Ministério da Educação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
UNED Nova Friburgo
Bacharelado em Sistemas da Informação

Escalonamento

Sistemas Operacionais

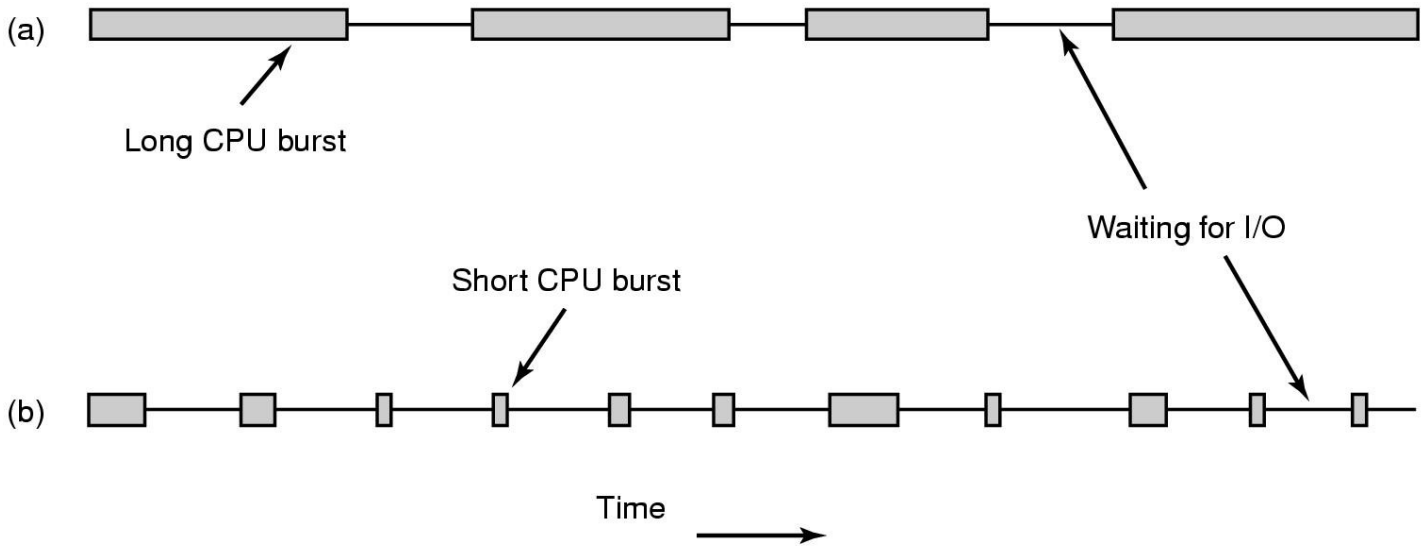
Prof. Bruno Policarpo Toledo Freitas
bruno.freitas@cefet-rj.br

Objetivos

- **Definir escalonamento, seus objetivos e principais algoritmos**
- **Diferenciar algoritmos de lote, interativos e de sistemas de tempo real**
- **Compreender como diferentes algoritmos interferem no desempenho geral do sistema**
- **Aprimorar a capacidade de avaliar vantagens e desvantagens de algoritmos**

Comportamento de Processos

- **Utilização da CPU alterna com períodos de espera de I/O.**
  a) Processos CPU-Bound
  b) Processos I/O-Bound



Quando escalonar?

- **Principais situações:**
  - Criação de um novo processo
  - Término de um processo
  - Bloqueio de I/O
  - Interrupções
- **Decisões de escalonamento podem ser tomadas segundo interrupções de relógio**
  - Preemptivos
  - Não-preemptivos

Categorias

- **Lote**
  - Não há necessidade de resposta rápida de programas
  - Reduz chaveamento de contextos
- **Interativo**
  - Evitar posse da CPU por longos períodos de tempo
  - São preemptivos
- **Tempo Real**
  - Não podem executar por longos períodos de tempo
  - Tarefas específicas para a aplicação.

Objetivos gerais

- **Justiça**
  - Todo mundo usar a CPU.
- **Aplicação de política**
  - Finalidade do sistema determina prioridades.
- **Equilíbrio**
  - Manter o computador ocupado.

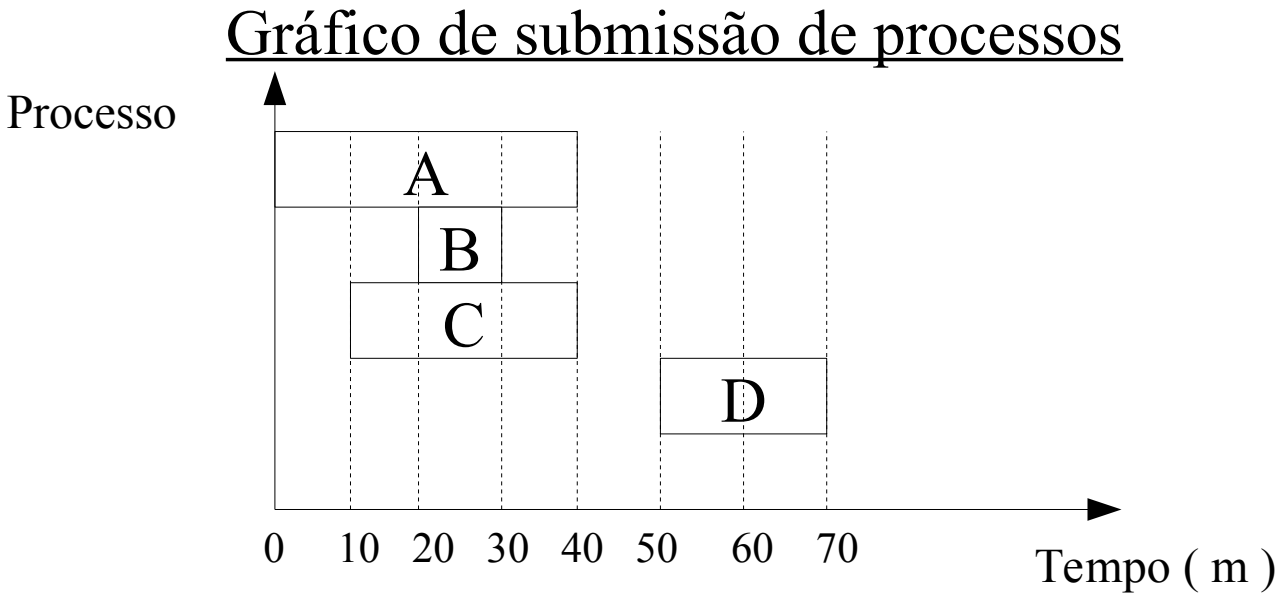
Escalonamento em sistemas de lote
Objetivos

- **Vazão (*throughput*)**
  - quantidade de tarefas terminadas em um dado período de tempo (geralmente horas)
- **Tempo de retorno (*turnaround time*)**
  - minimizar tempo entre submissão e término
- **Utilização da CPU**

Escalonamento em sistemas de lote
First Come, First Served (FCFS)

- **Ordem de submissão dos processos**
- **Vantagens**
  - Justo
- **Desvantagens**
  - Tempo de retorno maior

Escalonamento em sistemas de lote
First Come, First Served (FCFS)



Considerando que:

- A=40, B=10, C=30, e D=20
- Inícios: A=0, B=20, C=10. e D=50

Qual é o :

- Tempo médio de retorno?
- Vazão?

Ordem de escalonamento:

$A \rightarrow C \rightarrow B \rightarrow D$

Escalonamento em sistemas de lote
First Come, First Served (FCFS)

Tarefa	Tempo de computação	Submissão	Início	Fim
A	40	0	0	40
C	30	10	40	70
B	10	20	70	80
D	20	50	80	100

$$tr_p = t_{fim} - t_{submissao}$$

$$\bar{tr} = \frac{\sum_{p=1}^{n} tr_p}{n}$$

$$vazão = \frac{número\ de\ tarefas}{período\ de\ tempo}$$ (Geralmente horas)

Escalonamento em sistemas de lote
First Come, First Served (FCFS)

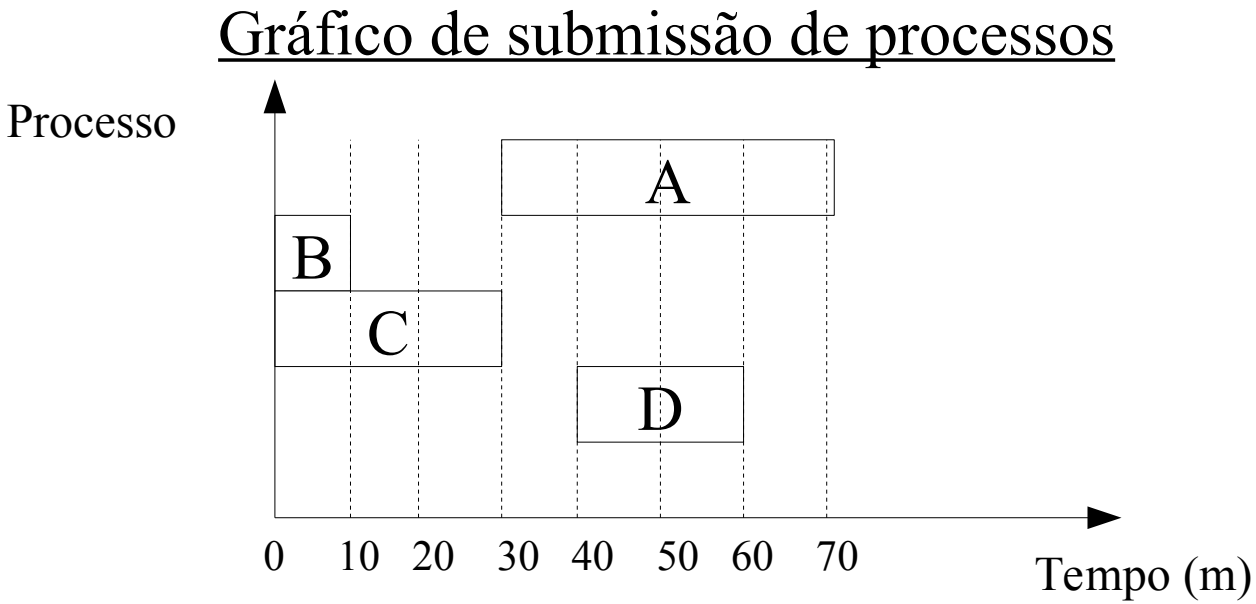
$$tr_A = 40\,m$$
$$tr_B = 60\,m$$
$$tr_C = 60\,m$$
$$tr_D = 50\,m$$
$$\bar{tr} = 52{,}5\,m$$

$$vazão = \frac{4\,tarefas}{2\,horas}$$
$$vazão = 2\,tarefas/hora$$

Escalonamento em sistemas de lote
Shortest Job First

- **Processos são alocados em ordem de tempo de computação**
- **Vantagens:**
  - Tempo de retorno melhor
- **Desvantagens:**
  - Como calcular tempo de execução?
  - Tarefas devem estar prontamente disponíveis
  - Injusto

Escalonamento em sistemas de lote
Shortest Job First



Considerando que:

- A=40m, B=10m, C=30m, e D=20m
- Inícios como na figura

Qual é o :

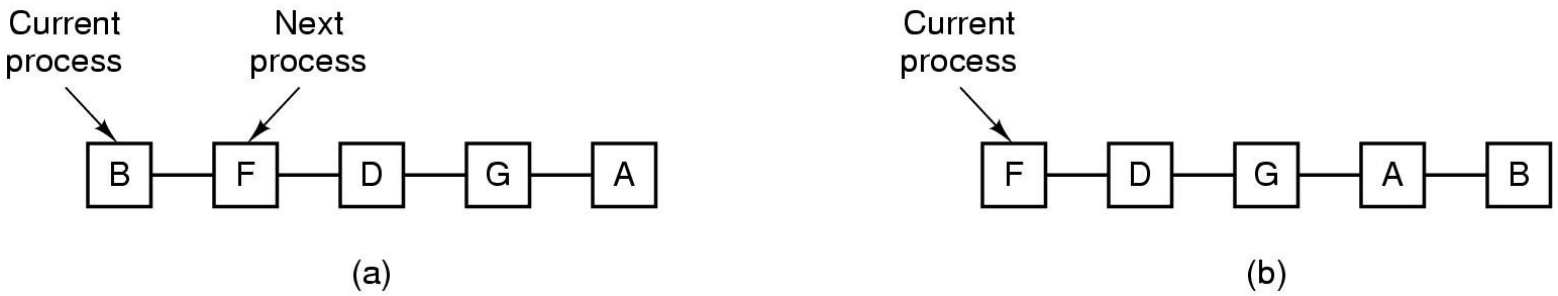
- Tempo médio de retorno?
- Vazão?

Ordem de escalonamento?

Escalonamento em sistemas interativos
Objetivos

- **Tempo de resposta**
  - Responder rapidamente às requisições
- **Proporcionalidade**
  - Satisfazer as expectativas dos usuários

Escalonamento em Sistemas Interativos
Round-Robin



- **Chaveamento Circular**
  - Mantém lista de processos prontos
  - A cada processo é dado um **quantum**
  - Ao término do quantum, processo sofre preempção

Escalonamento em sistemas interativos
Round-Robin

- **Questões importantes:**
  - Tamanho do Quantum
  - Chaveamento de Contexto
- **Duas situações:**
  - 1 ms chaveamento / 4 ms de quantum
  - 1 ms chaveamento / 100ms de quantum

$$U_{teórica} = \frac{quantum}{chaveamento + quantum}$$

$$U_{real} = \frac{tempo\,útil}{tempo\,transcorrido}$$

Escalonamento em sistemas interativos
Round-Robin

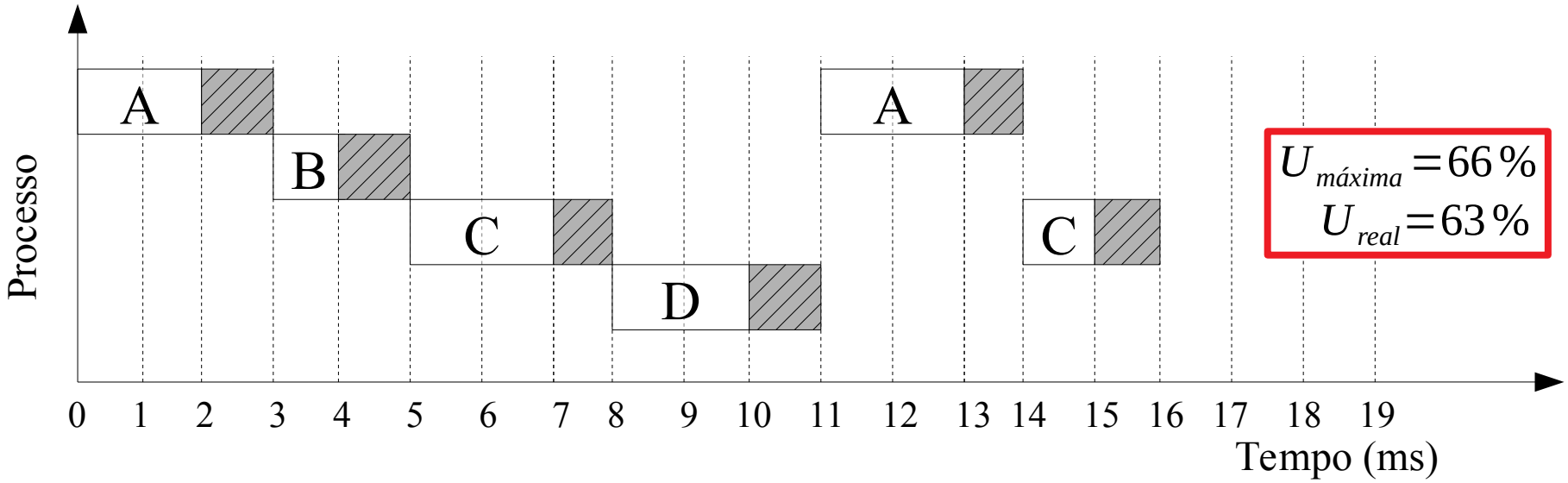
Sejam os tempos de execução dos seguintes processos:

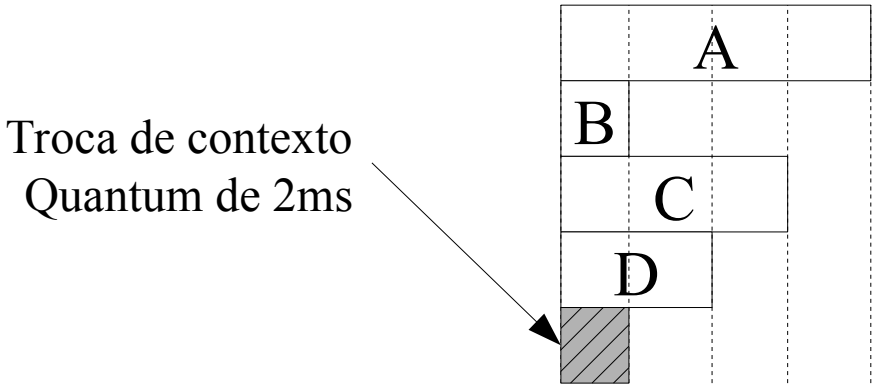
- **A=4 ms**
- **B=1 ms**
- **C=3 ms**
- **D=2 ms**

Considere, ainda, que esteja sendo utilizado um escalonamento por Round-Robin com o quantum valendo 2 ms e se gastando 1 ms para realizar uma troca de contexto.

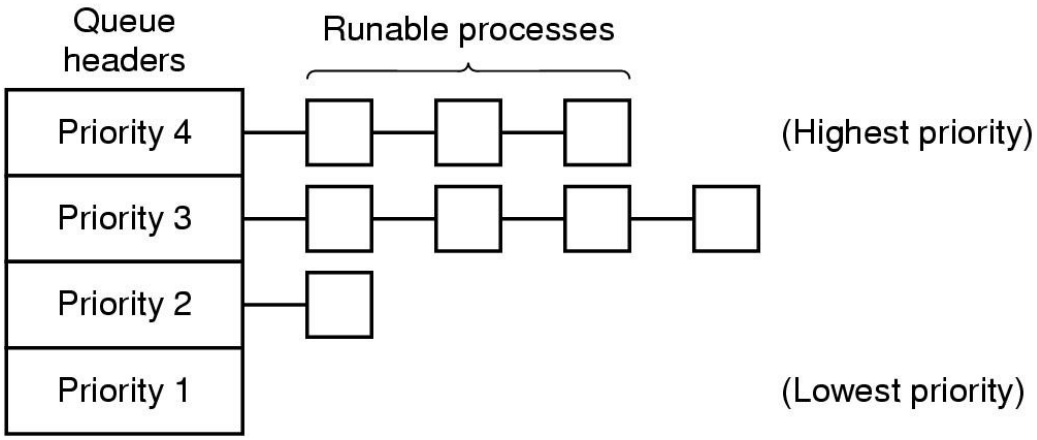
1) Esboce um gráfico de processo x tempo, mostrando qual processo está sendo executado em um dado instante de tempo.
2) Quais são as taxas de utilização do processador teórica e da carga de trabalho realmente executada?

Escalonamento em sistemas interativos
Round-Robin





Escalonamento em sistemas Interativos
Round-Robin com prioridades

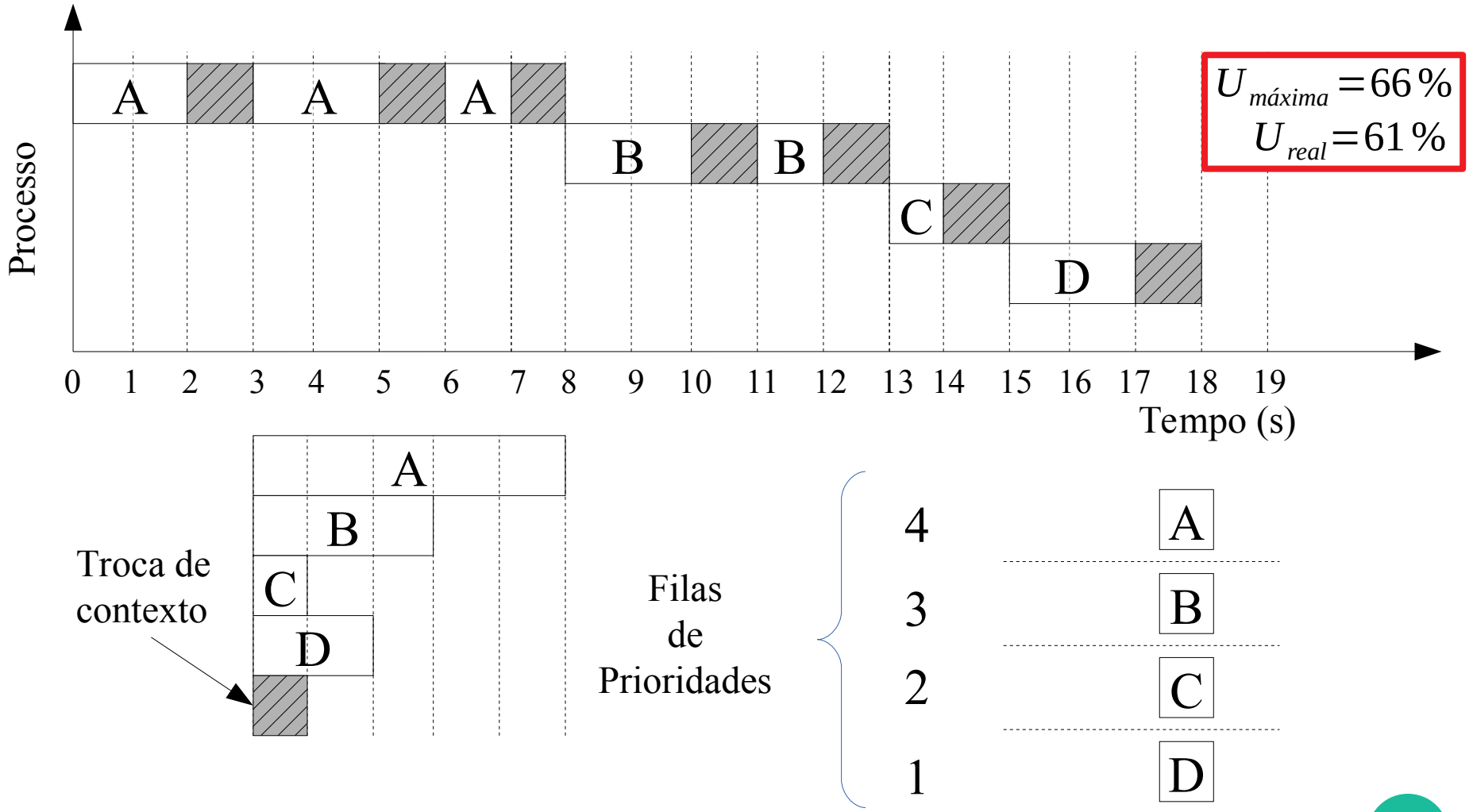


- **Semelhante ao Round-Robin, mas processos com prioridade maior são executados antes**
- **Diferentes implementações são possíveis:**
  - Prioridades fixas
  - Decaimento de prioridade a cada quantum

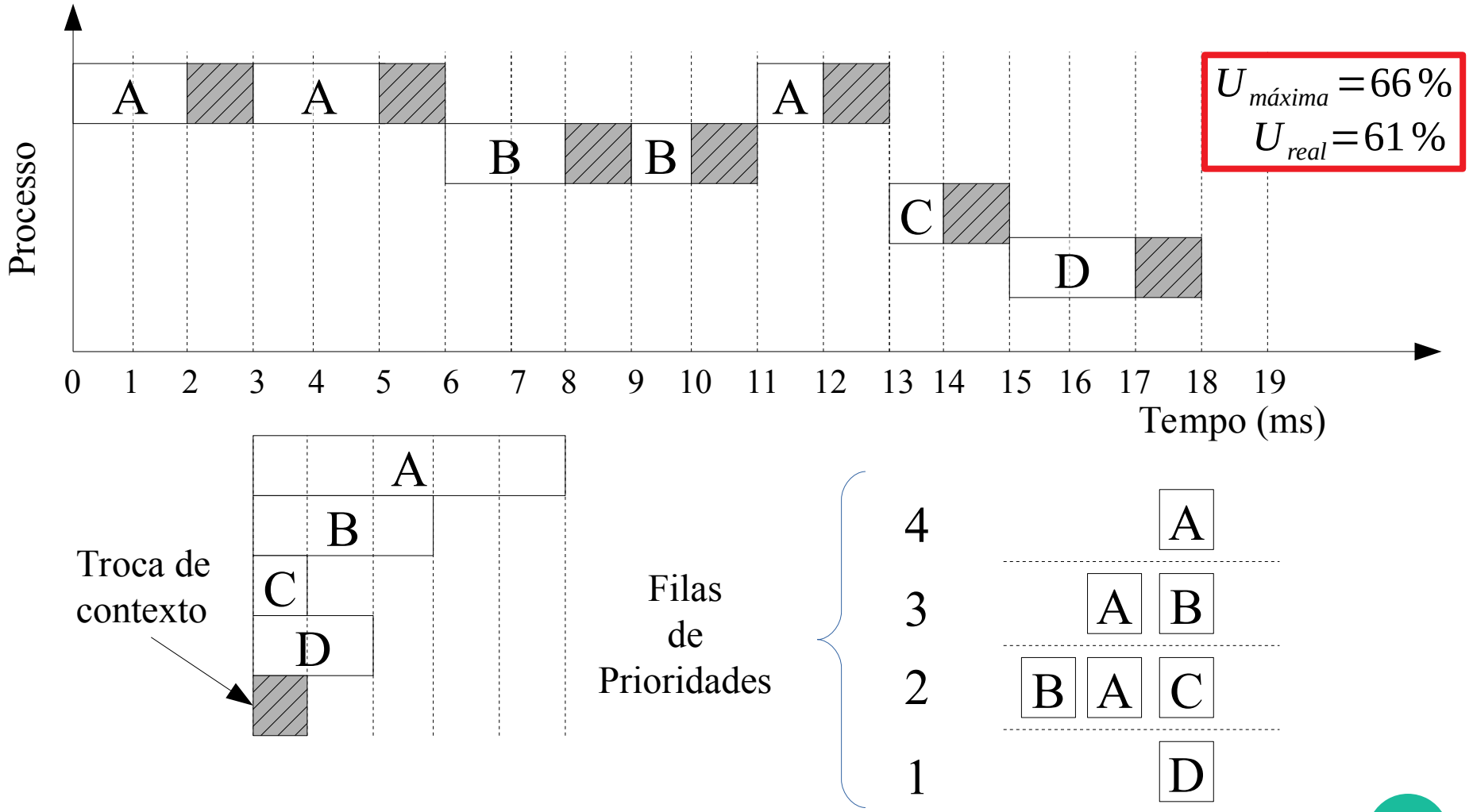
Escalonamento em sistemas interativos
Round-Robin com prioridades

- **Sejam os processos:**
  - A, com 5 ms, e prioridade 4
  - B, com 3 ms. e prioridade 3
  - C, com 1 ms. e prioridade 2
  - D, com 2 ms. e prioridade 1
  - Quantum com 2 ms.
  - Trocas de contexto com 1 ms.
- **Situação 1:**
  - Processos de menor prioridade devem esperar pelo término dos processos de maior prioridade
- **Situação 2:**
  - Decaimento da prioridade em 1 após execução do quantum, colocando o processo *no início da próxima fila de prioridade*

Escalonamento em sistemas interativos
Round-Robin com prioridades (fixo)



Escalonamento em sistemas interativos
Round-Robin com prioridades (decaimento)



Exercício

- **Sejam os processos:**
  - A, com 5 ms., e prioridade 1
  - B, com 3 ms. e prioridade 3
  - C, com 1 ms. e prioridade 2
  - D, com 2 ms. e prioridade 3
  - Quantum com 2 ms.
  - Trocas de contexto com 1 ms
- **Responda:**

a) Faça um gráfico do escalonamento dos processos acima por prioridades, considerando decaimento da prioridade em 1 após execução do quantum, colocando o processo *no final da próxima fila de prioridade*

b) Qual é a taxa de utilização *máxima* desse sistema?

c) Qual é a taxa de utilização real desse sistema?

Escalonamento em Sistemas de Tempo Real
Objetivos

- **Cumprimento de prazos**
  - *Soft* real time
    - Evitar degradação da experiência do usuário (mídias)
    - Evitar perda de dados
  - *Hard* real time
    - Evitar falhas catastróficas
- **Previsibilidade**

Escalonamento em Sistemas de Tempo Real

Um sistema de tempo-real é escalonável se:

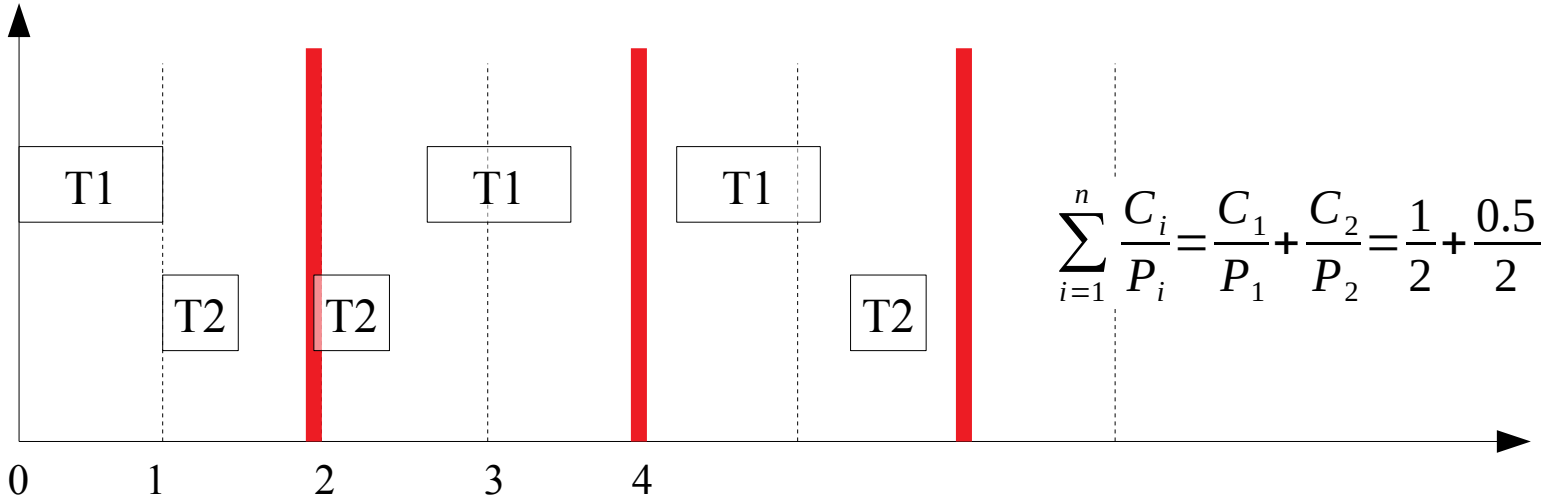
- **Dados:**
  - *m* eventos periódicos
  - Evento *i* ocorre dentro de um período P_i e requer C_i segundos
- **Então a carga só pode ser executada se:**

$$Carga = \sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1$$

Escalonamento em Sistemas de Tempo Real

- **Exemplo que funciona:**

	Tarefa 1	Tarefa 2
C (s)	1	0.5
P (s)	2	2

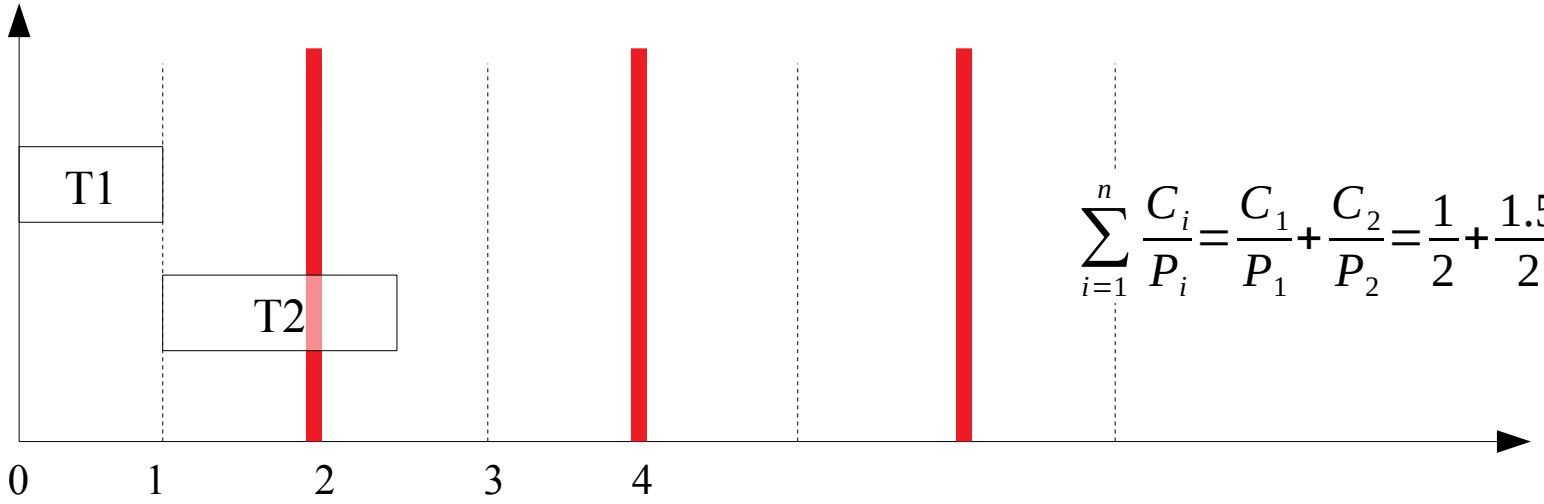


$$\sum_{i=1}^{n} \frac{C_i}{P_i} = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{0.5}{2} = 0{,}75 \leq 1$$

Escalonamento em Sistemas de Tempo Real

- **Exemplo que não funciona:**

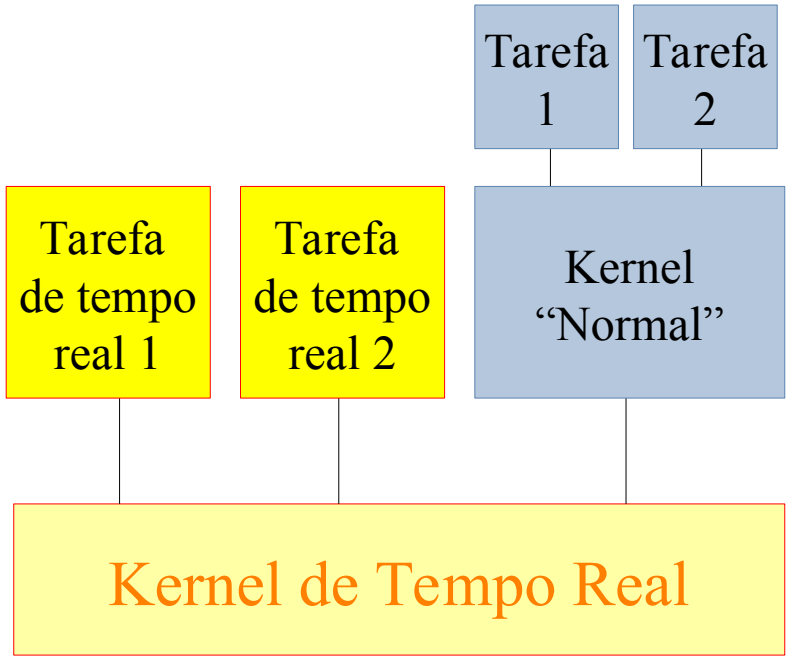
	Tarefa 1	Tarefa 2
C (s)	1	1.5
P (s)	2	2



$$\sum_{i=1}^{n} \frac{C_i}{P_i} = \frac{C_1}{P_1} + \frac{C_2}{P_2} = \frac{1}{2} + \frac{1.5}{2} = 1{,}25 > 1$$

Escalonamento em Sistemas de Tempo Real
Estrutura do Kernel

- **Para prover tempo real de fato, o kernel deve ser *preemptável***
- **Sistemas operacionais de tempo real:**
  - *VxWorks*
  - *RTEMS*
  - *RTOS*
  - *SCHED_DEADLINE no kernel Linux* *



* WARNING! Fiddling with these settings can result in an unpredictable or even unstable system behavior. As for -rt (group) scheduling, it is assumed that root users know what they're doing.

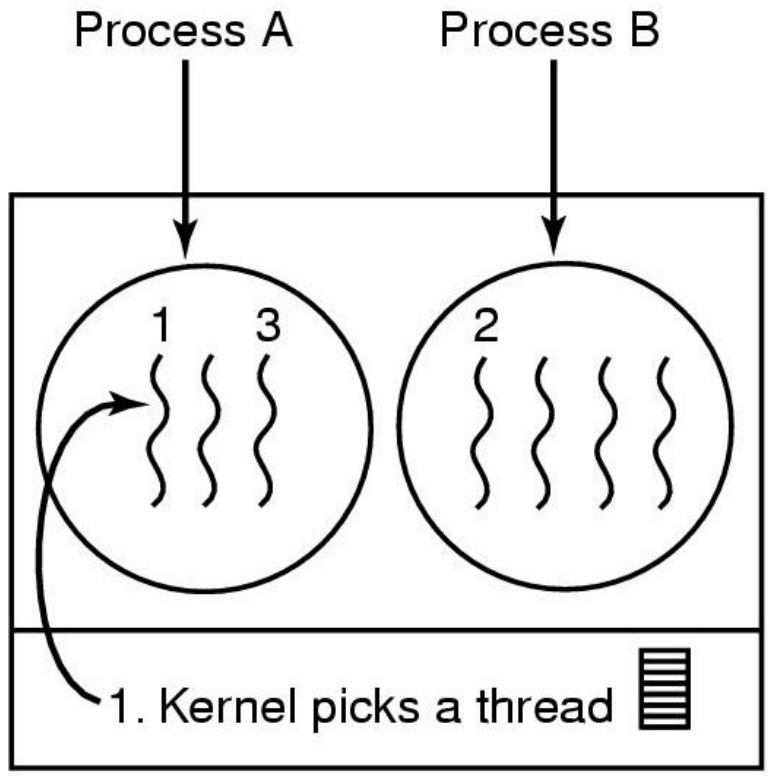
Ars Technica. Definitely not Windows 95: What operating systems keep things running in space?.

Política *versus* Mecanismo

- **Situação: nem sempre os processos são todos de usuários diferentes.**
- **Solução: algoritmo de escalonamento é parametrizado**
  - Separa o que é permitido fazer de como é ele implementado
  - Mecanismo no kernel
  - Programa *nice* no Linux
- **Parâmetros preenchidos pelos processos de usuário**
  - Política configurada pelo processo de usuário

Escalonamento de Threads
Threads de Kernel

- **Considerando:**
  - Quantum de 50ms
  - Threads trabalham por 5ms antes de passar a vez
- **É possível alternar para threads de outros processos dentro do quantum**



Possible: A1, A2, A3, A1, A2, A3
Also possible: A1, B1, A2, B2, A3, B3

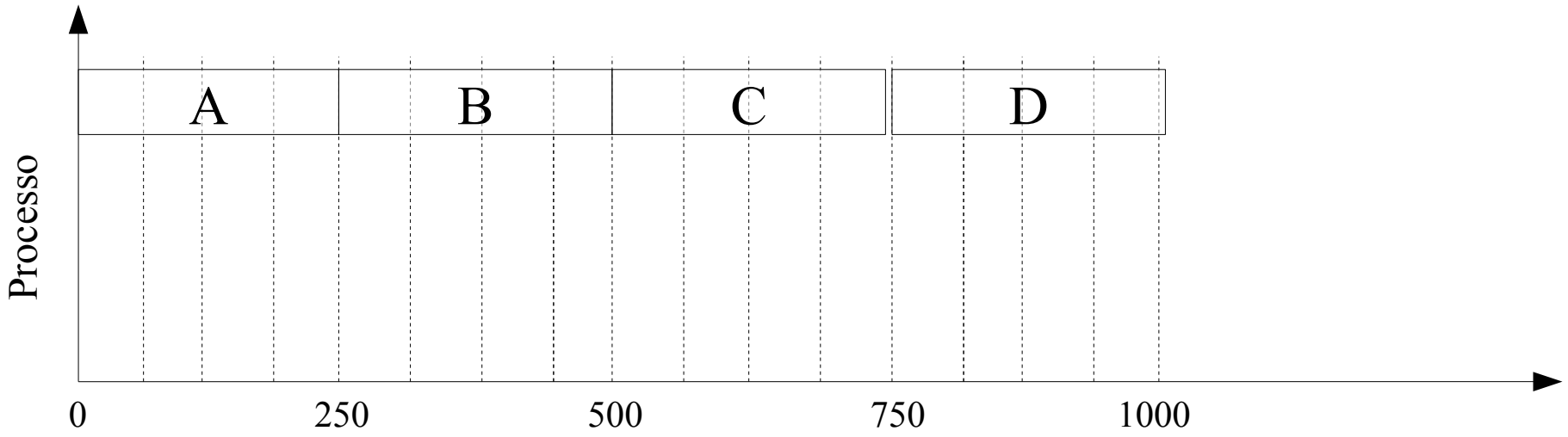
Estudo de caso: Linux

- **V6.6: Earliest Eligible Virtual Deadline First (EEVDF)**
  - Substitui o antigo *Completely Fair Scheduler* (CFS)
- **Ideia básica é dar prioridade aos processos que não conseguiram usar toda sua fatia de tempo**

(STOICA; ABDEL-WAHAB, 1995)

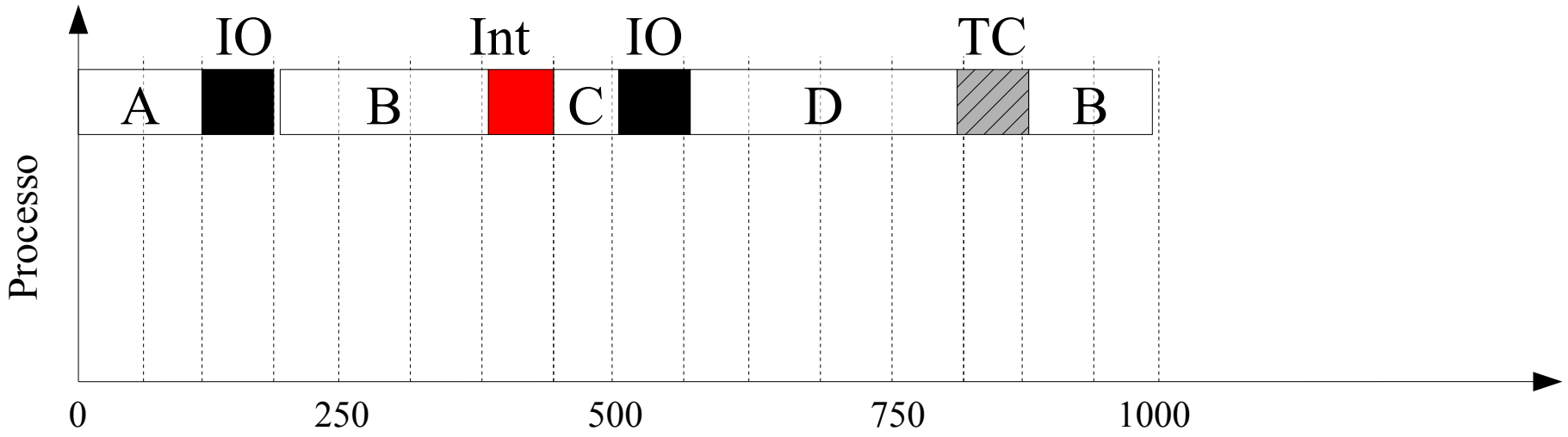
Estudo de caso: Linux

- **Mundo ideal:**
  - 4 processos A, B, C e D
  - Fatia de 250ms para cada um



Estudo de caso: Linux

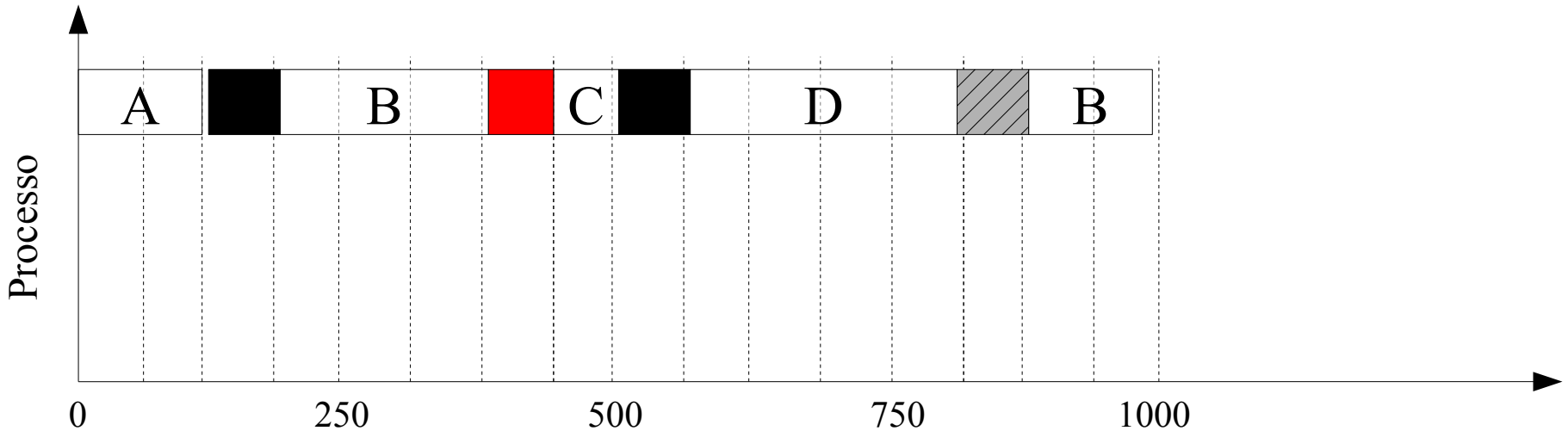
- **Mundo real:**
  - 4 processos A, B, C e D
  - Perdas por Interrupções, trocas de contexto, IO, etc



Estudo de caso: Linux

- **Mundo real:**
  - 4 processos A, B, C e D
  - Perdas por Interrupções, trocas de contexto, IO, etc

$$Lag = fatia_{teorica} - fatia_{recebida}$$

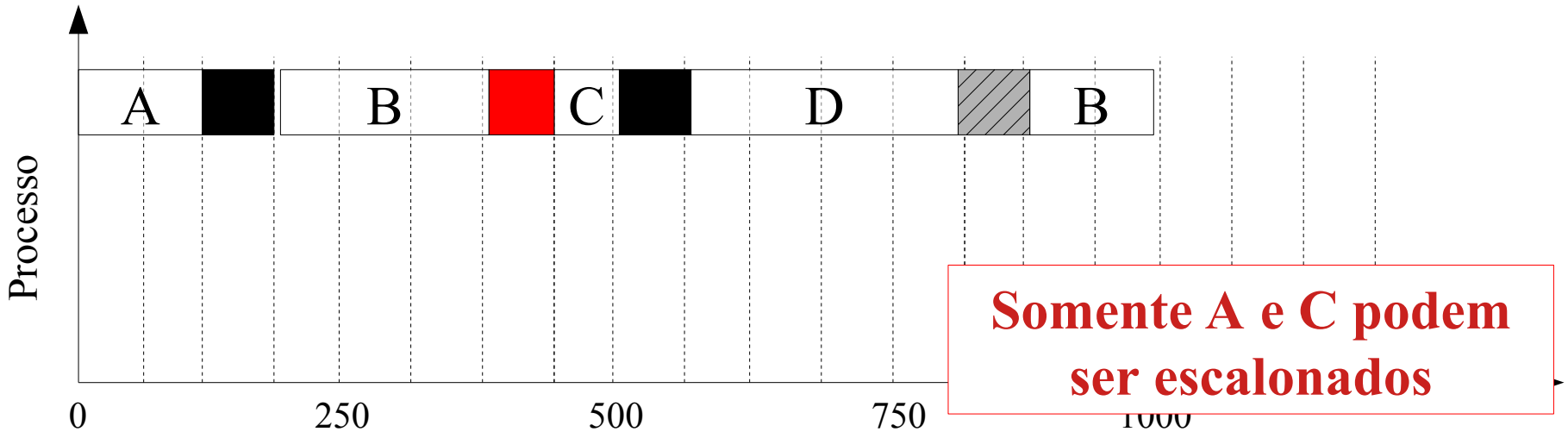


Estudo de caso: Linux

- **Mundo real:**
  - 4 processos A, B, C e D
  - Perdas por Interrupções, t… o, IO, etc

$$Lag = fatia_{teorica} - fatia_{recebida}$$

$$Lag_A = 250 - 125 = 125$$
$$Lag_B = 250 - 312{,}5 = -62{,}5$$
$$Lag_C = 250 - 62{,}5 = 187{,}5$$
$$Lag_D = 0$$



Somente A e C podem ser escalonados

Referências

- **TANENBAUM, A. S. *Sistemas Operacionais Modernos. 3ª. ed.***
  - Capítulo 2: seção 2.4.1, 2.4.3, 2.4.4, 2.4.5, 2.4.6
- **Slides originais de Andrew S. Tanenbaum**
  - http://www.cs.vu.nl/~ast/books/book_software.html

Referências

I. Stoica and H. Abdel-Wahab. 1995. Earliest Eligible Virtual Deadline First : A Flexible and Accurate Mechanism for Proportional Share Resource Allocation. Technical Report. Old Dominion University, USA.

Referências

- https://www.phoronix.com/review/linux-66-features
- https://kernelnewbies.org/Linux_6.6

Exercícios

- **Capítulo 2:**
  - 33, 35, 36, **37,** 41