先来先服务调度算法(FCFS)：

可用于作业调度，也可用于进程调度。用于作业调度时，从后备作业队列中选择一个或多个最先进入该队列作业，调入内存，为它们分配资源、创建进程，放入就绪队列。用于进程调度时，从就绪队列中选择一个最先进入该队列的进程，为之分配处理机，运行。

短作业(进程)优先调度算法(SJ(P)F)：

可用于作业调度和进程调度。用于作业调度时，从后备队列中选择一个或若干个估计运行时间最短的作业，调入内存运行。用于进程调度时，从就绪队列中选出一个估计运行时间最短的进程，将处理机分配给它，执行至完成或发生某事件而被阻塞放弃处理机。

优先级调度算法：

分为非抢占式优先级算法和抢占式优先级算法。

非抢占式优先权算法中，系统把处理机分配给就绪队列中优先权最高的进程后，该进程便一直执行下去，直至完成；或因发生某事件使该进程放弃处理机时，系统方可再将处理机重新分配给另一优先权最高的进程。抢占式优先权调度算法中，系统把处理机分配给优先权最高的进程执行，但在其执行期间，只要又出现了另一个其优先权更高的进程，进程调度程序就立即停止当前进程(原优先权最高的进程)的执行，重新将处理机分配给优先权更高的进程。

高响应比优先调度算法：

优先权为响应比：（等待时间+要求服务时间）/要求服务时间。

若作业的等待时间相同，则要求服务的时间愈短，其优先权愈高。对于长作业，作业的优先级可以随等待时间的增加而提高，当其等待时间足够长时，其优先级便可升到很高，从而也可获得处理机。

时间片轮转调度算法：

系统将所有的就绪进程按先来先服务的原则排成一个队列，每次调度时，把CPU 分配给队首进程，并令其执行一个时间片。执行的时间片用完时，调度程序停止该进程的执行，并将它送往就绪队列的末尾；然后，再把处理机分配给就绪队列中新的队首进程，同时也让它执行一个时间片。

多级反馈队列调度算法：

设置多个就绪队列，并为各个队列赋予不同的优先级。每个队列采用不同调度算法。

linux上主要有两大类调度算法，CFS(完全公平调度算法）和实时调度算法。linux调度算法实现的高层数据结构主要有运行实体、调度类、运行队列，调度实体用于调度时间记账，linux中CFS和实时调度使用不同的调度实体。调度运行队列，对于不用的调度算法同样运用不用的运行队列，对于CFS调度，运用的是红黑树，而对于实时调度为组链表。