



# ESCALONAMENTO DE PROCESSOS

---

[flaviamsn@ifba.edu.br](mailto:flaviamsn@ifba.edu.br)

# PORQUE É NECESSÁRIO ESCALONAR?

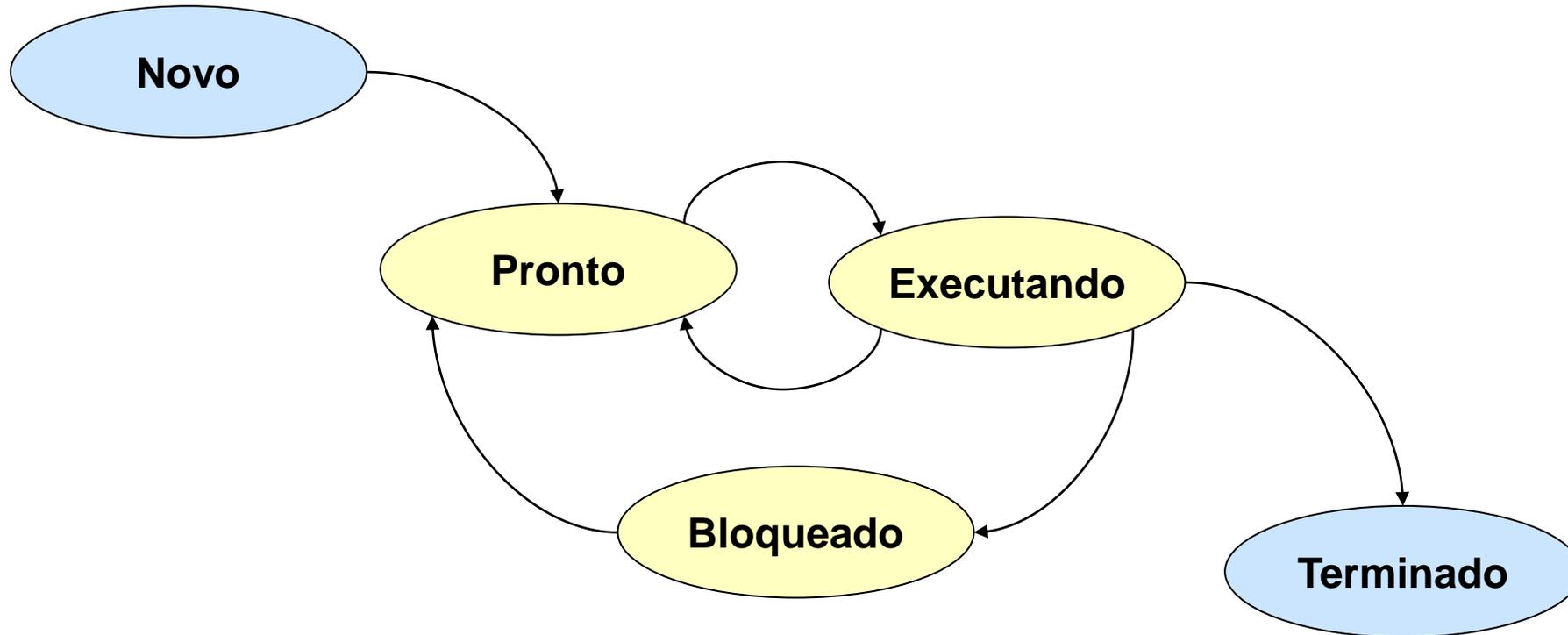
Processos precisam ser executados

Processos concorrem a CPU

Escalonador:

- Componente (implementação) do sistema operacional
- Determina a ordem de execução dos processos baseado num *algoritmo de escalonamento*
- Lê a fila que contém os processos no estado “pronto” e os ordena para execução

# O QUE PROVOCA O ESCALONAMENTO?



# TIPOS DE ALGORITMO DE ESCALONAMENTO

## Preemptivo:

- Execução de um processo dura um tempo pré-determinado
- Quando o tempo acaba, o processo é interrompido.

## Não-preemptivo:

- Processo fica em execução até que:
  - Termine
  - Libere a CPU VOLUNTARIAMENTE
  - Seja bloqueado por falta de recurso

# O QUE AFETA A PERFORMANCE DE UM ALGORITMO DE ESCALONAMENTO?

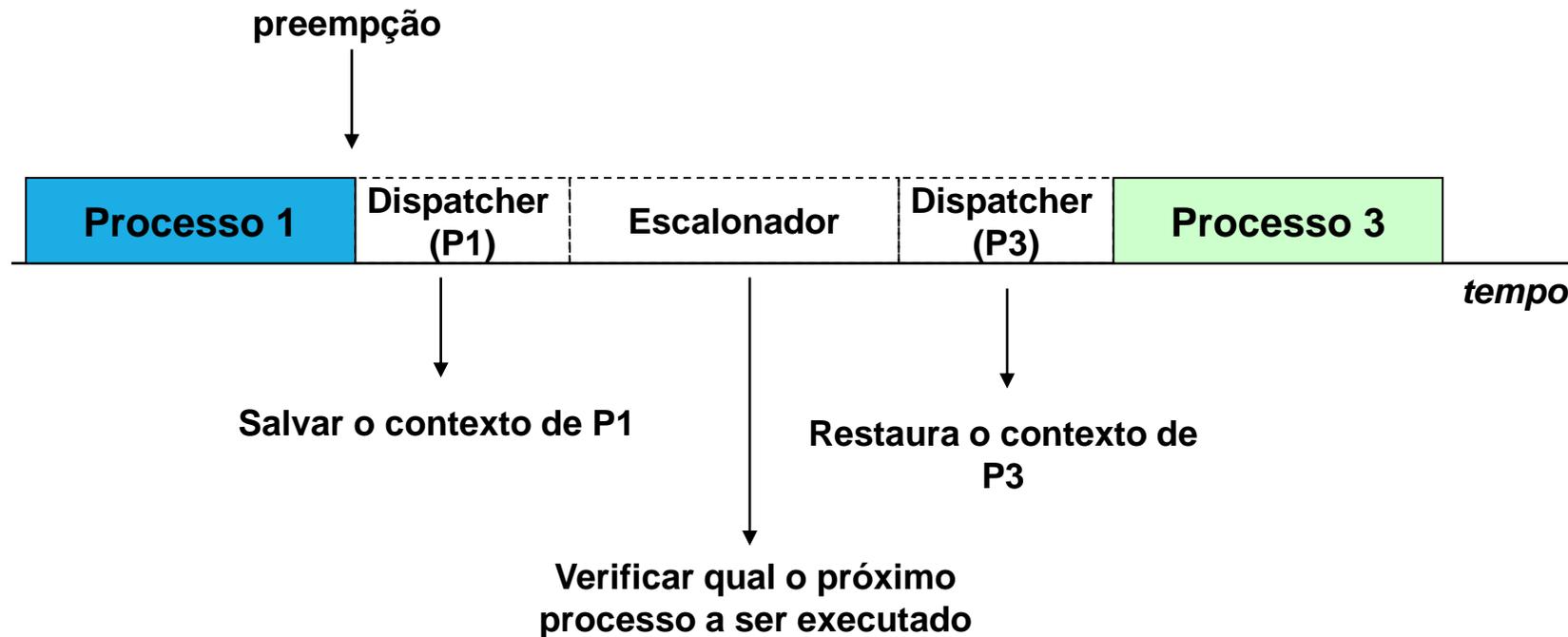
Cada processo possui informações que permitem definir precisamente seu estado.

- Tais informações definem o **contexto** do processo

## Troca de Contexto

- Mecanismo que permite ao escalonador interromper uma tarefa, e executá-la posteriormente, sem corromper seu estado.
- Separação do escalonamento
  - Escalonamento = Política + Mecanismo

# ILUSTRAÇÃO DA TROCA DE CONTEXTO



# QUAL O OBJETIVO DO ESCALONAMENTO?

DEPENDE do tipo de sistema operacional

- Lote:
  - Não possui usuários aguardando → pode ser preemptivo ou não
  - Não possui muita troca de contexto
  - OBJETIVOS:
    - melhorar o throughput (vazão)
    - melhorar o turnaround (tempo entre submissão e finalização)
    - manter a CPU ocupada

# QUAL O OBJETIVO DO ESCALONAMENTO?

- Propósito Geral:
  - Possuem usuários interagindo
  - Precisam ser preemptivos
  - OBJETIVOS
    - melhorar o tempo médio de resposta
    - atender as expectativas dos usuários
- Tempo real:
  - Em geral são preemptivos
  - OBJETIVO:
    - cumprir requisitos lógicos
    - cumprir requisitos temporais

# QUAL O OBJETIVO DO ESCALONAMENTO?

Independente do tipo de sistema operacional, TODOS os algoritmos de escalonamento precisam atender a alguns critérios:

- Justiça (fairness)
- Aplicação da política de escalonamento
- Equilíbrio (balance) entre as partes do sistema

# ESCALONAMENTO PARA SISTEMAS EM LOTE

## FCFS – First Come First Served (ou FIFO)

- Primeiro processo da fila de pronto é o escolhido para executar.
- Não-preemptivo
- Fácil de entender
- Fácil de programar
- “Justo”
- Processos de baixo custo de execução podem esperar muito tempo para ser executado

# ESCALONAMENTO PARA SISTEMAS EM LOTE

## FCFS (ou FIFO)

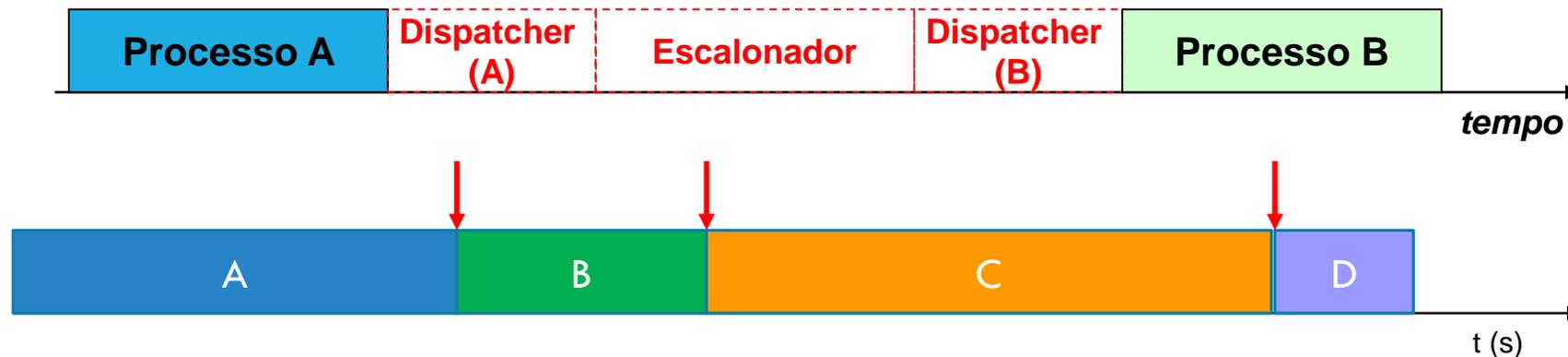
- Fazer o escalonamento para os seguintes processos:

Processo	Custo de execução
A	12 (t=0)
B	8 (t=0)
C	15 (t=0)
D	5 (t=0)

Imagine que no instante de tempo  $t=2$ , chega a fila de pronto um Processo E com custo de execução 3.

Em que instante ele será executado?

$$TRM = (12 + 20 + 35 + 40) / 4 = 26,75$$



# ESCALONAMENTO PARA SISTEMAS EM LOTE

## FCFS (ou FIFO)

- Fazer o escalonamento para os seguintes processos:

Processo	Custo de execução
A	12 (t=0)
B	8 (t=0)
C	15 (t=0)
D	5 (t=0)

Tempo de resposta de cada tarefa

$$TRM = \frac{(12 + 20 + 35 + 40)}{4} = 26,75$$

TRA + TRB + TRC + TRD

Número de processos



# ESCALONAMENTO PARA SISTEMAS EM LOTE

## Menor Job Primeiro (SJF)

- O *job* de menor custo executa primeiro.
- Não-preemptivo
- Fácil de entender
- Fácil de programar
- “Justo”
- Para ser adequado requer que todos os jobs estejam disponíveis simultaneamente

# ESCALONAMENTO PARA SISTEMAS EM LOTE

## Menor Job Primeiro

- Fazer o escalonamento para os seguintes processos

Processo	Custo de execução
A	12
B	8
C	15
D	5

Tempo de resposta de cada tarefa

$$TRM = \frac{(5 + 13 + 25 + 40)}{4} = 20,75$$

TRD + TRB + TRA + TRC

Número de processos

Cálculo do Tempo de Resposta

TR = tempo\_bloqueio + custo

ou

TR = instante\_final - instante\_inicial



# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Alternância circular (*round-robin*)

- Processos executam dentro de uma fatia de tempo predefinida (**quantum**)
- Preemptivo
- Simples
- Justo
- Amplamente utilizado
- Tamanho do *quantum* pode ser um problema

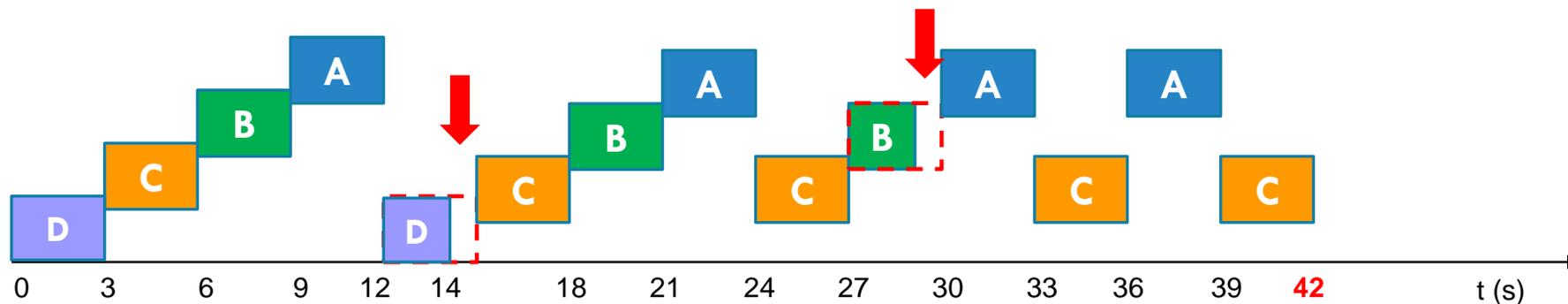
# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Round-Robin

- Fazer o escalonamento para os seguintes processos considerando um *quantum* = 3

Processo	Custo de execução	Instante de chegada
A	12	0
B	8	0
C	15	0
D	5	0

Há desperdício de CPU!



# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

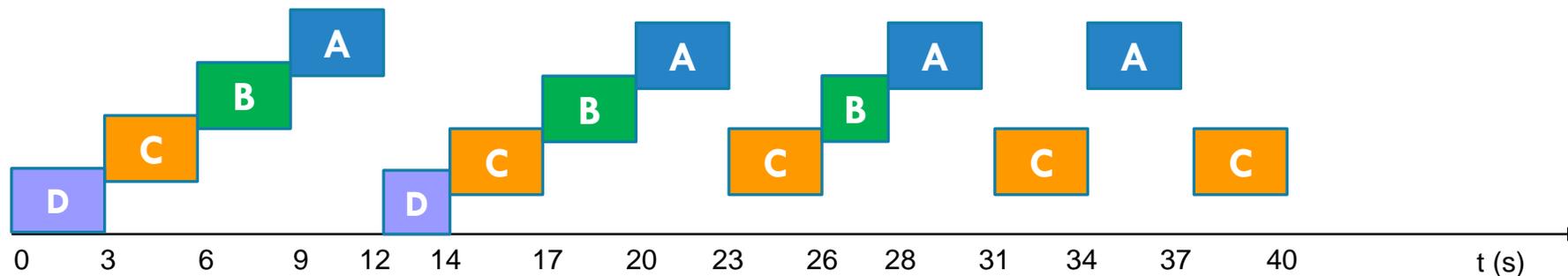
## Round-Robin

- Fazer o escalonamento para os seguintes processos considerando um *quantum* = 3

Processo	Custo de execução	Instante de chegada
A	12	0
B	8	0
C	15	0
D	5	0



**MELHORIA DO ALGORITMO DA ALTERNANCIA CIRCULAR!**



# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Prioridade

- Processos tem diferentes prioridade de execução
- Preemptivo
- Baseado nos ciclos da CPU ou *quantum*
- Prioridade pode ser atribuída estaticamente ou dinamicamente
- Pode ser implementado considerando filas de prioridades
- Pode ocasionar **starvation** (inanição)

# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Prioridade

- Fazer o escalonamento para os seguintes processos

Assumindo que a maior prioridade é 1

Processo	Custo de execução	Prioridade
A	12	3 (t=0)
B	8	4 (t=5)
C	15	2 (t=7)
D	5	1 (t=0)



# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Prioridade

- Fazer o escalonamento para os seguintes processos

Processo	Custo de execução	Prioridade
A	12	3 (t=0)
B	8	4 (t=5)
C	15	2 (t=7)
D	5	1 (t=0)

Assumindo que a maior prioridade é 1

**Sugestão de Davi:**

Imagine que no instante de tempo  $t=10$ , chega a fila de pronto um Processo E com custo de execução 4 e prioridade 2.

Em que instante ele será executado?



# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

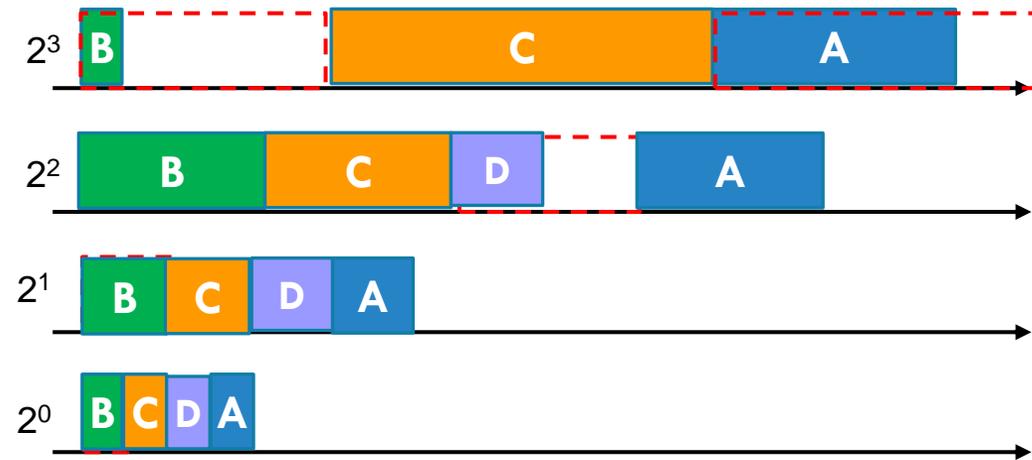
## Filas múltiplas

- Assume mais de uma fila de escalonamento
- Assume quantum de tamanho diferente em cada fila
- Preemptivo
- Tem um número elevado de preempção/troca de contexto
- Pode não ser eficiente

# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Filas Múltiplas

Processo	Custo de execução	Instante de chegada
A	12	0
B	8	0
C	15	0
D	5	0



Ordem de execução: BCDA



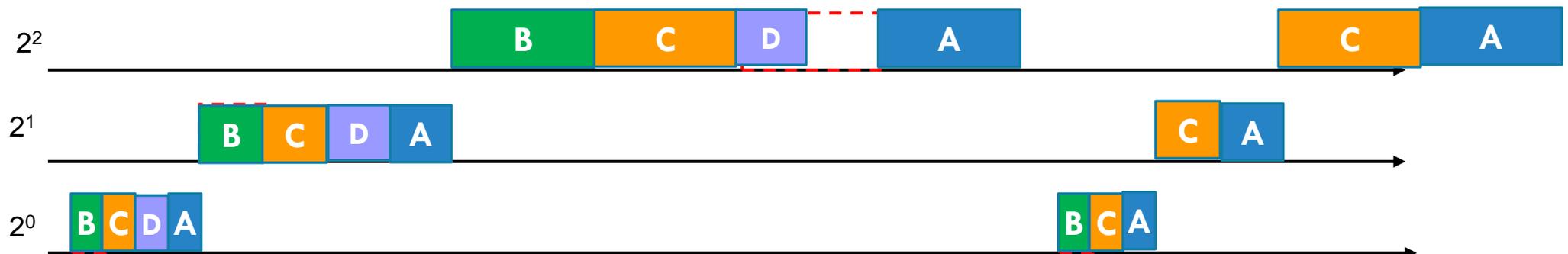
O algoritmo implementa várias filas, cada uma com quantum de tamanho diferente

# ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

## Filas Múltiplas

Processo	Custo de execução	Instante de chegada
A	12	0
B	8	0
C	15	0
D	5	0

Ordem de execução: BCDA



# PARA A PRÓXIMA AULA

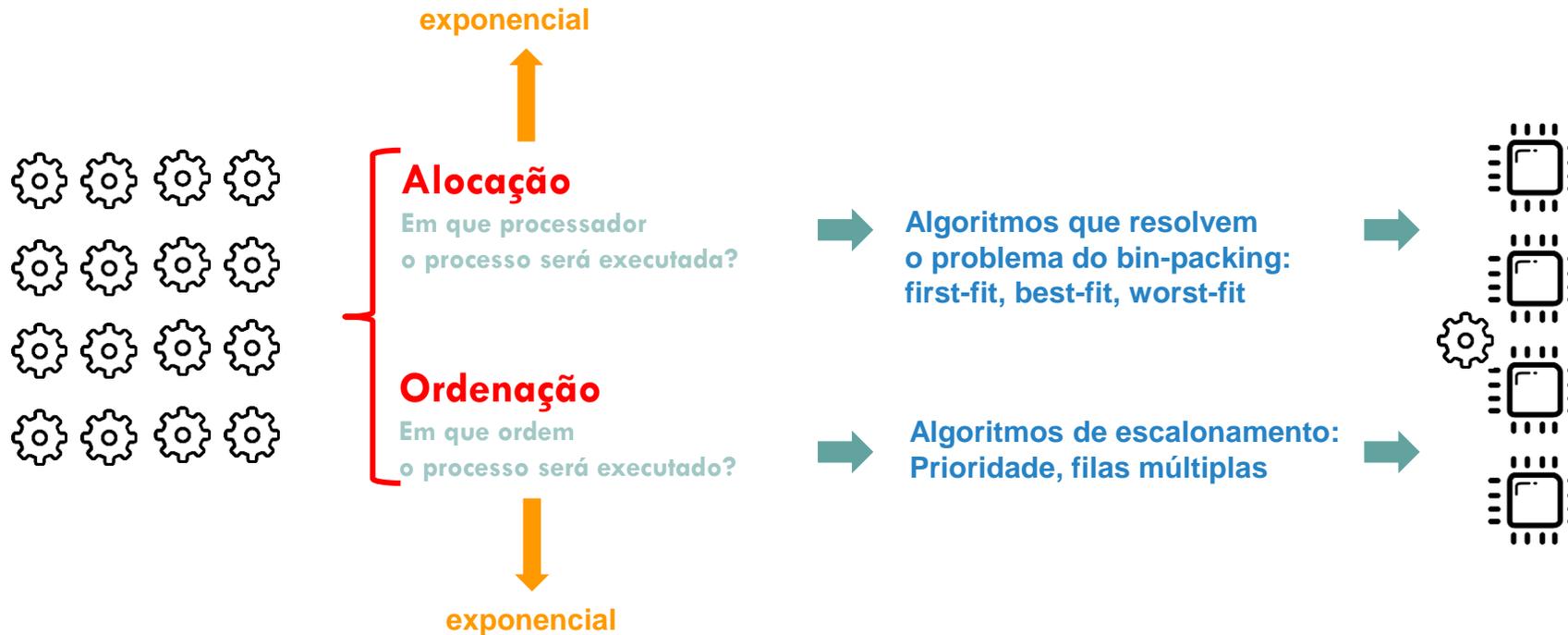
Trazer todos os exercícios dos slides respondidos.

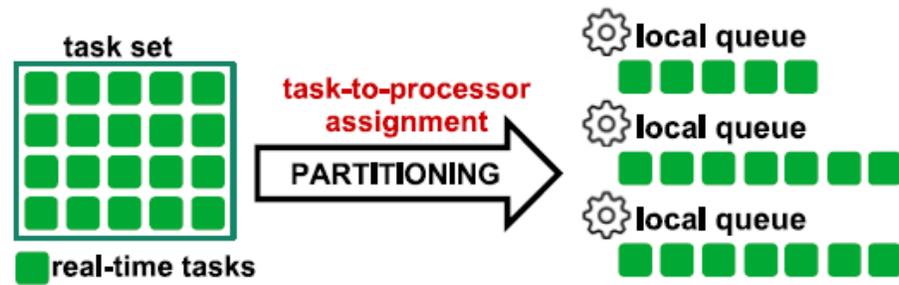
Verificar as implementações de semáforo para o problema do produtor consumidor.

Escalonamento com múltiplas filas.

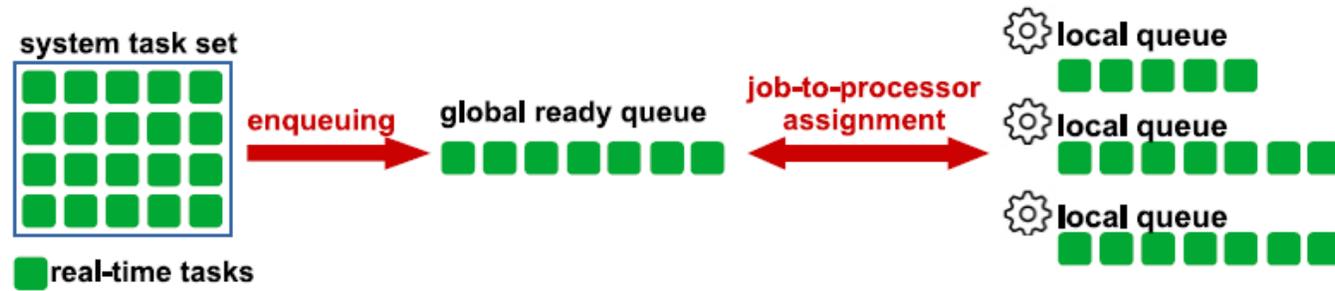
Descrever a diferença entre processos I/O- Bound e CPU-Bound.

# ESCALONAMENTO MULTIPROCESSADO





(a) Partitioned scheduling approach.



(b) Global scheduling approach.

Use the first-fit algorithm to pack the weights 11, 2, 15, 5, 6, 17, and 7 into bins with capacity 20.

