



FACULDADES METROPOLITANAS UNIDAS  
ANALISE E DESENVOLVIMENTO DE SISTEMAS

Mateus da Silva Oliveira 2211111

ESCALONAMENTO DE PROCESSOS

São Paulo - sp

2021

O funcionamento de um computador é demandado vários métodos de se organizar um processo, dando aos usuários a opção de utilizar desses processador e encontrar a melhor solução para suas tarefas. Reproduzindo e salvando arquivos constantemente, a CPU tem a responsabilidade de calcular todos as tarefas requeridas, compondo a parte mais complexa do computador, dando assim o resultado requerido pelo usuário.

Os dados quando são processado passa por algumas etapas antes de ser calculado, ao sair do HD, entra em estado de novo onde é observado se há alguma pendência ou empecilho de que ele possa ser executados, logo depois é levado ao pronto, onde os processos são colocados em uma fila de acordo com o escalonador, depois da entra em modo de execução onde é calculado e reproduzido as tarefas, se houver algum evento que precisa da resposta do usuário, é enviado ao modo de espera, e depois desse evento ser respondido eles voltam ao pronto, o ultimo estado é o de termino onde o programa é finalizado.

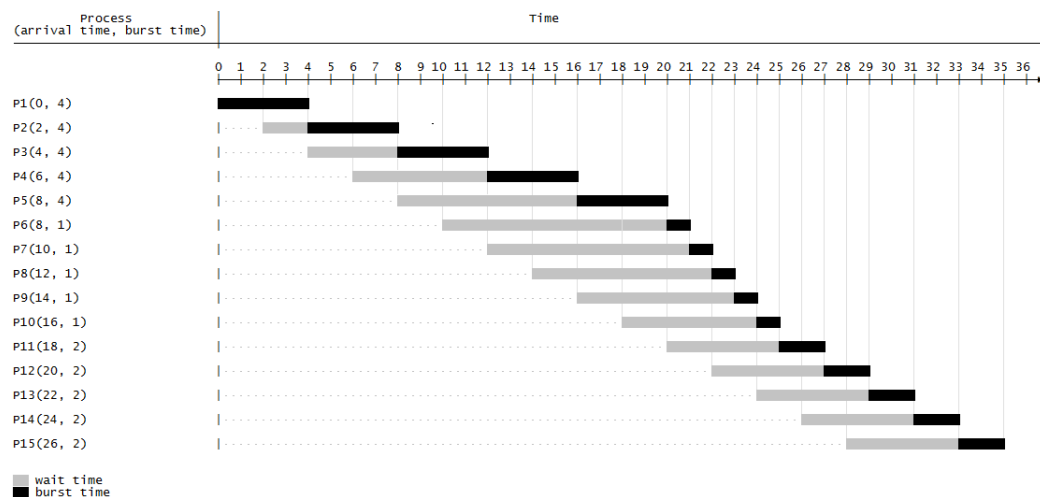


Quando o processo está em execução, pode retornar ao estado de pronto, pois como não pode ter mais de um processo no processador pode haver alguns processos mais importantes e que se esperar pode gerar alguns problemas, por isso alguns modelos de escalonamento podem permitir a interrupção de outros processos, isso é chamado de preempção, alguns modelos de escalonamento permitem a preempção outros seguem outra ordem de escolha de processos. Os modelos que permitem determinam um tempo para cada processo, esse tempo determinado, é chamado de quanton, que é uma unidade de medida determinado pelo computador que normalmente dura alguns milissegundos.

Entre os modelos de escalonamento FIFO, de prioridade, SJF e o RR, cada um deles tem seus benefícios e malefícios, e tendo os três seus papéis, e suas prioridades respectivos.

## FIFO

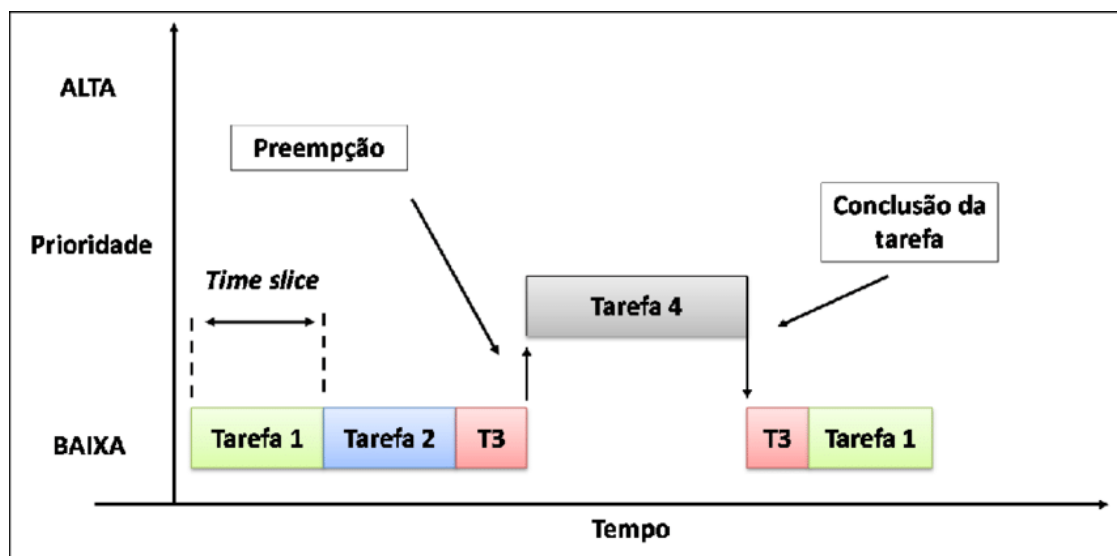
O FIFO(first in first out) também conhecido como FCFS(first come first served), segue uma linha de raciocínio simples, como o próprio nome diz, o primeiro a entrar é o primeiro a sair, o primeiro processo que chegar a fila do pronto, será o primeiro a ser processado até ser finalizado ou entrar em modo de espera dando assim espaço para o segundo processo a chegar na fila do pronto, seguindo até o ultimo processo, e pro não ter nenhum tipo de interrupção direta, esse escalonador não permite a preempção.



Ele pro ser tão fácil de entender é fácil de programar e de implementar, porem as desvantagens é que processos mais longos, vão atrasar outros processos, e também prejudicam processos que precisam passar varias vezes pelo estado de espera(I/O bound).

## PRIORIDADES

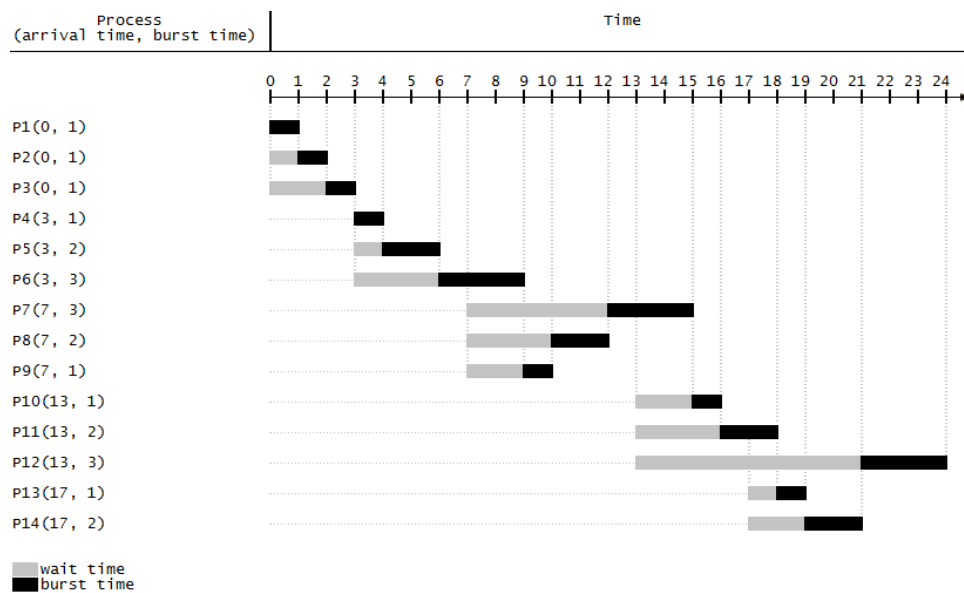
Ela segue uma a ordem de prioridade de acordo com o cada processo, cada processo vem com um certo nível de prioridades, as prioridades do sistema pode ser definido por um programador ou pelo próprio sistema, se um processo de prioridade maior do que esta sendo processado estiver em estado de pronto, poderá ter um preempção, o processo de prioridade menor é salvo e entra em estado de pronto e o de prioridade maior toma o lugar de processo.



Por definir uma ordem de parar temporariamente as tarefas menos relevantes para priorizar tarefas mais relevantes, pode deixar as tarefas com menos prioridades cada vez mais distante de ser processado, mas pode se arrumar esse problema se houver uma priorização dinâmica conforme é interrompido o processo.

## SJF

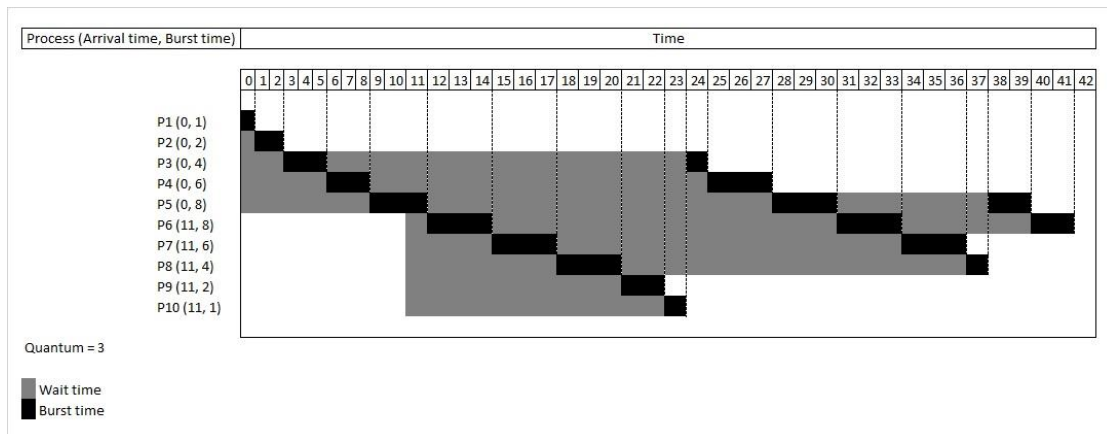
O SJF(shortest job first) como o próprio nome sugere, tem como prioridade os processos menores, trazendo uma melhor produtividade em relação aos processos I/O bound que passam mais tempo esperando um in put e out put, pois vai colocar ele em primeiro lugar assim que chegar na fila de pronto. Normalmente ela preve o tempo de acordo com o tempo do mesmo processo em experiencias anteriores.



Esse método priorizar os processos menores, ele dá uma melhor produtividade aos processos que pode terem muitas entradas e saída rodando, também por não permitir a preempção ele não interrompera os mais longos, mas ainda pode deixá-los mais dependendo pode deixa-los mais longe de ser processado.

## RR

O RR(round-robin) segue em um processamento alternado entre todos os que entrarem, ele da uma quantidade de quantons para o processo ficar na CPU e depois ele desses determinados quantons ele salva o progresso e sede o espaço para o próximo processo da fila, podendo ter como sub escalonador outros modelos, determinando qual o próximo a ser processado. O round-robin é considerado um escalonador que permite a preempção já que interrompe os processos a cada tempo determinado.



Por ter uma ordem alternada de determinados quantos, pode prolongar alguns processos que seriam rápidos, mas daria a todos os processos serem calculado todos terão um pequeno cálculo simultâneo.