UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

ESCALONAMENTO DE PROCESSOS

Sistemas Operacionais
Engenharia de Software
Prof. Jeferson Silva

PORQUE É NECESSÁRIO ESCALONAR?

Processos precisam ser executados!

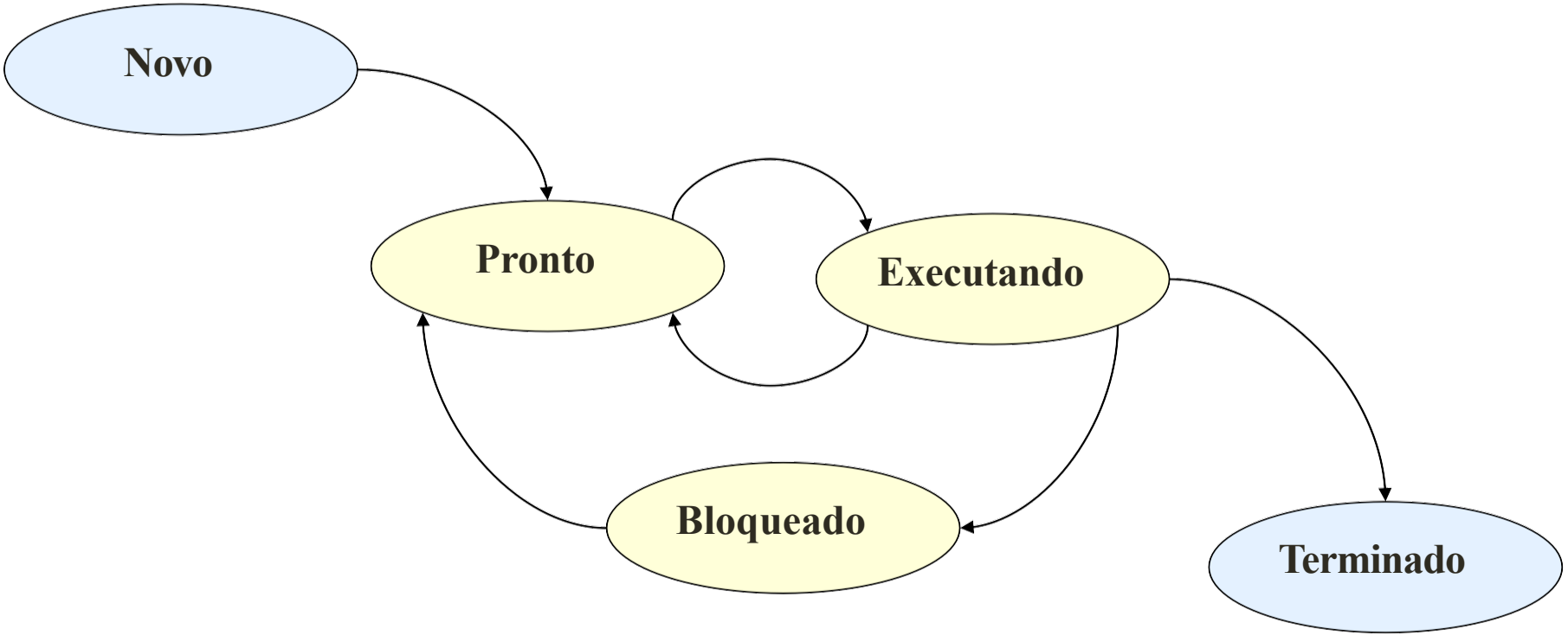
Escalonador:

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 O DESEMPENHO DE UM ALGORITMO DE ESCALONAMENTO?

Cada processo possui informações que permitem definir precisamente seu estado.

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

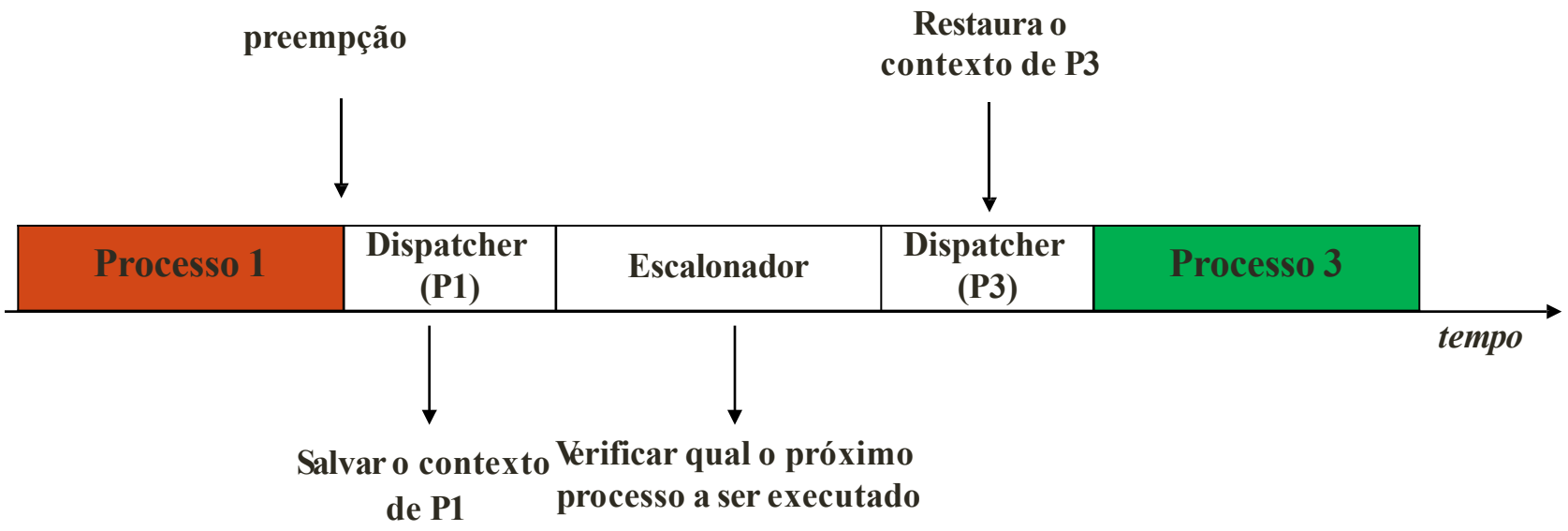
Troca de Contexto

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?

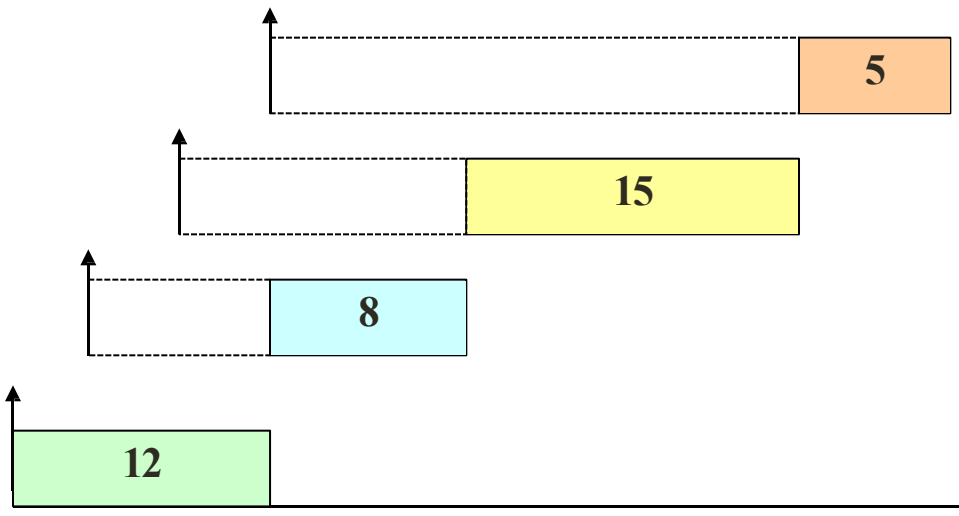
Independente do *tipo* de sistema operacional, TODOS os algoritmos de escalonamento precisam atender a alguns critérios:

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

ESCALONAMENTO PARA SISTEMAS EM LOTE

- FCFS (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

FCFS

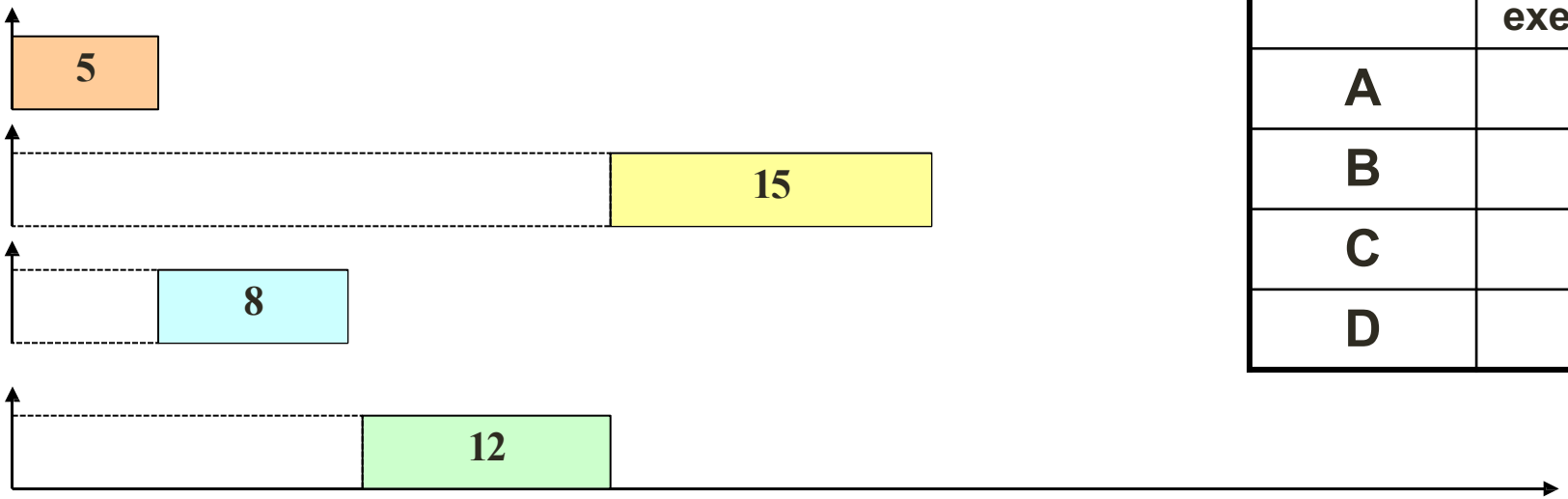


Processo	Custo de execução	Instante de chegada
A	12	t = 0
B	8	t = 3
C	15	t = 5
D	5	t = 10

ESCALONAMENTO PARA SISTEMAS EM LOTE

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

SJF – *SHORTEST JOB FIRST*



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

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

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
- A implementação de filas pode representar um problema!

ESCALONAMENTO EM SISTEMAS DE PROPÓSITO GERAL

Filas Múltiplas

- Processos executam dentro de uma fatia de tempo predefinida (***quantum***)
- Preemptivo
- Justo
- Tamanho do *quantum* variável ➔ trocas de contexto.
- Adaptável para diferentes tamanhos de processo
- Os processos são promovidos a medida que o tempo passa