and to exit them.

4) Sources

http://en.wikipedia.org/wiki/Scheduling_%28computing%29
<http://www.personal.kent.edu/~rmuhamma/OpSystems/os.html>
<http://web.cs.wpi.edu/~cs3013/c07/lectures/Section05-Scheduling.pdf>
<http://homepages.uel.ac.uk/u0219091/scheduler.htm>
http://mekbuda.blogspot.com/2010_05_01_archive.html