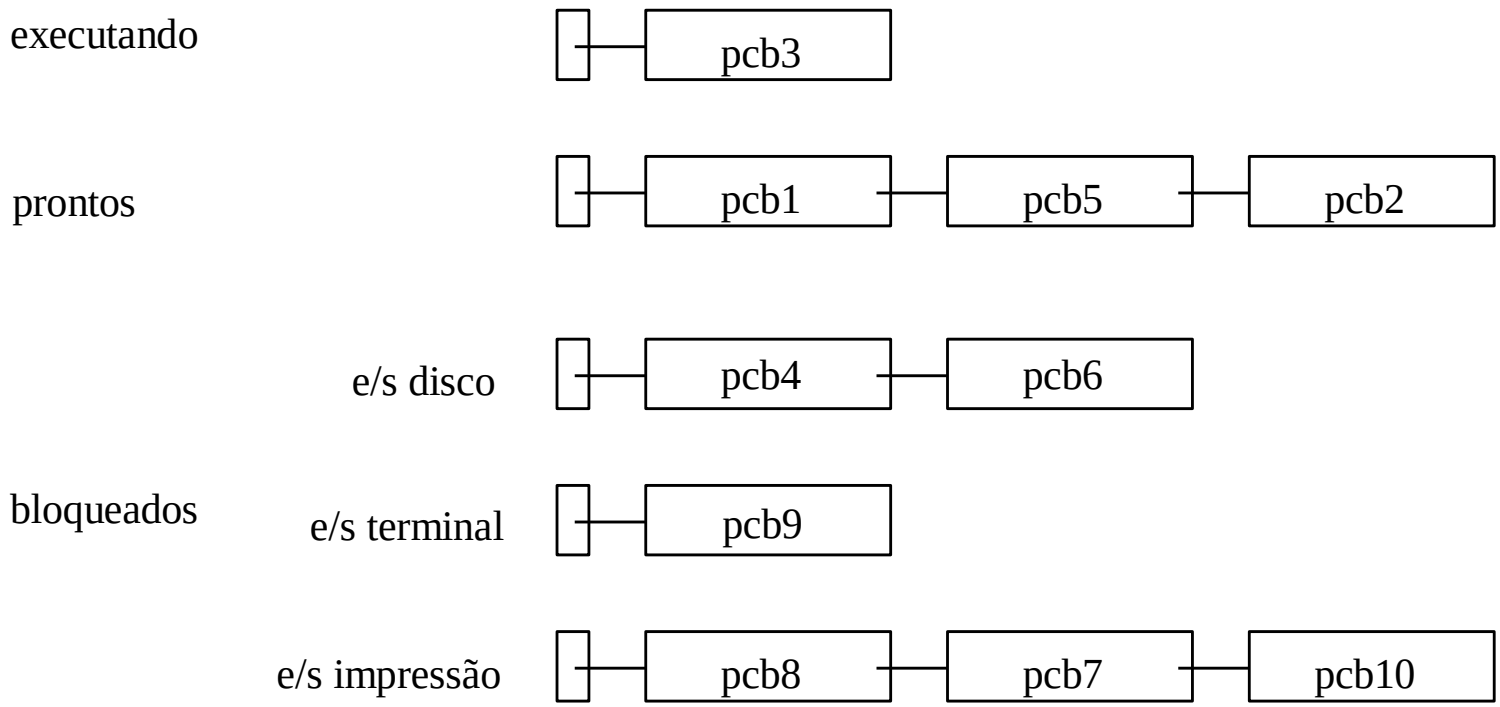
Processos

Prof. Dr. João Carlos Lopes Fernandes
E-mail: joao.fernandes1@docente.unip.br

Cada estado, tem uma fila controlada pelo PCB

(Proccess control block)



Razões para Suspender Processos

Swapping: para liberar espaço na memória principal para trazer outro processo da memória secundária, o SO pode suspender um processo:

- em *background;*
- utilitário;
- suspeito de estar causando problemas;

Solicitação de usuário interativo;
Temporização: determinados processos são executados periodicamente; (Solicitação do processo pai).

Conceito de Escalonamento

Escalonamento consiste em determinar, dentre os processos prontos, qual o próximo processo a ser executado;

Realizado por um componente do sistema operacional denominado escalonador;

Dois tipos de escalonadores:

- longo prazo;
- curto prazo

Escalonador longo prazo:

memória secundária → memória principal

Escalonador curto prazo:

memória principal → processador

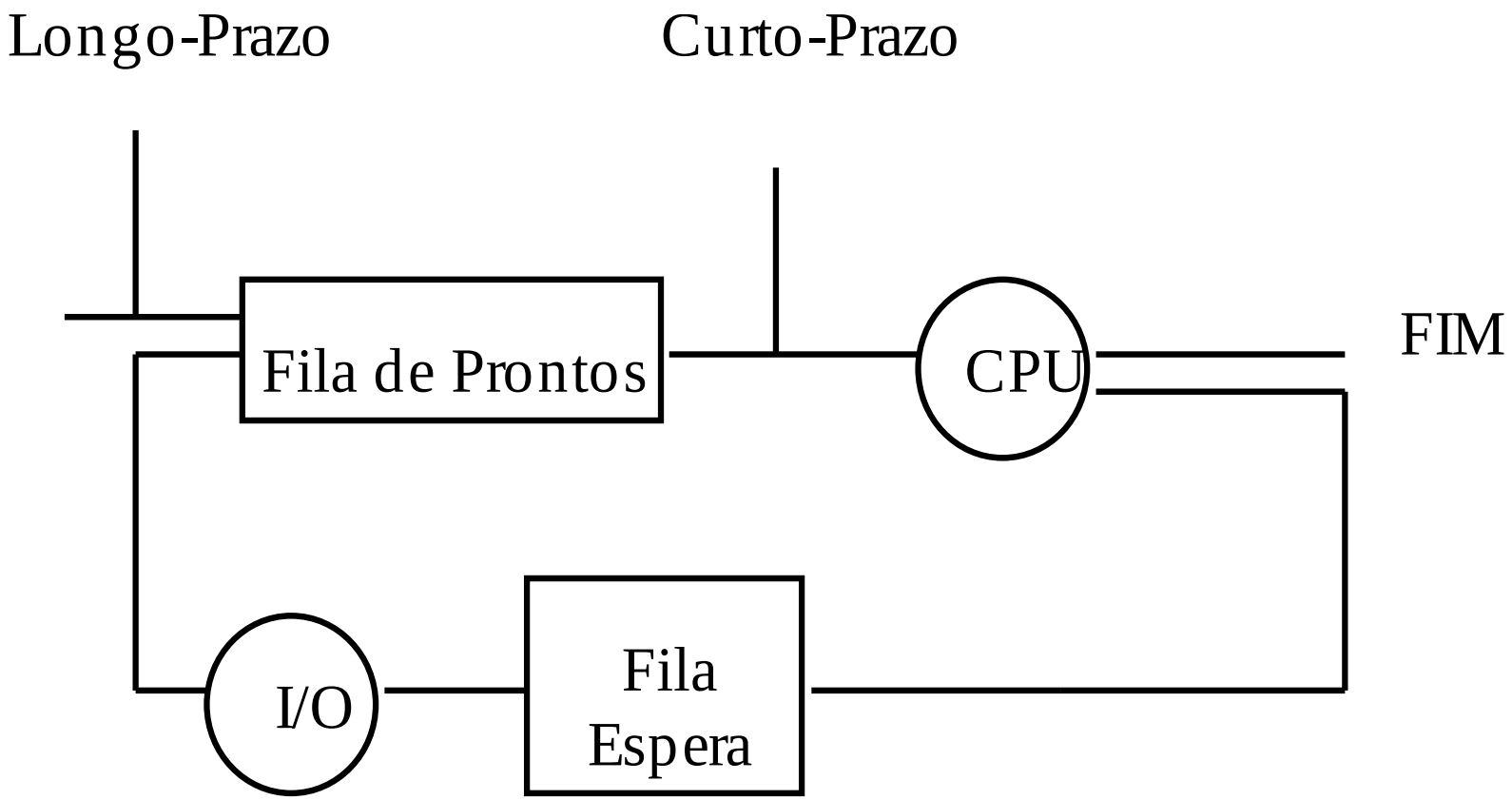
Principais objetivos:

maximizar a utilização do processador;

maximizar o número de processos completados por unidade de tempo;

garantir que todos os processos recebam o processador;

minimizar o tempo de resposta para o usuário.


Longo-Prazo
Curto-Prazo
Fila de Prontos
CPU
FIM
I/O
Fila
Espera

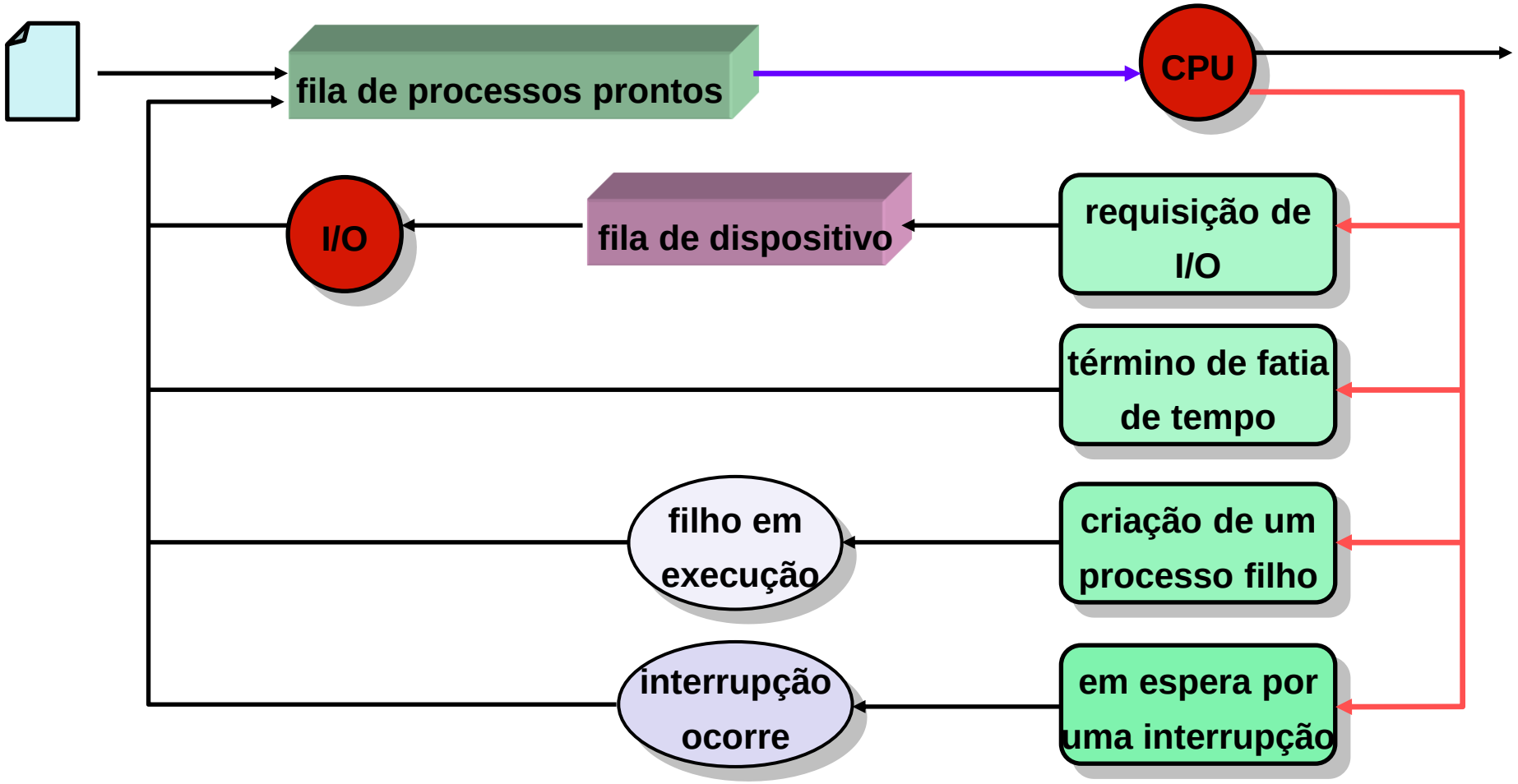
Dispatcher

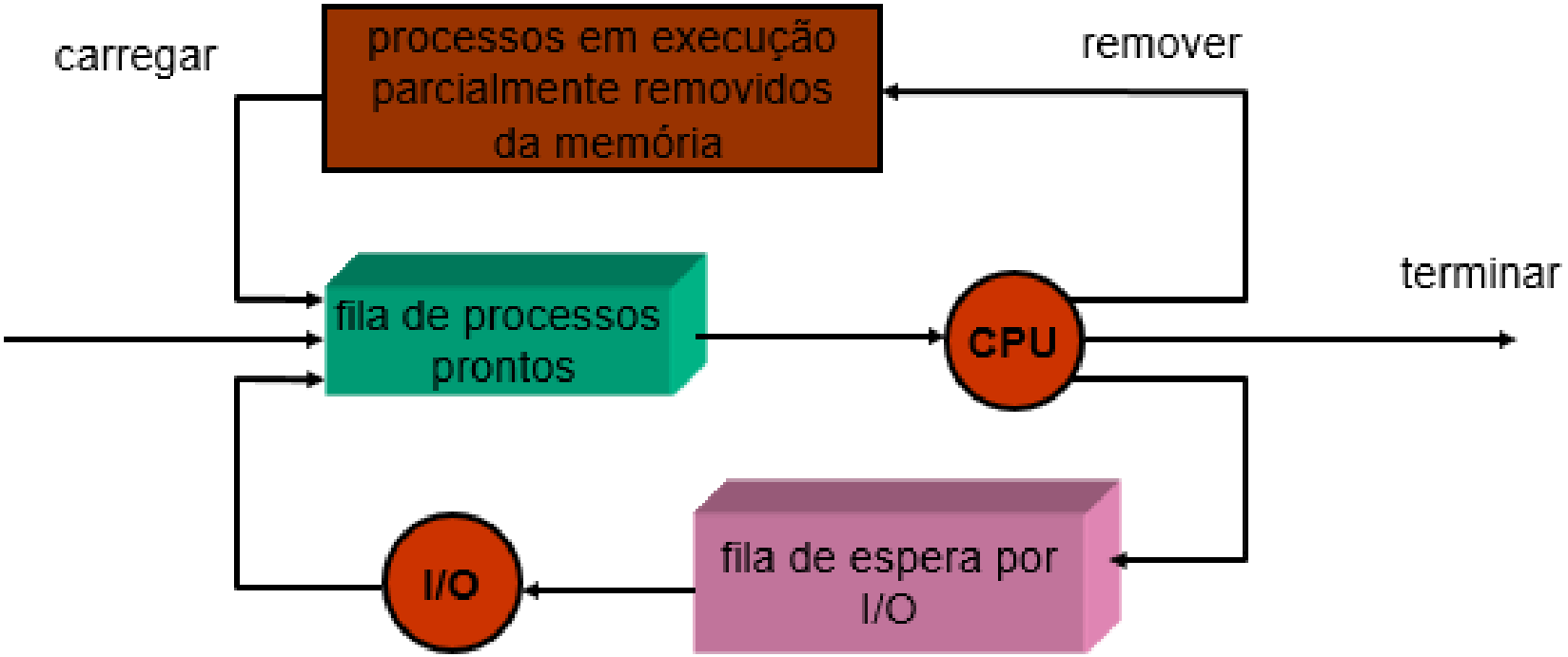
Responsável por passar o controle da CPU para o processo selecionado pelo escalonador de curto prazo, envolve:

- mudança de context;
- mudança para o modo usuário;
- salto para a posição adequada dentro do processo; selecionado para reiniciar sua execução.

Latência de despacho

⇒ Tempo gasto pelo *dispatcher* para interromper um processo e começar a execução de um outro


fila de processos prontos
CPU
I/O
fila de dispositivo
requisição de I/O
término de fatia de tempo
filho em execução
criação de um processo filho
interrupção ocorre
em espera por uma interrupção


carregar
processos em execução
parcialmente removidos
da memória
remover
terminar
fila de processos
prontos
CPU
I/O
fila de espera por
I/O

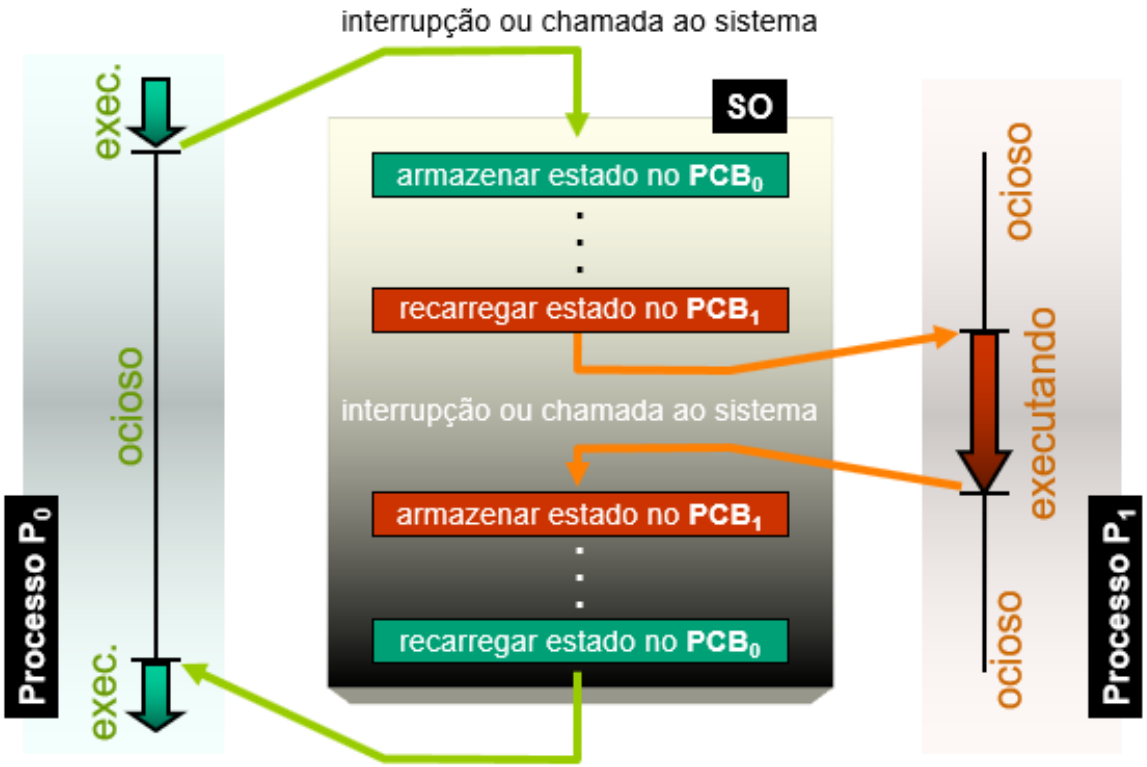
Mudança de contexto

CPU é chaveada para outro processo $\Rightarrow$ SO deve salvar o estado do processo antigo e carregar o estado do novo processo;

Implica *overhead* $\Rightarrow$ SO não realiza nenhum trabalho útil durante os chaveamentos;

Tempo consumido é dependente do suporte de hardware fornecido.

Chaveamento da CPU



Escalonador da CPU é invocado muito freqüentemente (milissegundos):

⇒ precisa ser rápido

Escalonador de processos é invocado com muito pouca freqüência (segundos, minutos):

⇒ pode ser lento

O escalonador de processos controla o grau de multiprogramação do Sistema.

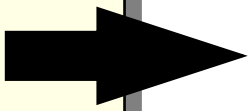
Os escalonadores são implementados por algoritmos dentro do sistema operacional, critérios para comparar a eficiência dos algoritmos:

utilização da CPU (1)
taxa de saída (*throughput*) (2)
turnaround time (3)
tempo de espera (4)
tempo de resposta (5)

Objetivos maximizar (1) e (2)
minimizar (3), (4) e (5)

Tipo de processamento
- *batch*
- interativo
- *CPU bound*
- *I/O bound*

Tipo de sistema
- monoprogramado (?)
- multiprogramado
- *time-sharing*
- tempo-real
- multiprocessado



política de escalonamento (*scheduling policies*)

Critérios de Escalonamento

Orientados ao Usuário e Desempenho:

Uso do processador ⇒ mede a porcentagem de tempo em que a CPU está ocupada;

- importante em tempo compartilhado;
- não muito importante em sistemas monousuário e tempo-real;

Tempo de resposta:

- processos interativos;
- tempo entre uma requisição e o início da resposta do ponto de vista do usuário;
- qual seria o tempo de resposta ideal ?

Orientados ao Usuário e Desempenho:

Deadlines (prazos) ⇒ quando o prazo de término pode ser especificado;

o sistema deveria fazer o melhor esforço para atender todos os prazos;

Previsibilidade ⇒ um dado processo deveria executar sempre em um tempo médio previsível;

a carga do sistema não deveria impor variações.

Orientados ao Sistema e Desempenho:

Throughput (vazão) ⇒ número de processos completados por unidade de tempo, depende:

- do tamanho dos processos;
- das políticas de escalonamento;

Turnaround ⇒ intervalo de tempo entre a submissão de um processo e o seu término:

- inclui o tempo de execução, espera por recursos;
- medida para sistemas *batch;*

Waiting time ⇒ quantidade total de tempo que um processo esteve esperando na fila de prontos.

Orientados ao Sistema:

Justiça ⇒ processos devem ser tratados igualmente, a menos que especificado o contrário;

processos não deveriam sofrer *starvation;*

Prioridades ⇒ processos mais prioritários devem efetivamente ser favorecidos:

problema da inversão de prioridade;

Balanceamento de recursos ⇒ recursos devem ficar ocupados o máximo possível:

processos que não vão utilizar recursos sobrecarregados devem ser favorecidos.

Longa duração ⇒ decisão de se adicionar um processo ao *pool* de processos para serem executados:

admissão ao sistema;

Duração média ⇒ decisão de se adicionar ao número de processos que está completamente ou parcialmente na memória:

swapping, memória virtual.

Curta duração $\Rightarrow$ decisão de qual processo disponível será executado:

interrupção de *clock* e I/O, chamadas ao sistema, *signals;*

I/O $\Rightarrow$ decisão de qual processo que está na fila de espera por uma requisição de I/O será tratado.

Tipos:

não-preemptivo: processo executando não pode ser interrompido;

preemptivo: processo pode ser retirado do processador;

Políticas mais comuns:

First-Come-First-Served (FCFS);

Shortest Job First (SJF);

Prioridade;

Múltiplas Filas;

Round-Robin.

First-Come-First-Served (FIFO)

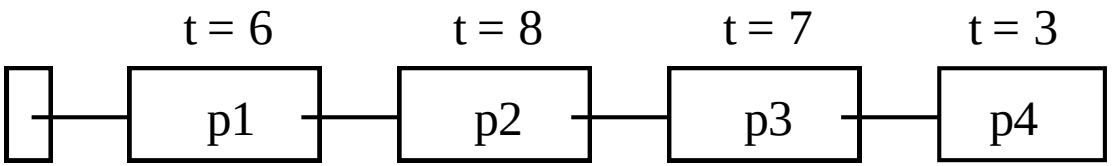
Não preemptivo por definição;

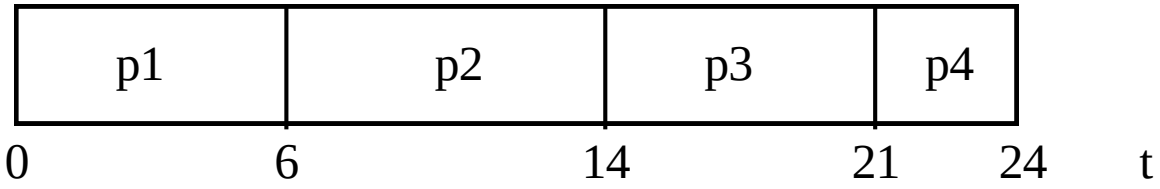
Primeiro processo da fila é o primeiro a ser executado;

Processos usam a CPU até terminar todo processamento;
Mesmo com alguma intercalação, processos com menor prioridade podem prejudicar;
processos com maior prioridade inversão de prioridade *starvation.*

First-Come-First-Served

PROCS.	TE	TT
p1	0 ut	6 ut
p2	6 ut	8 ut
p3	14 ut	7 ut
p4	21 ut	3 ut





Shortest-Job-First

Pode ser preemptiva ou não-preemptiva;

Cada processo é associado ao seu tempo de uso do processador;

Escalonado o processo com o menor tempo de CPU:

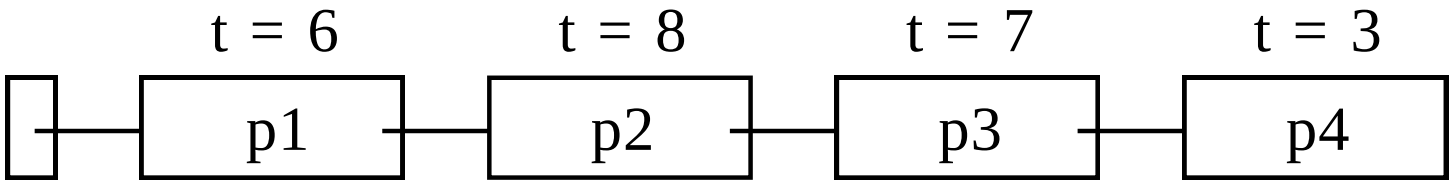
- privilegiam processos menores;
- reduzem o tempo médio de espera na fila de prontos;

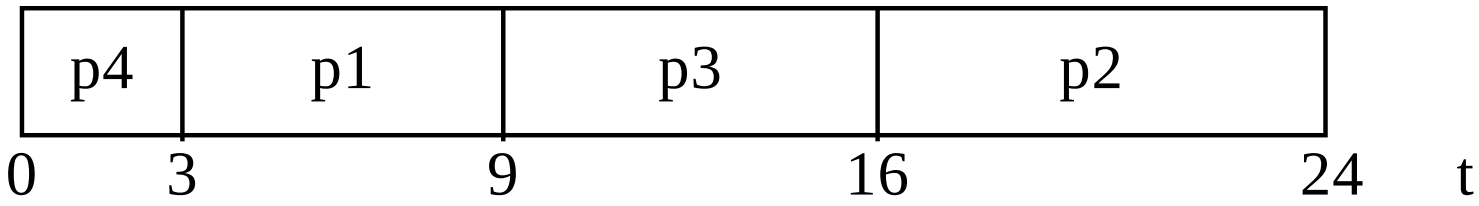
Problema:

- Como determinar quanto tempo de CPU será necessário?

Shortest-Job-First

Tanto o escalonamento FIFO quanto o SJF não são utilizados em sistemas de *time-sharing* (por quê ?)





Shortest-Job-First

A política *SJF* é ótima, minimizando o tempo médio de espera de um conjunto de processos;

Dificuldade: determinar antecipadamente o tempo de processador de cada processo;

Na prática, o tempo é estimado, é utilizada uma aproximação.

Prioridade

A cada processo é atribuída uma prioridade;

O processo com maior prioridade é atribuído ao processador;

Pode ser não-preemptiva ou preemptiva:

não-preemptiva: o processo libera espontaneamente o processador;

preemptiva : o processo executando é interrompido caso chegue à fila de prontos um processo com maior prioridade.

Prioridade

Vantagens:

é possível fazer diferenciação entre processos

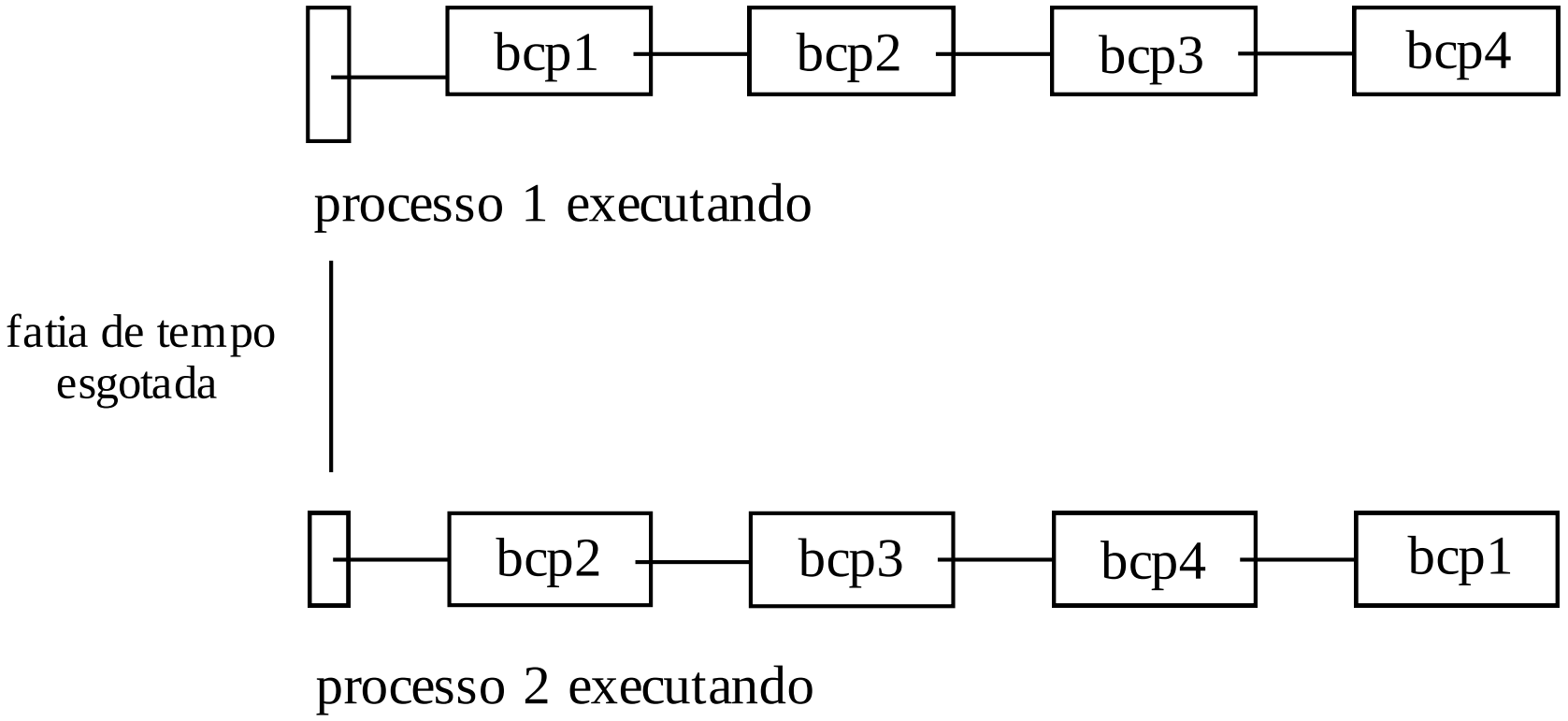
adaptabilidade (prioridades dinâmicas)

Desvantagem:

starvation: um processo com baixa prioridade pode nunca ser atribuído ao processador;

solução: aumentando, em intervalos regulares, a prioridade dos processos que estão há muito tempo esperando.

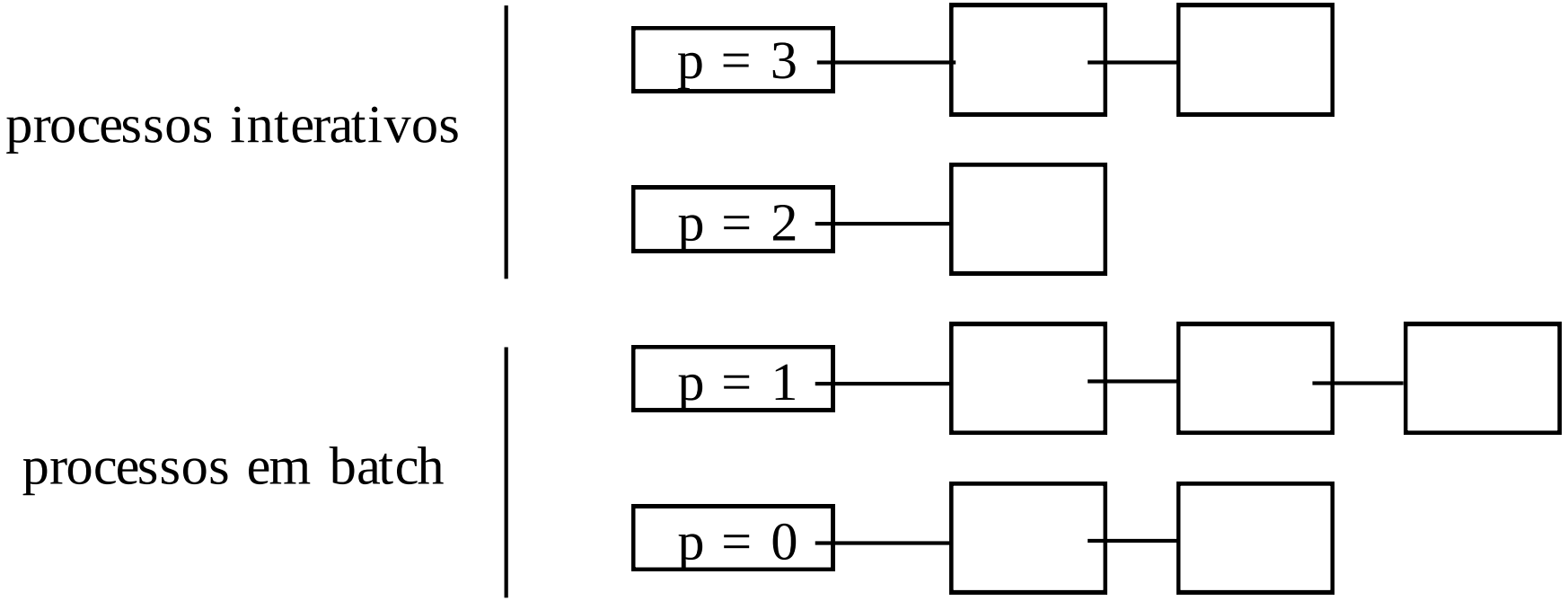
Round-Robin


bcp1
bcp2
bcp3
bcp4
processo 1 executando
fatia de tempo esgotada
bcp2
bcp3
bcp4
bcp1
processo 2 executando

Múltiplas Filas

Política do tipo preemptiva;
Prioridades são atribuídas às classes de processos;
Processos das classes de maior prioridade recebem o processador;
Processos podem migrar entre classes de acordo com seu comportamento;
Vantagem: adaptabilidade de acordo com o comportamento do processo.

Múltiplas Filas



Múltiplas Filas com Realimentação

Escalonamento anterior a classificação dos processos era estática;

Se processo alterar seu comportamento, o esquema pode falhar (não existe reclassificação);

Seria interessante que o SO:

- reconhecesse a alteração de comportamento de um processo;
- ajustasse dinamicamente o seu tipo de escalonamento.

No escalonamento por múltiplas filas com realimentação (*multi-level feed-bak queues*):

é permitido que os processos sejam movimentados entre as filas;
ajuste dinâmico (*mecanismo adaptativo*);
processo é direcionado para uma das filas em função de seu comportamento.

Criação do processo:

⇒ prioridade mais alta e *quantum* mais baixo

Cada fila pode implementar uma política de escalonamento diferente para chegar a CPU:

FIFO *com quantum;*
SJF;
RR.

Processo é reescalonado dentro da mesma fila quando:

processo volta ao estado de pronto;

sofre preempção por outro processo de uma fila mais prioritária;

Processo é direcionado para fila de menor prioridade e maior *quantum* quando:

processo esgota o seu *quantum* (sofrendo preempção);

Quanto maior a prioridade menor o *quantum;*

Escalonamento de uma fila só acontece depois que todas as outras filas de prioridade mais alta estão vazias;

Fila de menor prioridade $\Rightarrow$ *Round-Robin.*

Atende as necessidades de escalonamento de diversos tipos de processos;

Processos *I/O bound:*

bom tempo de resposta: maior prioridade;

permanecem a maior parte do tempo nas filas de alta prioridade;

usa pouco a CPU;

Processos *CPU bound:*

com o transcorrer do processamento sua prioridade vai sendo reduzida;

É um mecanismo complexo e gera *overhead*, mas os resultados são satisfatórios.