

REVISÃO GERAL

Tarefa:

- Uma tarefa (task) é um fluxo seqüencial de instruções, construído para atender uma necessidade específica.
- Uma seqüência de instruções é escrita em uma linguagem qualquer (como C/C++), que então é compilada e gera uma seqüência de instruções em linguagem de máquina.
- A execução desta seqüência de instruções em linguagem de máquina é denominada tarefa ou atividade.

Programa: Conjunto de seqüência de instruções escrita para resolver um problema específico. Um programa é estático sem estado interno ou interação com outros componentes.

Tarefa: Execução pelo processador das seqüências de um programa. Uma tarefa é dinâmica, ou seja, possui um estado interno bem definido em cada instante de sua execução e, interage com outras entidades, como usuários, periféricos, tarefas, etc. Programas podem ser implementados como processos ou threads.

Gerência de Processos

- Em um computador existem vários processos competindo pelo processador.
- Cada processo tem um comportamento, duração e importância diferentes.
- Cabe ao SO organizar e escalonar (decidir a ordem de execução) as tarefas.

Threads:

Existem aplicações que possuem necessidades especiais como:

- Processamento paralelo.
- Operações de I/O assíncronas.
- Operações de I/O paralelas.

Nestes casos associar mais de uma tarefa em um mesmo contexto pode trazer algumas vantagens:

- Economizar recursos computacionais.
- Facilidades de criação/destruição, comunicação, sincronização, etc.

Escalonamento

- Em um SO convencional possui mais tarefas que processadores e cada tarefa possui um comportamento próprio (duração, uso de recursos, E/S, etc.).
- Para amenizar este problema os SO compartilham o processador, de forma a atender todas as tarefas da melhor forma possível.
- Cabe ao escalonador (scheduler) decidir a ordem de execução destas tarefas. O algoritmo usado no escalonador define o comportamento do SO, fazendo com que o SO execute as tarefas de forma mais eficiente e rápida.

Escalonadores podem ser preemptivos ou não-preemptivos.

Sistemas não-preemptivos: a tarefa em execução permanece no processador tanto quanto possível e só abandona o processador caso termine de executar, solicite alguma operação de E/S ou libere explicitamente o processador. Estes sistemas também são conhecidos como cooperativos.

Exemplo (FCFS):

A forma mais simples de escalonamento consiste em atender as tarefas em seqüência conforme as tarefas se tornam prontas. Este é o escalonador FCFS (First-Come, First Served).

tarefa	t_1	t_2	t_3	t_4
ingresso	0	0	1	3
duração	5	2	4	3

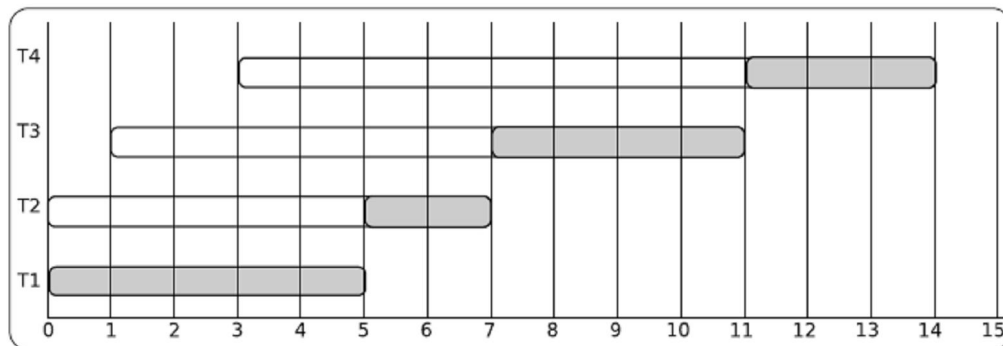


Figura 1: FCFS

Sistemas preemptivos: nestes sistemas a tarefa pode perder o processador caso termine seu quantum, execute uma chamada de sistema ou alguma tarefa mais prioritária acorde. A cada interrupção o escalonador pode

reavaliar as tarefas prontas e decidir se mantém ou substitui a tarefa em execução.

Exemplo (RR):

O escalonador Round-Robin (ou escalonamento por revezamento) é o algoritmo FCFS adicionado de preempção por tempo. Nele cada tarefa é executada por um determinado quantum e depois é retirado do processador e retorna para o final da fila de tarefas prontas.

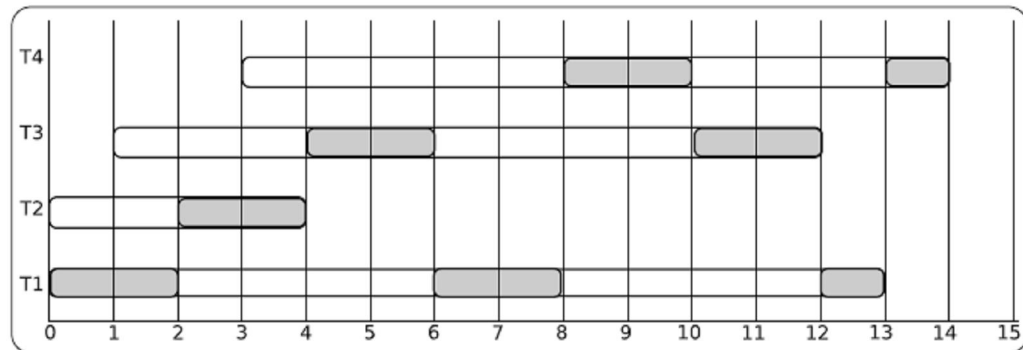


Figura 2: Round-Robin

Impasses/Deadlock:

Deadlock (interbloqueio, blocagem, impasse) Ele é uma falha e não um erro, no contexto de sistemas operacionais (SO), caracteriza uma situação em que ocorre um impasse e dois ou mais processos ficam impedidos de continuar suas execuções, ou seja, ficam bloqueados.

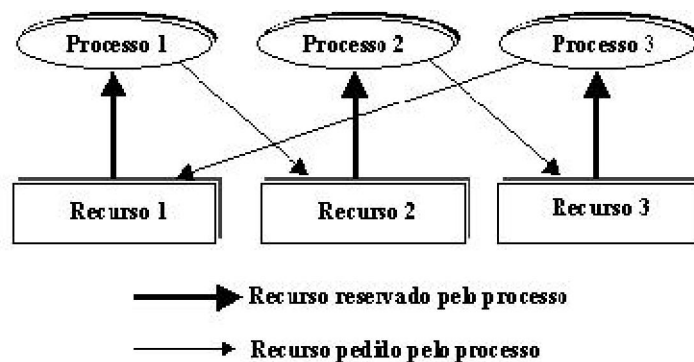


Figura 3

Podemos citar como exemplo de situação de deadlock, não relacionado a computação, mas que facilita o entendimento do que seja uma situação de deadlock, dois carros

seguindo em direção oposta numa pista que permite apenas a passagem de um veículo. Nesse caso os dois ficam impedidos de continuar seu percurso.

Questões:

- 1) Explique a diferença entre programa e tarefa?
- 2) Qual a função do gerenciador de processos?
- 3) Quais as vantagens do uso de threads? Cite exemplos de aplicações onde se podem aproveitar as vantagens do uso de threads?
- 4) Como se comportaria o algoritmo RR se for inativada a função de preempção de tarefas do sistema operacional?