

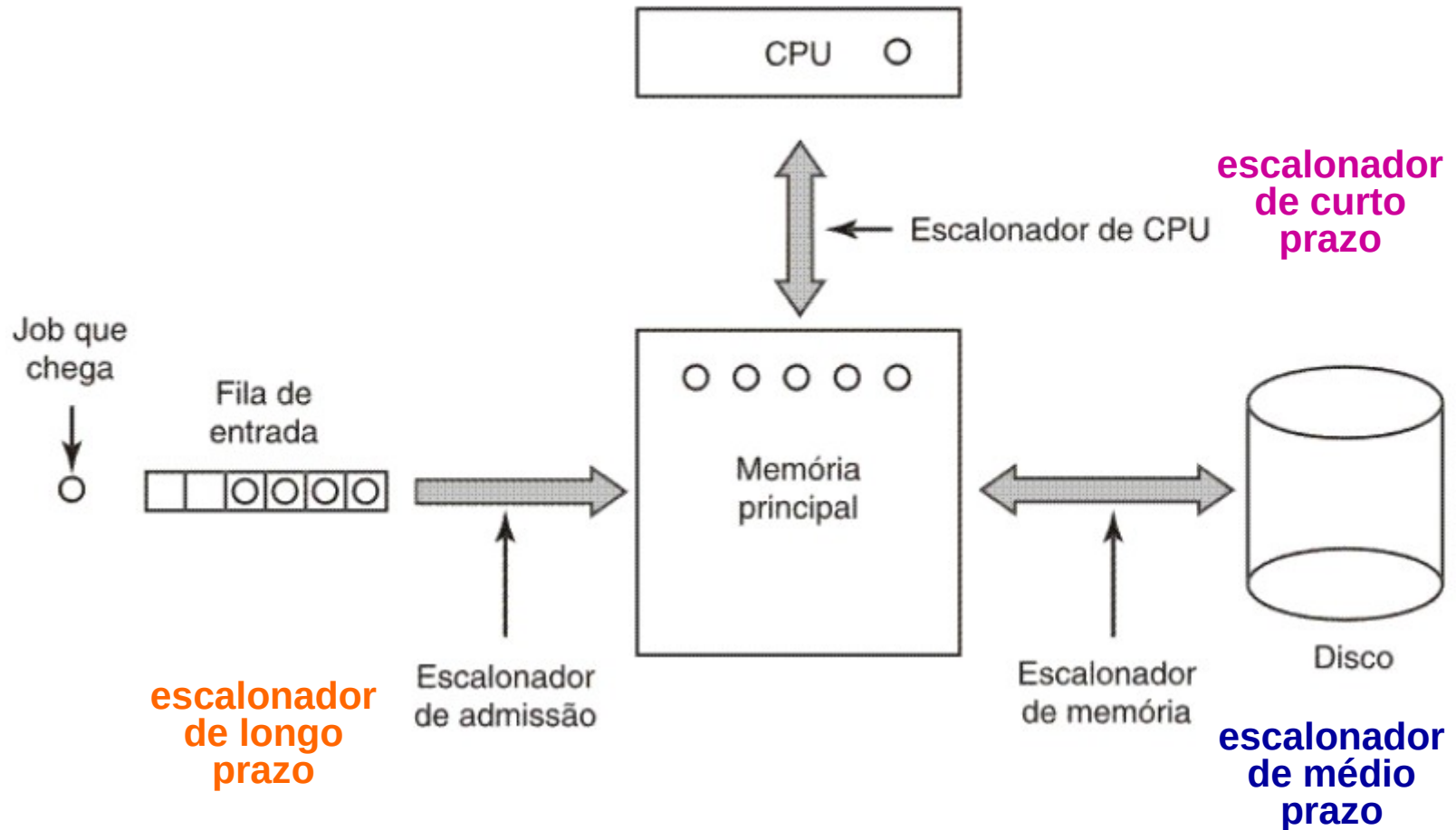
Escalonamento

Fontes:
Silberschatz cap 3
Tanenbaum cap 2

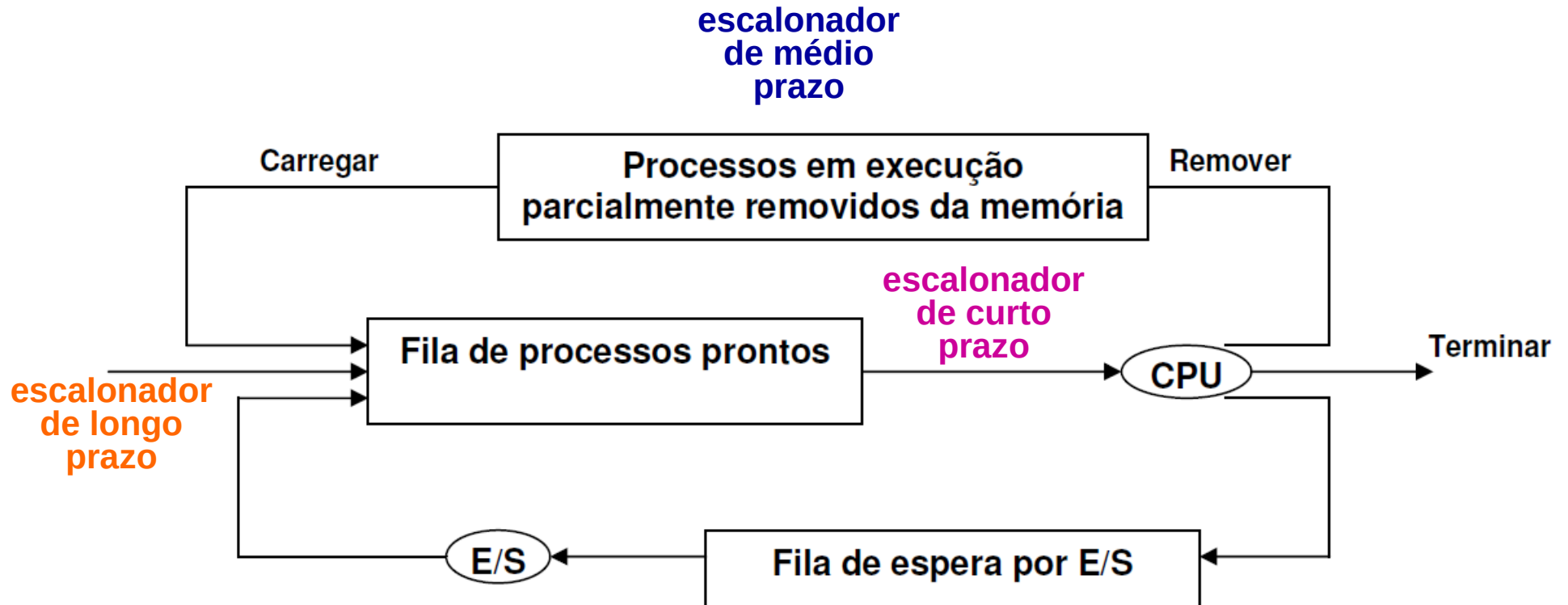
- *Scheduling*
 - função fundamental do SO
 - quase todos os recursos do computador são submetidos ao scheduling antes do uso
- Níveis
 - Escalonador de longo prazo
 - Escalonador de médio prazo
 - **Escalonador de curto prazo**



Escalonadores (1)



Escalonadores (2)



Escalonador de longo prazo (1)

- **job scheduler/long-term scheduler**
 - gerencia a criação dos processos
 - seleciona processos a serem levados p/ **fila de prontos** (admissão ao sistema)
 - *I/O-bound x CPU-bound*
 - diversos processos I/O-bound → fila ready quase sempre vazia
 - diversos processos CPU-bound → fila de dispositivos quase sempre vazia
 - pouco invocado (segundos, minutos) → pode ser lento
 - memória secundária → MP
 - controla **grau de multiprogramação** (nº de processos em memória)
 - balanceamento de carga
-

Escalonador de longo prazo (2)

- **job scheduler/long-term scheduler**
 - quanto maior o nro de processos admitidos, menor a porcentagem de tempo de CPU para cada um
 - em alguns sistemas, o escalonador de longo prazo pode não existir ou ser muito simples
 - Ex.: sistemas time-sharing inserem novos processos diretamente na fila ready – curto prazo
 - pouco utilizado em SOs hoje em dia

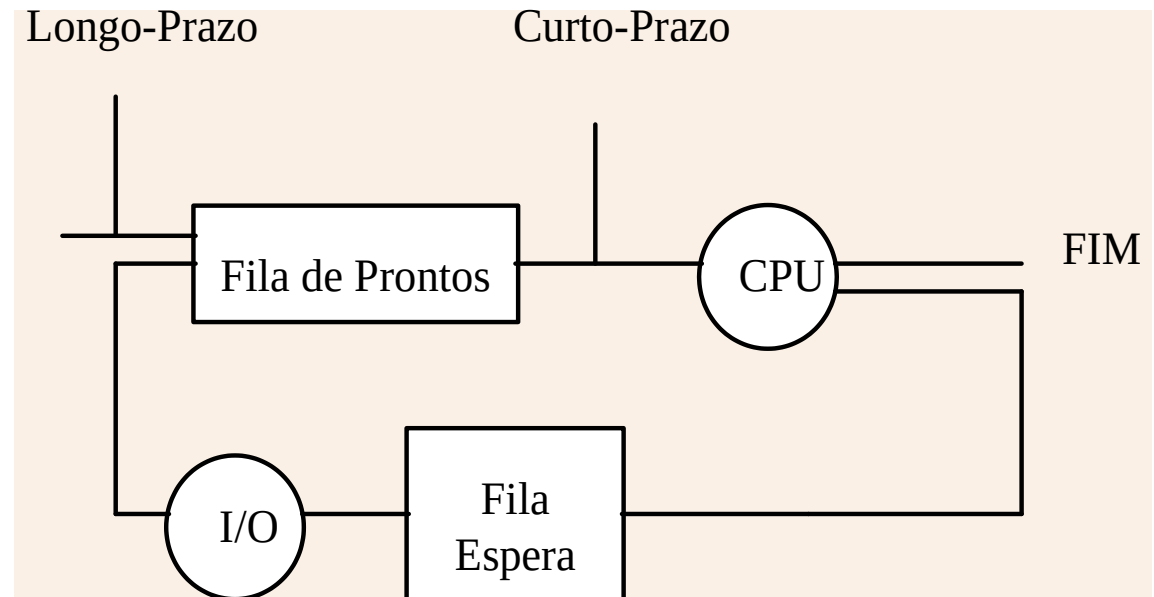
Escalonador de médio prazo

- **medium-term scheduler**
 - comum em sistemas com memória virtual
 - gerencia a memória
 - memória → disco
 - escalonador de memória
 - swapper
 - associado ao **swapping** de processos
 - objetivo: liberar espaço em memória
 - remove processos que aguardam evento p/ área de swap
 - permite que novos processos sejam executados

Escalonador de curto prazo (1)

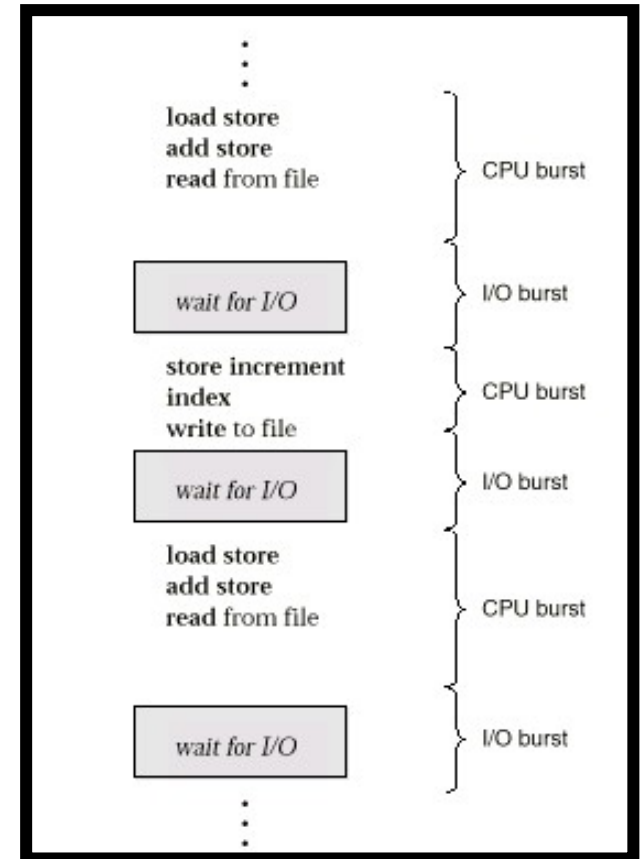
- **CPU scheduler/short-term scheduler**

- atua junto ao **dispatcher**
- decide a alocação da CPU (seleciona o processo a **executar** na CPU)
- MP → processador
- invocado frequentemente (milisegundos)
- deve ser rápido



Escalonador de curto prazo (2)

- *Escalonamento da CPU*
- processos obedecem um ciclo que alterna entre processamento e espera por E/S
- **distribuição de ciclos** de CPU
- preocupação: **troca de contexto**



Obrigações do escalonador da CPU

- Escolher o processo a ser executado
- Fazer uso **eficiente** da CPU
 - Alternar entre processos
 - alto custo computacional
 - Tarefas:
 - Salvar registradores do processo na tabela de processos
 - Salvar conteúdo da cache
 - Selecionar o novo processo
 - Inicializar o novo processo (ex. carga das informações da tabela de processo na memória)

Dispatcher

- módulo que dá o **controle** da CPU ao processo selecionado pelo scheduler de curto prazo
- deve ser rápido
- envolve:
 - troca de contexto
 - troca p/ modo usuário
 - desvio p/ ponto exato no programa do usuário p/ reiniciá-lo
- latência de dispatcher
 - tempo entre parar um processo e iniciar outro

Preempção

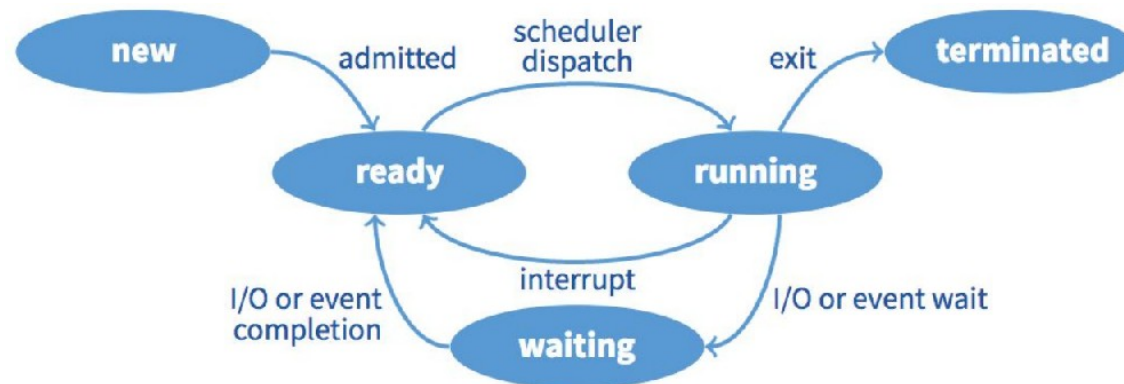
- Em SO, **preempção** é a capacidade de alterar a ordem de execução de um processo em detrimento de outro, de acordo com algum critério.
 - Preempção por tempo
 - término de fatia
 - tempo de ciclo de CPU
 - Preempção por prioridade
- Essencial para sistemas interativos → time-sharing

Tipos de escalonadores

- Não preemptivos
 - processo executa até terminar OU
 - processo executa até se bloquear (ex. E/S)
- Preemptivos
 - o escalonador escolhe um processo, que executa até quando o escalonador quiser OU
 - processo deixa a CPU quando se bloqueia OU
 - processo deixa a CPU quando termina

Escalonador da CPU - *short-term scheduler*

- decisões de escalonamento ocorrem quando um processo troca do estado:
 1. “*running*” p/ “*waiting*”
 2. “*running*” p/ “*ready*”
 3. “*running*” p/ “*terminated*”
 4. “*waiting*” p/ “*ready*”



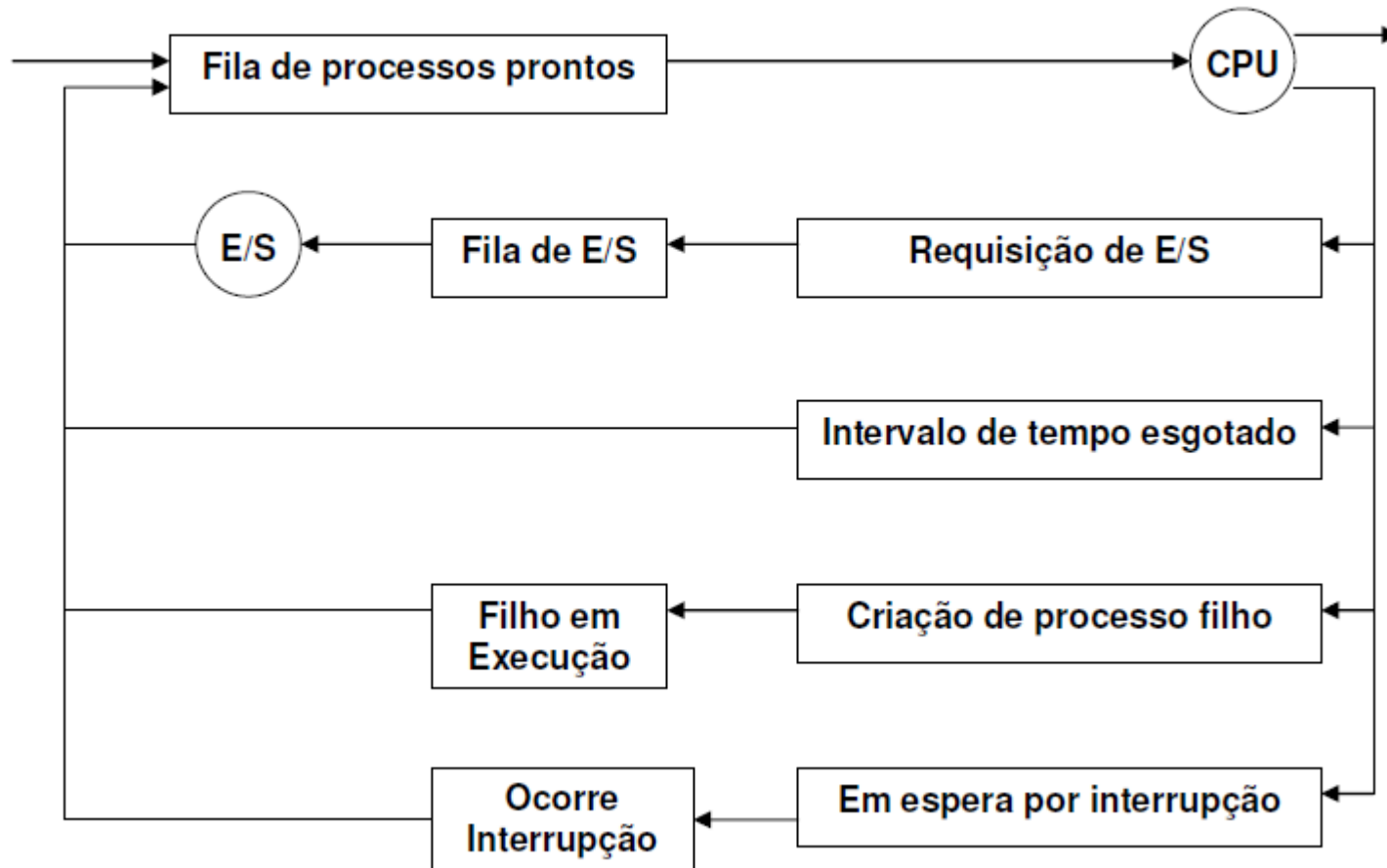
Exemplos de situações típicas para execução do escalonador (1)

- Dependem se o escalonador é preemptivo ou não preemptivo, se considera prioridades ou não...
 - Sempre que a CPU estiver livre e houver processos aptos a executar
 - Criação e término de processos
 - Um processo de mais alta prioridade fica apto a executar
 - Interrupção de tempo
 - Ex.: processo executou toda a fatia

Exemplos de situações típicas para execução do escalonador (2)

- Interrupção de dispositivos de E/S
- Interrupção por falta de página (ou segmento) em memória
 - Endereço acessado não está carregado na memória (está em disco)
- Interrupção por erro

Diagrama de transição entre filas



Objetivos do escalonador (1)

- Maximizar o uso da CPU
 - manter a CPU ocupada o máximo possível
- Maximizar o *throughput* (vazão do sistema)
 - *throughput* - n° de processos completados por intervalo de tempo
- Minimizar o tempo de espera (*waiting time*)
 - soma dos tempos que um processo espera na fila de prontos

Objetivos do escalonador (2)

- Minimizar o tempo de *turnaround* (t de execução)
 - *turnaround* - tempo desde a submissão até o término da execução do processo
 - soma os tempos de espera p/ ocupar a memória, espera na fila de prontos, execução na CPU e realizando I/O
 - limitado pela velocidade do dispositivo de E/S
- Minimizar o tempo de resposta (*response time*)
 - tempo desde a submissão de um pedido até que a primeira resposta seja produzida

Objetivos do escalonador (3)

- Justiça
 - Se refere à distribuição do processador entre os processos
 - 2 processos com comportamento e prioridade similares devem ter durações de execução similares
- Eficiência
 - Depende da rapidez da troca de contexto e da quantidade de tarefas I/O-bound