
Escalonamento em Sistemas de Propósito Geral

Fundamentos dos Sistemas de Tempo Real
2ª Edição

Rômulo Silva de Oliveira
Edição do Autor, 2020

www.romulosilvadeoliveira.eng.br/livrotemporeal



Como escalonar as tarefas em um sistema de propósito geral ?

Escalonamento de Tarefas

- Sistemas de tempo real são organizados em torno do conceito de tarefas
- Tarefas podem ser implementadas como:
 - Funções em um executivo cíclico
 - Tratadores de interrupções
 - Threads
- Forma mais comum: tarefa implementada como thread
 - Pode ser com microkernel ou com kernel complexo
- Em geral existem muito mais tarefas do que processadores
- É preciso definir que algoritmo será usado para escolher qual tarefa será executada a seguir
 - No caso de multicore, quais serão executadas

Sistemas de Propósito Geral

- Vários objetivos em sistemas de propósito geral
- Aumentar a capacidade de processamento de dados (*throughput*)
- Reduzir os sobrecustos (*overhead*)
- Oferecer uma ilusão de paralelismo entre os programas do usuário
 - Reduzir o tempo médio de resposta percebido pelos usuários
- Esses objetivos são por vezes conflitantes

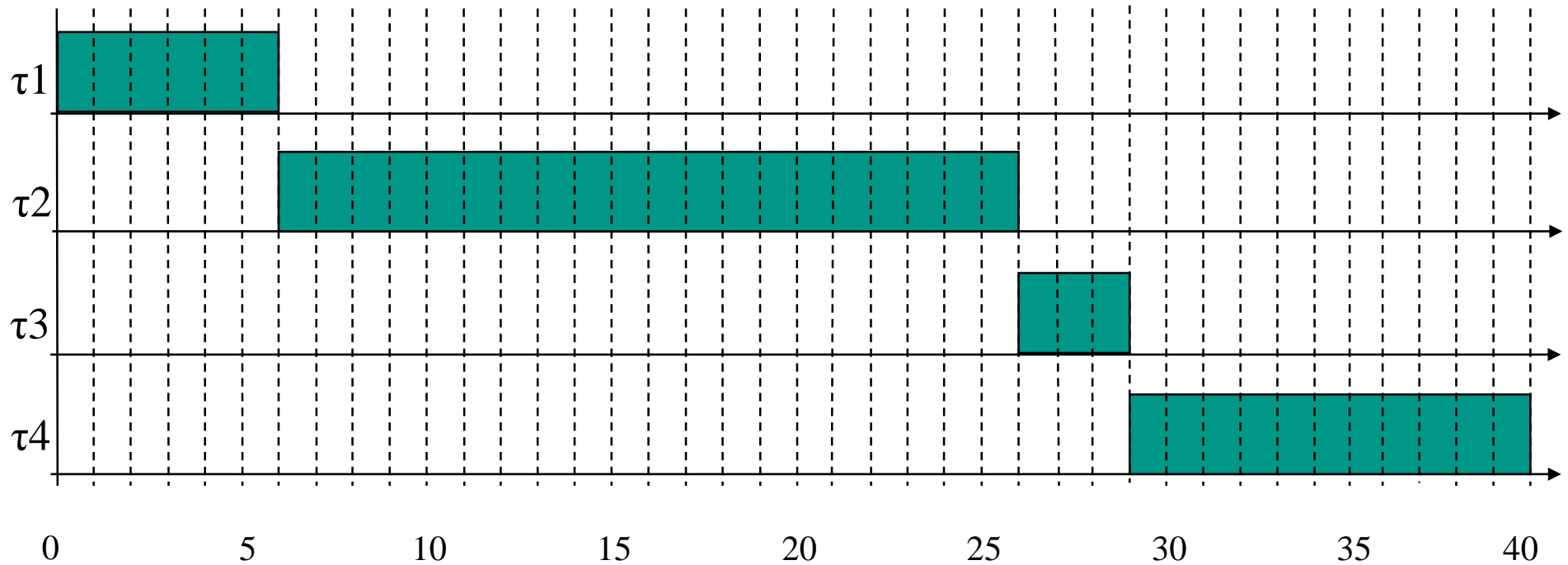
Ordem de Chegada 1/3

- Algoritmo **Ordem de Chegada**
 - **FCFS** (*First-Come, First-Served*) ou **FIFO** (*First-In, First-Out*)
- Fácil implementação:
 - Toda tarefa que fica apta vai para o fim da fila
 - Sempre que o processador fica livre, a tarefa do início da fila executa
 - Tarefa executa até que termine ou que fique bloqueada (sleep, receive, scanf, etc)
- Impossibilita postergação indefinida de uma tarefa
 - Uma vez que tarefa entrou na fila, ela será executada no futuro

Ordem de Chegada 2/3

- Algoritmo **Ordem de Chegada**
 - **FCFS** (*First-Come, First-Served*) ou **FIFO** (*First-In, First-Out*)
- Algoritmo intrinsecamente não preemptivo
 - Uma vez que começou a executar, continua até ficar bloqueada
 - Tarefas que precisam de muito processador deixam todas as tarefas na fila esperando
 - Diminui a percepção de paralelismo no sistema
 - Não é um bom algoritmo para sistemas com tarefas interagindo com pessoas
 - É um péssimo algoritmo para sistemas de tempo real
- Custo de implementação é baixo
 - Acontecem poucos chaveamentos de contexto

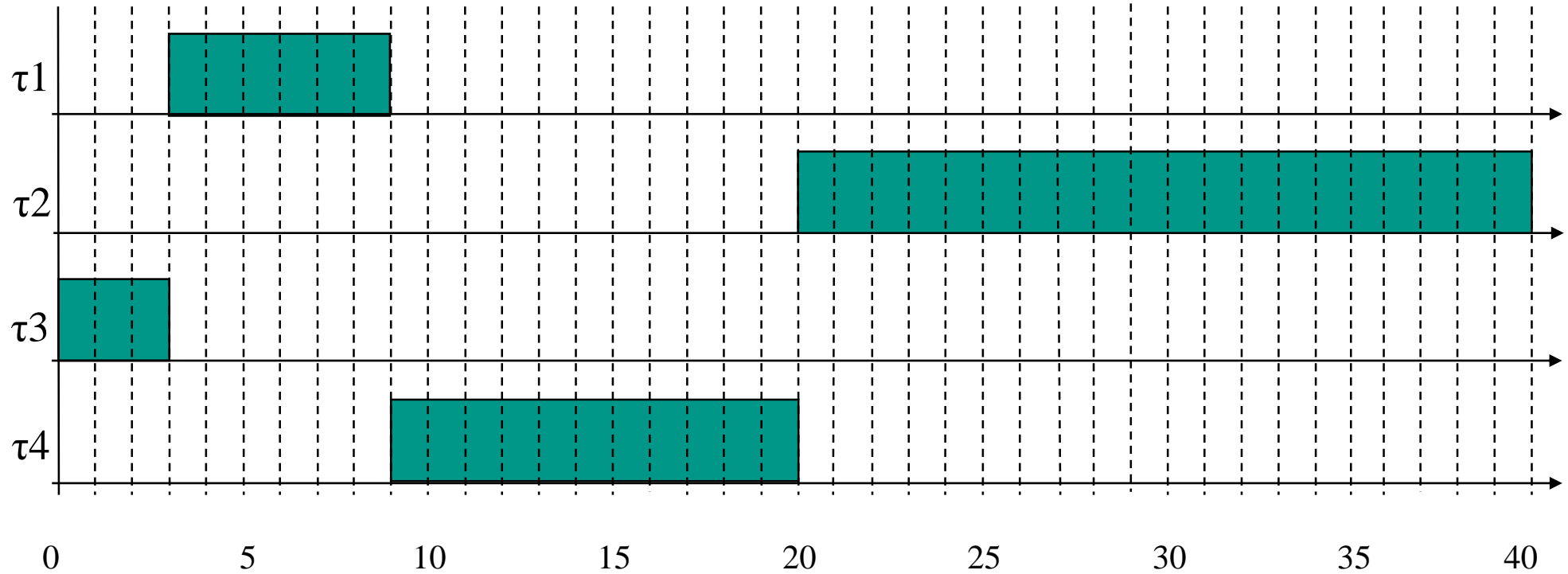
Ordem de Chegada 3/3



SJF (Shortest Job First) 1/2

- Algoritmo conhecido como **SJF** (*Shortest Job First*)
- Executa antes a tarefa que necessite de menos tempo de processador até ficar bloqueada
- Na prática sua implementação é muito difícil ou mesmo impossível
 - Determinar antecipadamente quanto tempo de processador é necessário até o próximo bloqueio
- Minimiza o tempo médio de espera na fila de aptos
 - Relevante em sistemas de propósito geral
- Pouco usado na prática

SJF (Shortest Job First) 2/2

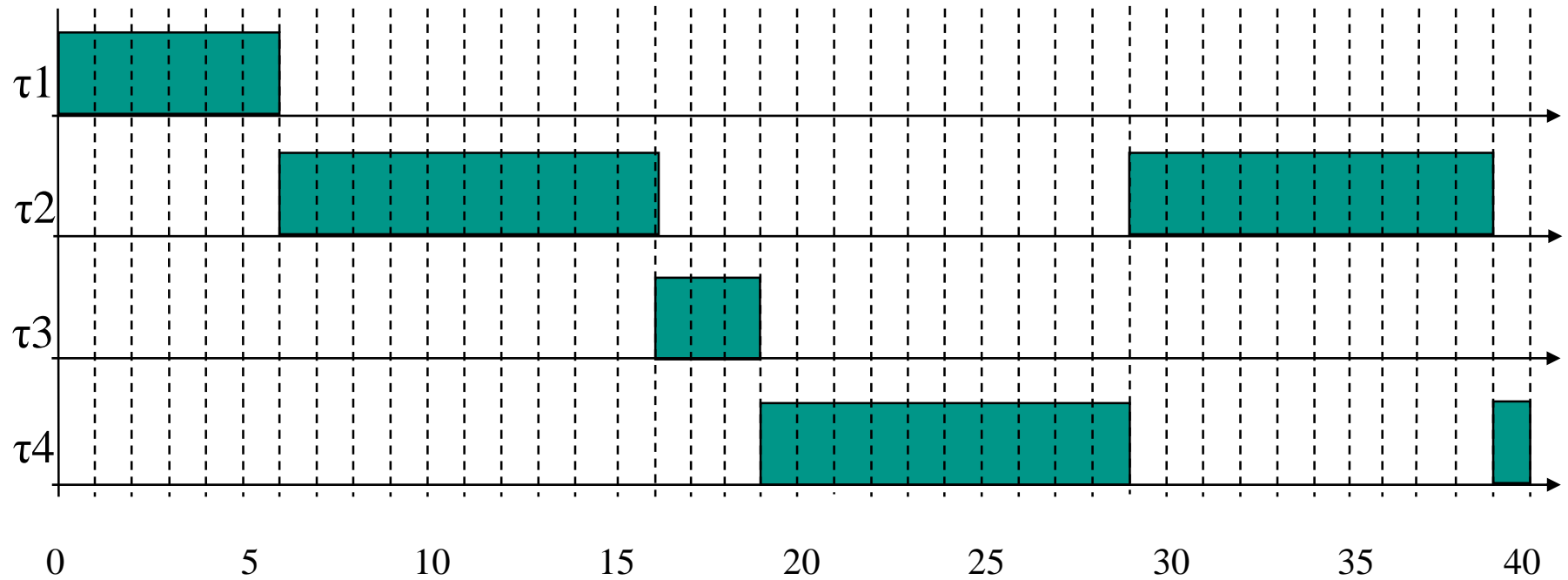


Fatias de Tempo 1/4

- Algoritmo **Fatias de Tempo** ou **RR** (*round-robin*)
- Quando uma tarefa torna-se apta ela é inserida no fim da fila
- Quando o processador fica disponível a primeira tarefa da fila executa

- O sistema operacional define uma fatia de tempo (quantum) máxima
- Caso a tarefa não libere o processador antes
 - ao final da fatia de tempo uma interrupção de timer alerta o escalonador
 - salva o contexto da tarefa em execução
 - insere ela no final da fila de aptos
 - coloca para executar a nova primeira tarefa da fila
- Método é intrinsecamente preemptivo
 - Tira o processador das tarefas mesmo quando elas gostariam de continuar executando
- Impossibilita a postergação indefinida
 - Uma vez na fila, tarefa será executada tão logo as tarefas na sua frente executem

Fatias de Tempo 2/4



- Como definir a duração da fatia de tempo ?
- Um extremo: fatia de tempo é infinita
 - Tarefa jamais irá esgotar sua fatia de tempo
 - Comporta-se como o Ordem de Chegada
- Outro extremo: uma única instrução de máquina
 - Aparência de paralelismo entre as tarefas será perfeita
 - Tempo de chaveamento de contexto seria muito maior do que a fatia de tempo
- Escolha do valor para a fatia de tempo é um balanço entre
 - valores pequenos para criar a ilusão de paralelismo entre tarefas
 - valores grandes para reduzir o custo de implementação (*overhead*)

Fatias de Tempo 4/4

- Existem muitas herísticas para definir a fatia de tempo
- Exemplo: usar a mediana das durações de ciclo de processamento
 - Ciclo de processamento é o tempo de processador necessário para a tarefa entre duas situações de bloqueio
 - Pode-se registrar os valores observados no passado
 - Metade das vezes a fatia de tempo será estourada
 - Metade das vezes a tarefa ficará bloqueada antes de gastar toda a fatia de tempo
 - Quebra as execuções longas com baixo custo de implementação

- Algoritmo baseado em **prioridades**
 - Executa antes a tarefa apta com prioridade mais alta
 - Fila de aptos mantida ordenada pelas prioridades
 - Quando uma tarefa fica apta, ela é inserida na fila conforme a sua prioridade
 - Quando o processador fica disponível, tarefa com prioridade mais alta executa
- Como definir a prioridade de cada tarefa é uma questão complexa

- Como definir a prioridade de cada tarefa é uma questão complexa
- Sistemas operacionais de propósito geral:
 - Pode ser feito pelo usuário (por exemplo, “nice” no Linux)
 - Dentro dos programas (por exemplo, “setpriority()” no Linux)
- Neste caso o critério é do usuário
 - Ele define quais programas quer executar antes
- Pode também ser feita pelo kernel
 - Por exemplo, processos que usam muito tempo de processador tem prioridade reduzida para não monopolizar o recurso

- Prioridades podem gerar postergação indefinida
 - Uma tarefa com prioridade muito baixa, sempre existe alguma outra tarefa na fila com prioridade mais alta que ela
- Sistemas operacionais de propósito geral empregam mecanismo chamado de **Envelhecimento** (*aging*)
 - A medida que tarefa “envelhece” na fila, sem executar, sua prioridade é lentamente elevada pelo sistema
 - Um dia sua prioridade torna-se competitiva e ela executa
 - Elevação de prioridade é lenta e gradual, ela tem baixa prioridade
 - O mecanismo de envelhecimento quer apenas evitar que a tarefa fique na fila para sempre sem executar
 - Depois de executar e ficar bloqueada, retorna com sua prioridade original

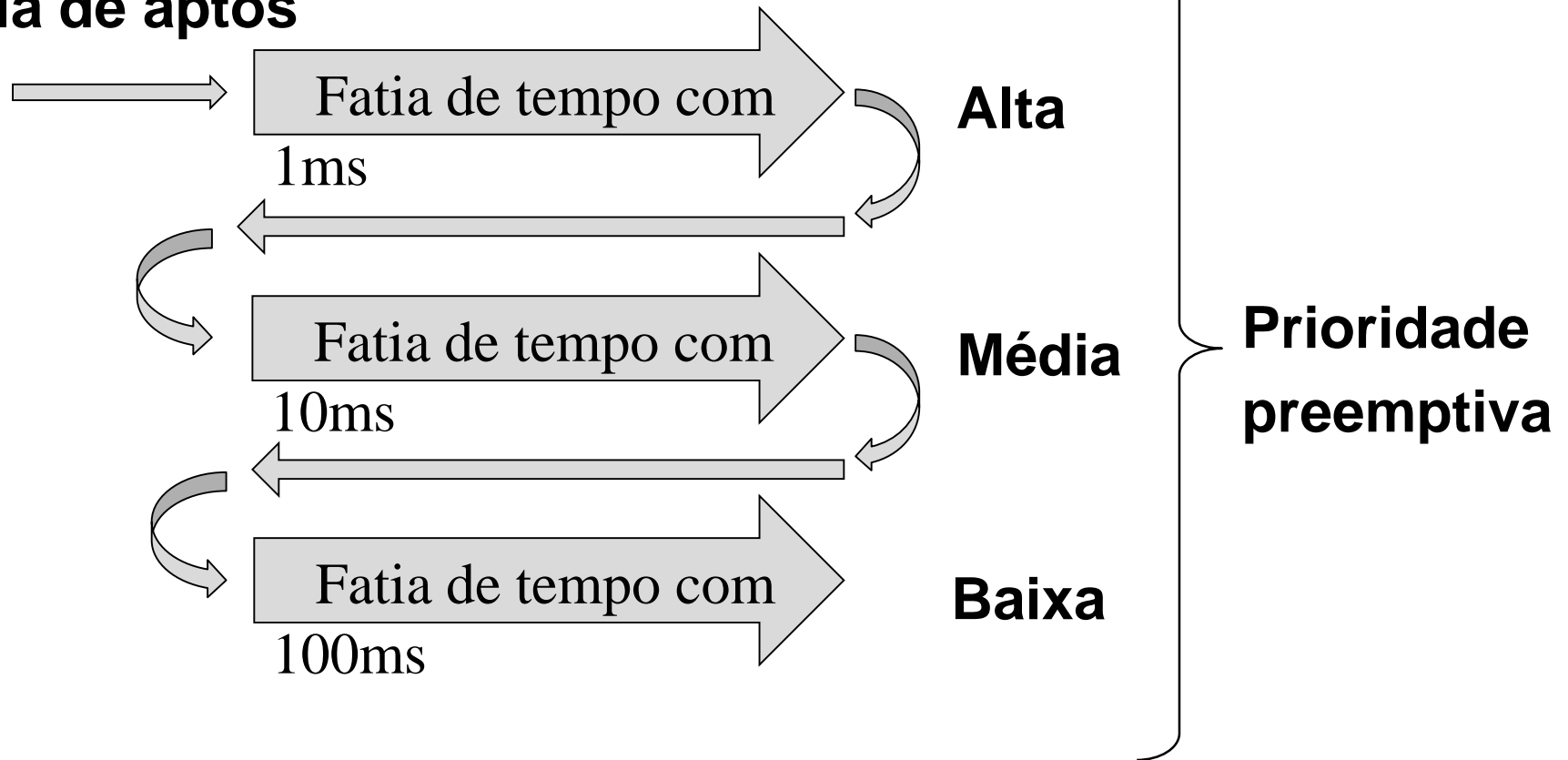
- O que fazer quando uma tarefa de prioridade mais baixa está executando e uma tarefa de prioridade mais alta torna-se apta ?
- Prioridades Preemptivas:
 - Contexto da tarefa de baixa prioridade é salvo
 - Ela é re-inserida na fila de aptos
 - Tarefa de alta prioridade assume o processador
- Esta é a forma natural de implementar prioridades, pois respeita a atribuição de prioridades feita pelo sistema e/ou o usuário

- Prioridades Não Preemptivas
 - Quando tarefa é colocada para executar, ela permanece até que faça uma chamada de sistema e fique bloqueada
 - Tarefa de alta prioridade liberada (ficou apta) é inserida na fila de aptos
 - Ela não “preempta” a tarefa de baixa prioridade em execução
- Prioridades não preemptivas geram **inversão de prioridades**
- Tarefa de alta prioridade na fila de aptos espera pela tarefa de baixa prioridade em execução
- Tal situação não é desejada em sistemas operacionais de propósito geral e muito menos em sistemas de tempo real

Múltiplas Filas 1/3

- Maioria dos sistemas operacionais de propósito geral emprega uma mistura dos métodos básicos apresentados: **Múltiplas Filas**
- Tipicamente são usadas prioridades para permitir ao sistema e ao usuário definirem o que deve ser executado antes
- Diversas tarefas podem receber a mesma prioridade
- É necessário um algoritmo para ordenar tarefas que possuem a mesma prioridade.
- Tipicamente usada fatia de tempo
 - Alguns sistema pequenos usam ordem de chegada
- Cada sistema operacional de propósito geral emprega uma solução de escalonamento ligeiramente diferente dos demais
 - Além de permitir um certo nível de configuração por parte do administrador

Tarefas chegando na fila de aptos



- Maioria dos sistemas operacionais de propósito geral emprega uma mistura dos métodos básicos apresentados: **Múltiplas Filas**
- Cada sistema operacional de propósito geral emprega uma solução de escalonamento ligeiramente diferente dos demais
- Além de permitir um certo nível de configuração por parte do administrador

- Escalonamento de Tarefas
- Sistemas de Propósito Geral
- Ordem de Chegada
- SJF (Shortest Job First)
- Fatias de Tempo
- Prioridades
- Múltiplas Filas

