

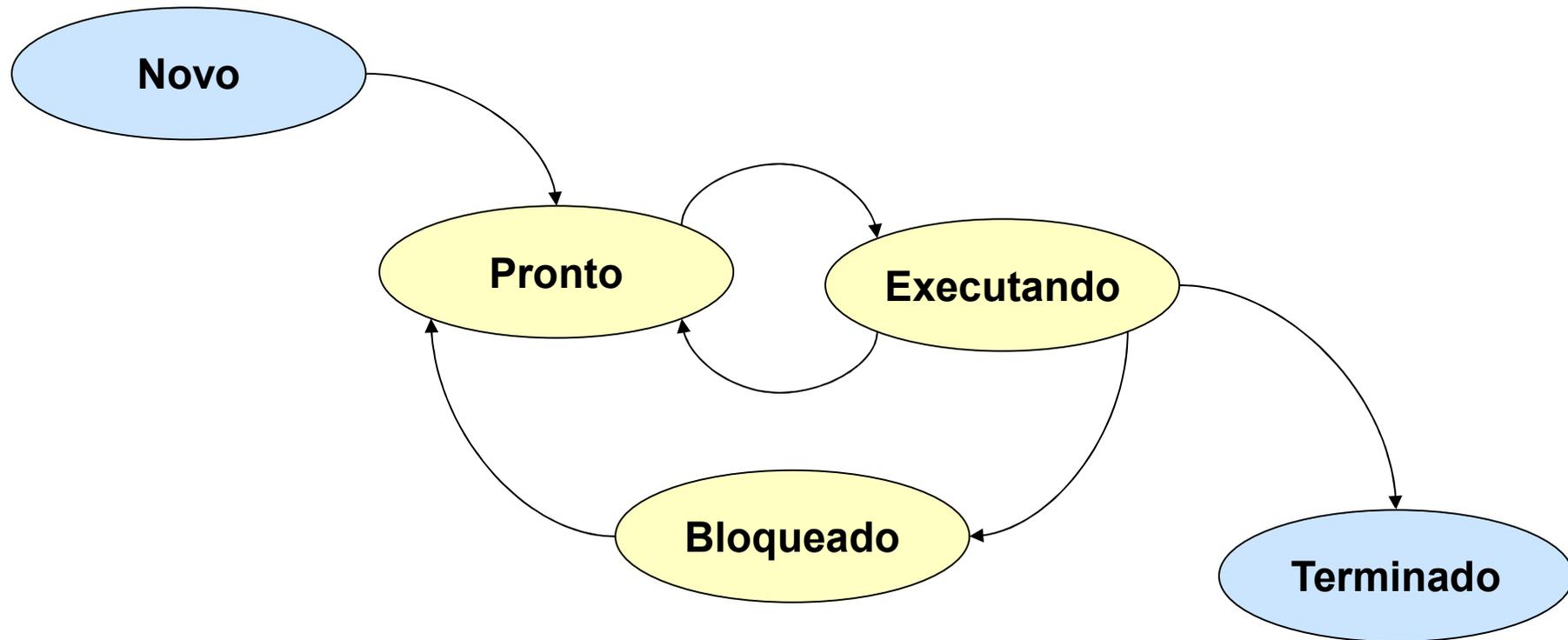
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
BAHIA
Campus Salvador

Escalonamento de Processos

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.

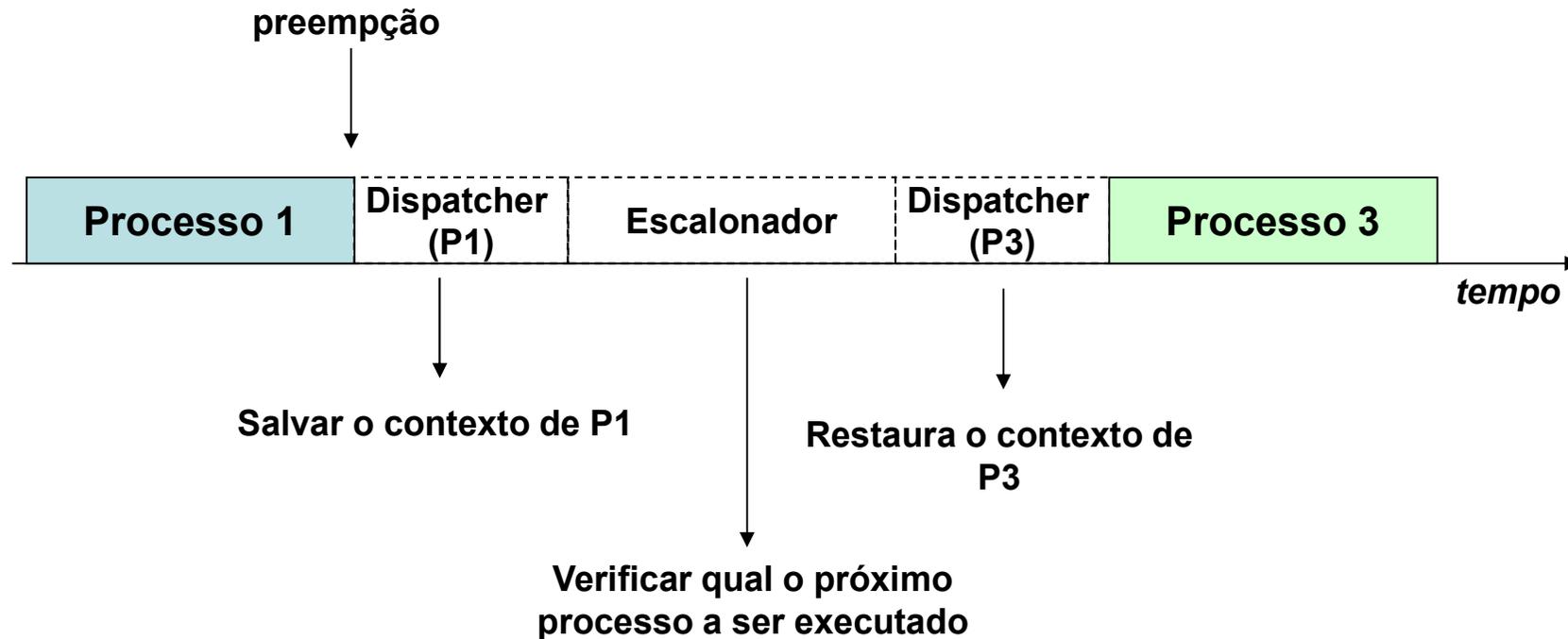
- Não-preemptivo:

- 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)

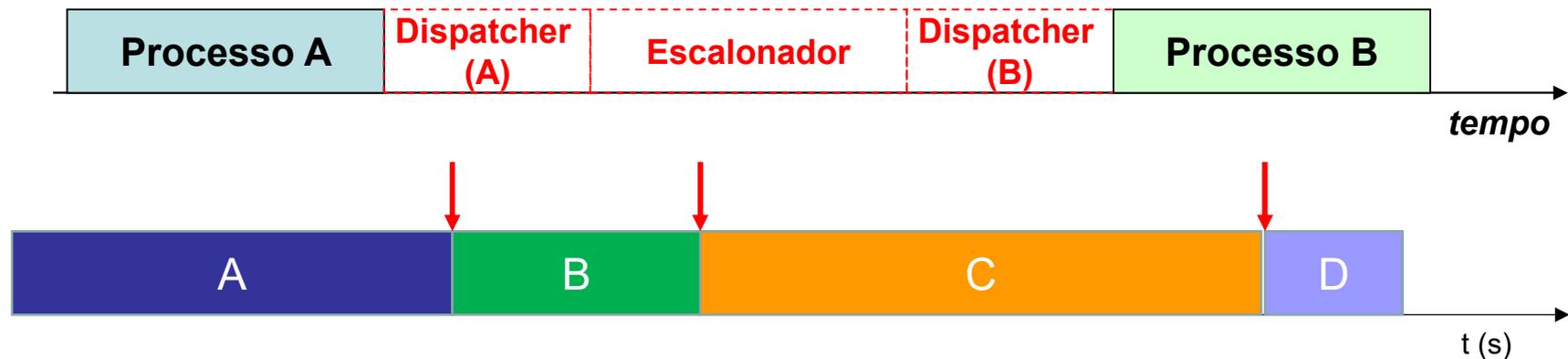
– Fazer o escalonamento para os seguintes processos:

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)

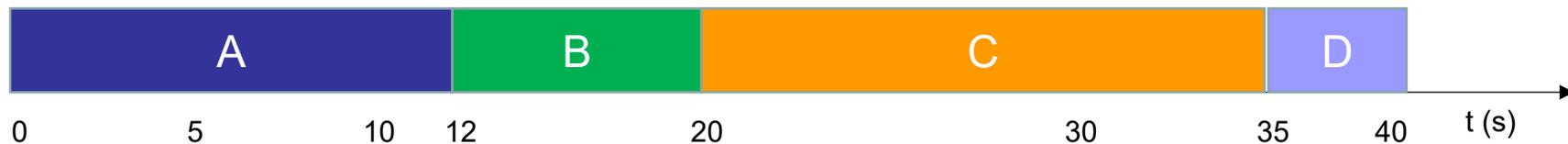
– Fazer o escalonamento para os seguintes processos:

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

$$\text{TRM} = \frac{(12 + 20 + 35 + 40)}{\text{TRA} + \text{TRB} + \text{TRC} + \text{TRD}} \cdot 4 = 26,75$$

↙ 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

$$TRM = \frac{(5 + 13 + 25 + 40)}{TRD + TRB + TRA + TRC} \cdot 4 = 20,75$$

↳ 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

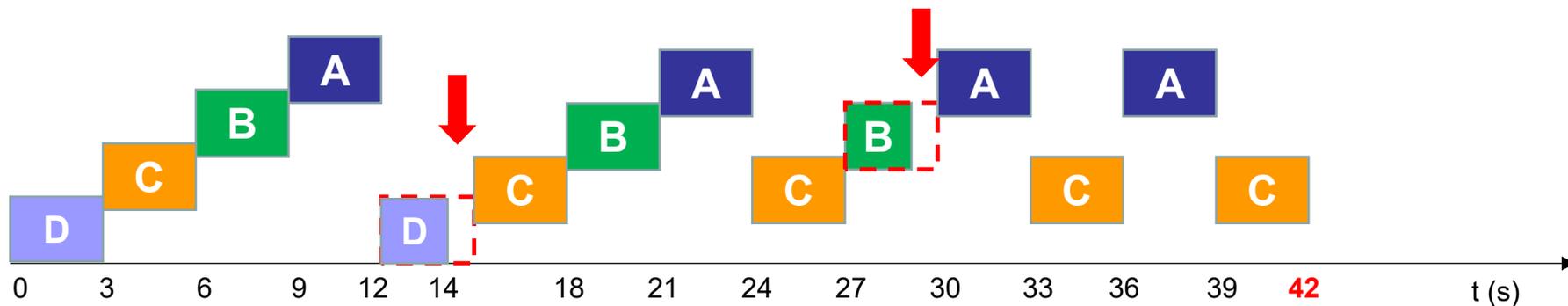
Escalonamento em sistemas de propósito geral

■ 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

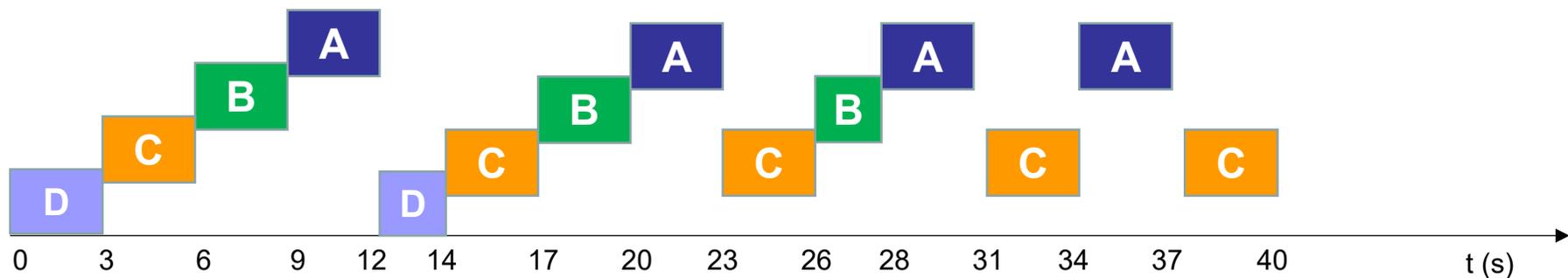
■ Round-Robin

- 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)

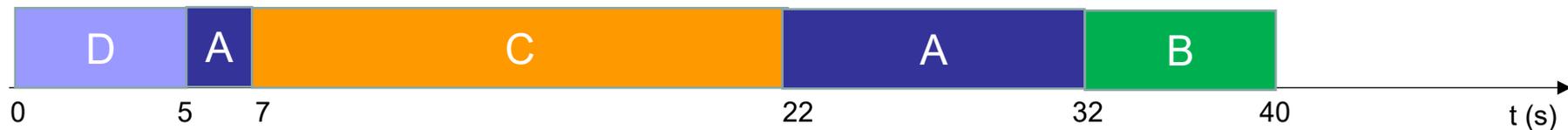
Escalonamento em sistemas de propósito geral

■ 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

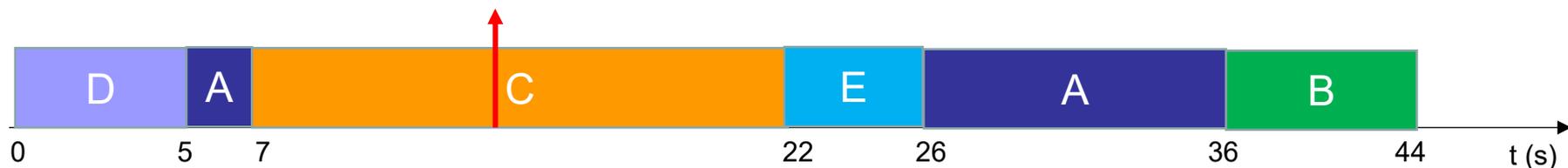
Processo	Custo de execução	Prioridade
A	12	3 (t=0)
B	8	4 (t=5)
C	15	2 (t=7)
D	5	1 (t=0)

Assumindo que a maior prioridade é 1

Sugestão de Davi:

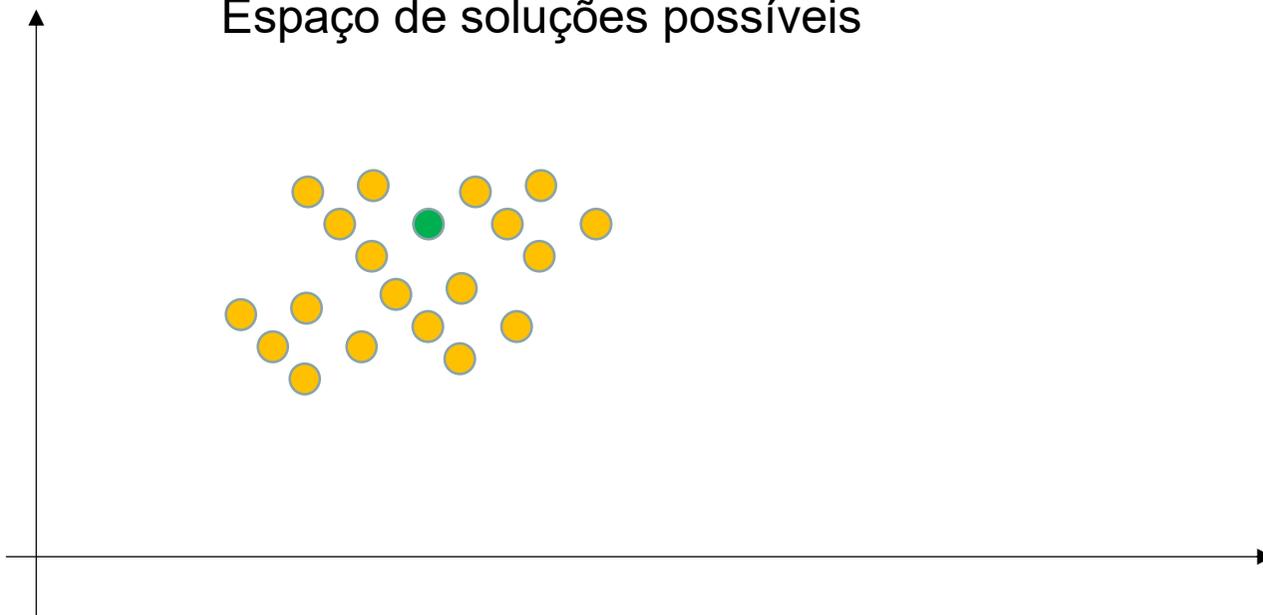
Imagine que no instante de tempo $t=10$, chega a fila de pronto um Processo E com custo de execução 4 e prioridade 2.

Em que instante ele será executado?





Espaço de soluções possíveis



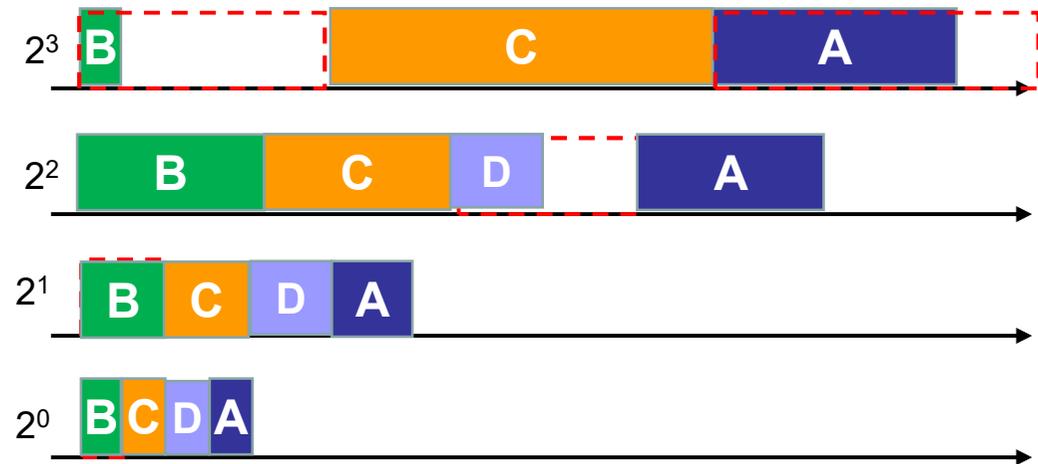
Escalonamento em sistemas de propósito geral

- 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

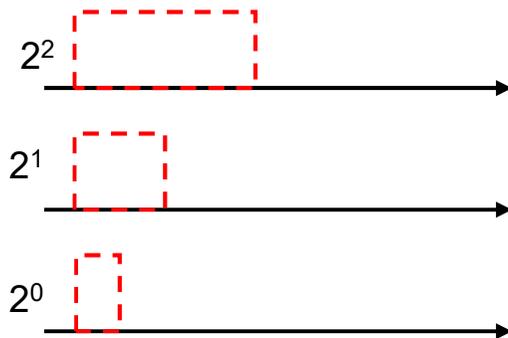
Escalonamento em sistemas de propósito geral

■ 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

