3. REVIEW OF LITERATURE

3.1 Burst Round Robin as a Proportional Share

In this paper a new weighting technique is implemented for Round-Robin CPU scheduler as an attempt to combine the low scheduling overhead of Round Robin Algorithms and favor short jobs. Higher process weights mean relatively higher time quantum. Shorter jobs will be given more time, so that they will be removed earlier from the ready queue. This aims to achieve better throughput, and waiting time, while trying to keep the context switches as low as possible. We start by formulating a hypothesis that a process weight is inversely proportional to its CPU burst time.[3]

The advantage we gain is that processes that are close to their completion will get more chances to complete and leave the ready queue. This will reduce the number of processes in the ready queue by knocking out short jobs relatively faster in a hope to increase the throughput and reduce the average waiting time. To assign specific weights to processes, we will need to normalize this equation and find a constant for this relation. The maximum CPU burst time is set to 100tu. We tried to classify the processes into five weight categories.

Usually when a set of processes are being executed, some processes go through a series of swapping in and out, especially if the memory is full. In this paper we consider processes that are blocked due to I/O requests only. When a process is blocked, it will be moved to a waiting queue to be further processed to an I/O device. After finishing its I/O, it will then move back to the ready queue. Those blocked processes have lost their share of CPU when they block for I/O. To achieve proportional fairness,