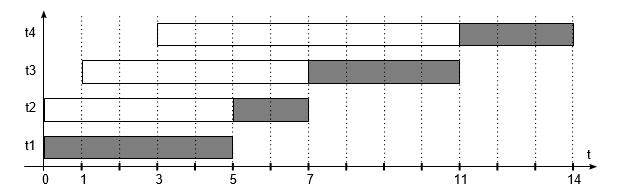
FCFS

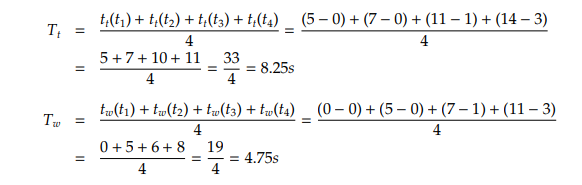
FCFS significa "primeiro que entra, primeiro que sai”, onde como seu próprio nome já diz, o primeiro que chega será o primeiro a ser executado, não-preemptivo, ou seja, executa o processo como um todo do início ao fim não interrompendo o processo executado até ser finalizado, apenas uma fila, processos que passam para o estado de pronto vão para o final da fila e são escalonados quando chegam no início. Vantagens: o mais simples entre os processos de escalonamento, até mais do que o Round-Robin, todos os processos tendem a serem atendidos. Desvantagens: muito sensível a ordem de chegada, se processos maiores chegarem primeiro aumentarão o tempo médio de espera, não garante um tempo de resposta rápido.

Novamente, o tempo de execução deverá ser previamente conhecido. Quando chega um novo job, seu tempo total é comparado ao tempo restante do processo em curso. Se para terminar, o novo job precisar de menos tempo que o processo atual, então esse será suspenso e o novo job será iniciado.

O diagrama de a figura a seguir, mostra o escalonamento do processador usando o algoritmo FCFS cooperativo (ou seja, sem *quantum* ou outras interrupções). Os quadros sombreados representam o uso do processador (observe que em cada instante apenas uma tarefa ocupa o processador). Os quadros brancos representam as tarefas que já ingressaram no sistema e estão aguardando o processador (tarefas prontas).



Calculando o tempo médio de execução (*Tt*, a média de *tt*(*ti*)) e o tempo médio de espera (*Tw*, a média de *tw*(*ti*)) para o algoritmo FCFS, temos:

****

Neste escalonamento todos os processos tendem a serem atendidos (por isso evita o fenômeno do starvation, ao menos que um processo possua um erro ou loop infinito). O loop infinito irá parar a máquina, pois com o FIFO não terá como dar continuidade a execução dos processos que estão aguardando na fila de espera.

O algoritmo FIFO não garante um tempo de resposta rápido pois é extremamente sensível a ordem de chegada de cada processo e dos antecessores (se existirem) e se processos que tendem a demorar mais tempo chegarem primeiro o tempo médio de espera e o turnaround acabam sendo aumentados.

Um dos problemas são: Inadequado para sistemas interativos (Um dos algoritmos mais antigos) e Processos CPU bound -> monopolização da CPU

SJF

Um escalonamento onde o menor processo ganhará a CPU e atrás do mesmo formar uma fila de processos por ordem crescente de tempo de execução, não-preemptivo. Desvantagem: baixo aproveitamento quando se tem poucos processos prontos para serem executados.

Dois esquemas:

● Não-preemptivo: processo em execução não é interrompido até que termine seu surto

● Preemptivo: se chegar novo processo com surto menor do que tempo restante processo atual, interrompe atual → Shortest remaining time first (SRTF)

O escalonamento SJF é vantajoso por sua simplicidade e também porque minimiza o tempo médio que cada processo leva desde quando ele é criado até o fim de sua execução, incluindo aqui o tempo de espera entre o momento em que ele é criado e o momento em que é selecionado para executar. No entanto, essa estratégia pode levar a inanição de processos com longos tempos de execução caso processos curtos sejam continuamente adicionados ao escalonador. Highest Response Ratio Next é um algoritmo similar que resolve este problema ao levar em conta o envelhecimento dos processos.

Um problema é determinar o tempo de execução de um processo antes de executá-lo (pode ser feito através de análise histórica de execução do processo)