Sistemas Operacionais: Escalonamento

Prof. Dr. Rafael Lopes Gomes
Universidade Estadual do Ceará (UECE)

Agenda

- Processos
- Threads
- Escalonamento de Processos

Escalonamento

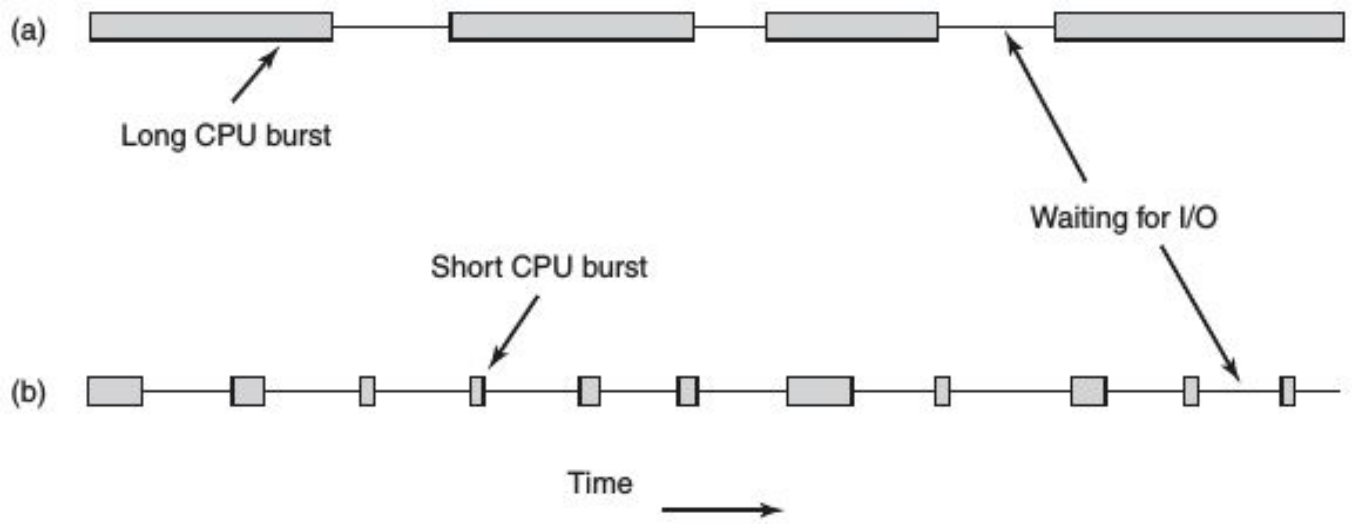
- Multiprogramação: tem múltiplos processos ou threads competindo simultaneamente pela CPU.
  - Sempre que dois ou mais processos estão no estado **pronto**.
- Se apenas uma CPU está disponível, uma escolha precisa ser feita sobre qual processo executar.
  - Quem realiza ? **Escalonador.**
  - Comportamento: **Algoritmo de Escalonamento**
- Em geral, quando o núcleo gerencia threads, o escalonamento é feito por thread, com pouca ou nenhuma noção sobre o processo.

Escalonamento

- Em computadores pessoais, o escalonamento tem baixa importância, pois há pouca concorrência em relação ao volume de recursos disponíveis.
- Escalonamento é crucial:
  - Servidores: diversos processos competem pela CPU.
  - Dispositivos móveis: não há abundância de recursos.
- Usar eficientemente a CPU:
  - Chaveamento de processos é custoso
    - Troca de modo usuário para núcleo.
    - Salvar estado do processo e carregar o estado do novo processo carregado.

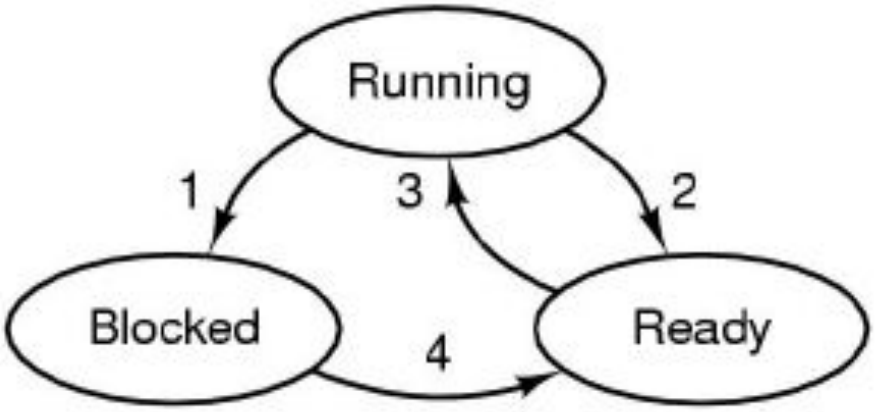
Escalonamento

- Processos limitados:
  - CPU: raras operações de E/S.
  - E/S: operações de E/S constantes.
- CPUs cada vez mais rápidas fazem os processos ficarem mais limitados a E/S.



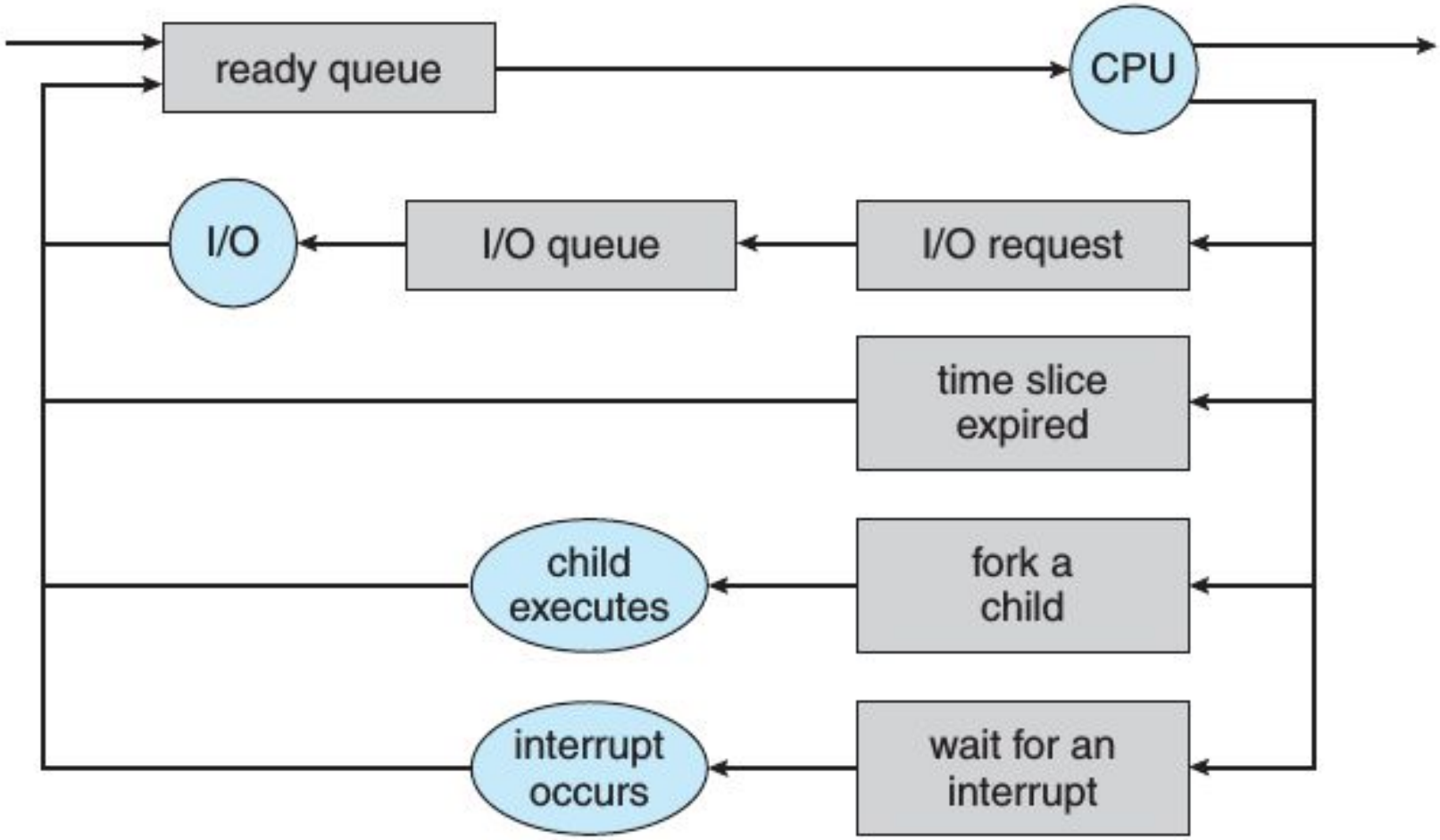
Escalonamento

- Quando escalonar:
  - Quando um processo é criado
  - Quando um processo termina
  - Quando um processo faz uma operação de E/S
  - Interrupção de relógio (sistemas preemptivos)



1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

Escalonamento



Escalonamento

- Categorias de algoritmos de escalonamento:
  - Não preemptivo: escolhe o processo e este executa até ser bloqueado (E/S ou a espera de outro processo) ou libere voluntariamente a CPU.
  - Preemptivo: escolhe o processo e o deixa executar por no máximo um certo tempo fixado.
    - Timeout: processo suspenso e o escalonador escolhe outro processo para executar.
- Ambientes para escalonar:
  - Lote (ou batch): não há usuários esperando por uma resposta rápida.
  - Interativo: normalmente servem múltiplos usuários apressados.
  - Tempo real: em geral, multimídia.

Objetivos do Escalonamento

- Justiça: dar a cada processo uma porção justa da CPU;
- Equilíbrio: manter todas as partes do sistema ocupadas (CPU, E/S, ...);
- **Vazão (ou Throughput)**: número de tarefas ou requisições atendidas por hora ou minutos;
- Utilização de CPU: tempo de ocupação da CPU;
- **Tempo de resposta**: tempo entre o início do processo e o término.
- **Tempo de espera**: tempo de um processo na fila de espera para ser executado.

Algoritmos de Escalonamento

- Em lote:
  - Primeiro a Chegar Primeiro a ser Sevido (FCFS)
  - Tarefa mais curta Primeiro (SJF)
- Em sistemas interativos:
  - Chaveamento circular (Round-Robin)
  - Prioridades
  - Múltiplas Filas ou Multinível
  - Loteria

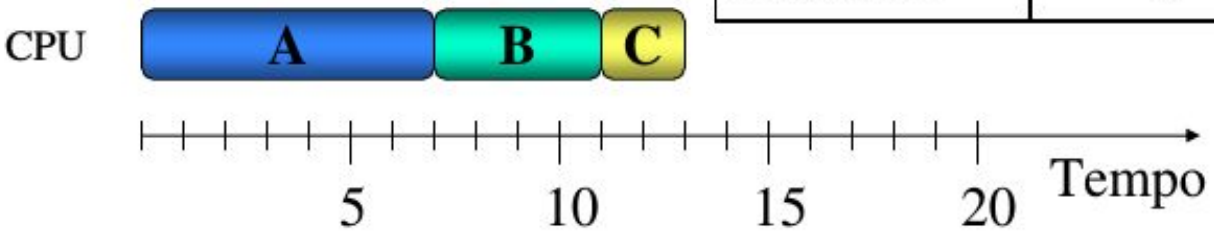
First-Come First-Served (FCFS)

- O processador é alocado seguindo a ordem de chegada dos processos à fila de processos prontos.
- O processo que tem a CPU não a libera até que acabe sua execução ou até que fique bloqueado por uma operação de E/S.
- Vantagem: simples de implementar
- Desvantagens:
  - Dependente da ordem de chegada dos processos
  - Altos tempos de espera
  - Favorece aos processos com muita carga de CPU

First-Come First-Served (FCFS)

- Exemplo: 3

Processo	Tempo de chegada	Etapas do proceso
Processo A	0	7_{CPU}
Proceso B	2	4_{CPU}
Processo C	3	2_{CPU}



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ}\ processos} = \frac{0 + (7-2) + (11-3)}{3} = \frac{0+5+8}{3} = 4{,}3$$

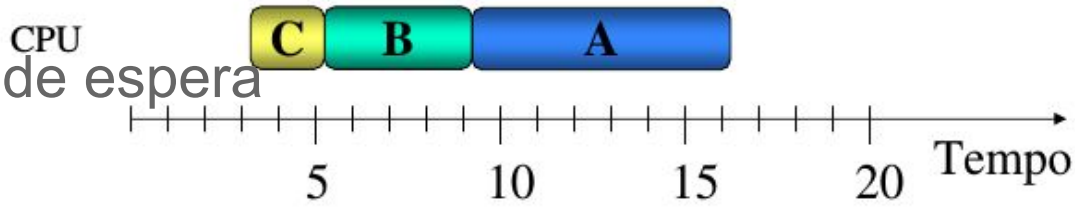
$$T\,Retorno_{medio} = \frac{T\,Retorno_A + T\,Retorno_B + T\,Retorno_C}{n^{\circ}\ processos} = \frac{7+11+13}{3} = 10{,}33$$

Shortest Job First (SJF)

- O processador é alocado ao processo com etapa de CPU mais breve (empate se aplica FIFO).
- Não preemptivo
  - O processo que possui a CPU somente a libera quando quando termina sua execução ou quando se bloqueia
- Com preempção
  - Se um outro processo chegar com uso de CPU menor do que o restante do processo atual, há preempção.
  - Esse esquema é conhecido como “Shortest Remaining Time First” (SRTF).

Shortest Job First (SJF)

- Desvantagens:
  - É preciso saber sobre as tarefas antecipadamente
  - Se muitas tarefas curtas chegarem, as mais longas nunca serão escalonadas.
- Vantagens:
  - Reduz o tempo médio de espera
  - Melhora vazão

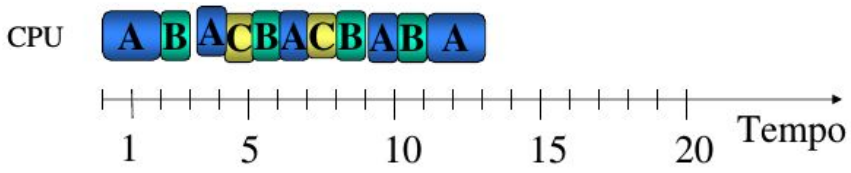


$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ}\ procesos} = \frac{(9-0)+(5-2)+(3-3)}{3} = \frac{9+3+0}{3} = 4$$

$$T\,\mathrm{Re}torno_{medio} = \frac{T\,\mathrm{Re}torno_A + T\,\mathrm{Re}torno_B + T\,\mathrm{Re}torno_C}{n^{\circ}\ procesos} = \frac{16+9+5}{3} = 10$$

Round-Robin

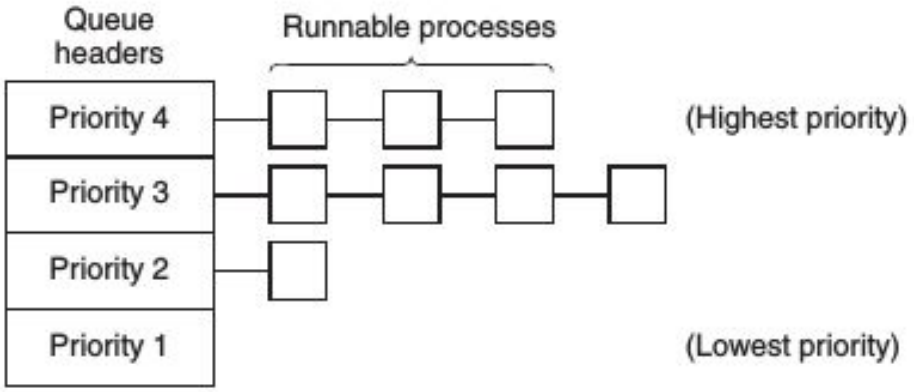
- Atribui-se a cada processo durante um intervalo de tempo um valor pré fixado de forma rotativa, denominado **quantum** (q).
- Semelhante ao FCFS, mas com comportamento de fila circular.
- Nenhum processo espera mais do que (n-1)*q para utilizar a CPU: Não ocorre starvation.
- Valor do quantum: desempenho
  - Curto: muitas trocas de contexto.
  - Longo: tende a se tornar o FCFS.



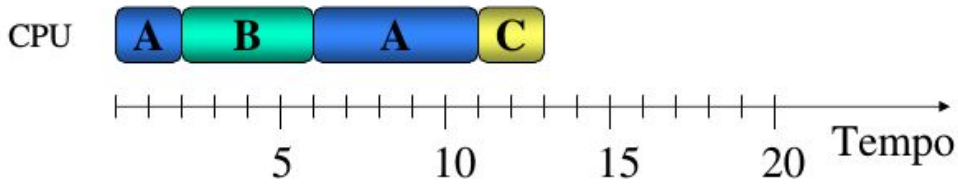
- Exemplo: Ca $TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^o\ processos} = \frac{6 + (7-2) + (6-3)}{3} = \frac{6+5+3}{3} = 4{,}6$

$$T\,\mathrm{Re}torno_{medio} = \frac{T\,\mathrm{Re}torno_A + T\,\mathrm{Re}torno_B + T\,\mathrm{Re}torno_C}{n^o\ processos} = \frac{13+11+8}{3} = 10{,}66$$

Prioridades



- Cada processo tem associado um valor inteiro que representa sua prioridade de execução.
- O escalonador escolhe o processo da fila de processos prontos que tenha a maior prioridade.
- Pode ser preemptivo ou não.
- Prioridade pode ser estática ou dinâmica.
- Exemplo: Prioridades A
  - 1 é mais alta.
  - 9 a mais baixa.



$$TEspera_{medio} = \frac{TEspera_A + TEspera_B + TEspera_C}{n^{\circ}\ processos} = \frac{6+0+(11-3)}{3} = \frac{6+0+8}{3} = 4{,}66$$

$$T\,\mathrm{Re}torno_{medio} = \frac{T\,\mathrm{Re}torno_A + T\,\mathrm{Re}torno_B + T\,\mathrm{Re}torno_C}{n^{\circ}\ processos} = \frac{11+6+13}{3} = 10$$

Multinível

- Fila de prontos é dividida em várias filas
- Cada fila possui seu próprio algoritmo de escalonamento
- Escalonamento entre filas:
  - Escolher qual fila será executada
  - Opções:
    - Prioridade: causa starvation
    - Dividir por tempo de execução

Multinível com Retroalimentação

- Sem retroalimentação: nunca é trocado de fila;
- Com retroalimentação: pode ser trocado de fila;
- Características:
  - número de filas;
  - algoritmos de escalonamento para cada fila;
  - método usado para determinar quando elevar um processo;
  - método usado para determinar quando rebaixar um processo;
  - método usado para determinar em que fila um
  - processo entrará quando esse processo precisar de atendimento;

Loteria

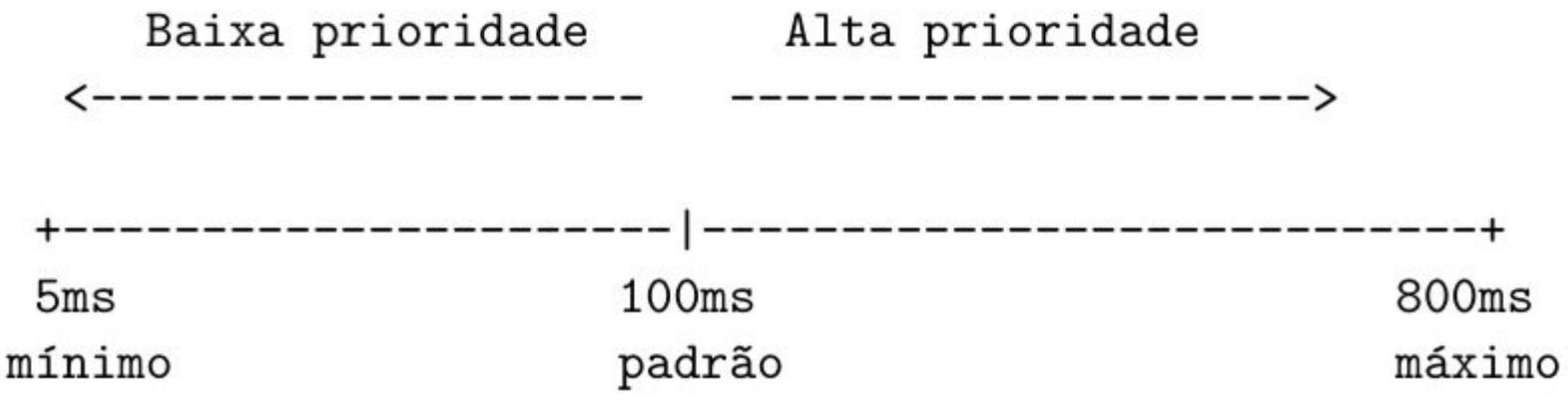
- Dar bilhetes aos processos e sortear um bilhete para executar.
- Processos mais importantes podem receber mais bilhetes.
- Pode ser estimado a partir de modelos probabilísticos

Avaliação dos Algoritmos

- Por simulação:
  - Informações podem ser geradas aleatoriamente (processos, chegadas, términos, etc).
- Por implementação
  - Mais realistas
  - Alto custo: mudar o kernel.

Linux

- Prioridade
  - Estática (nice): -20 a 19
    - Menor valor, Maior prioridade
    - ps –el
  - Dinâmica: 0 a 99
    - Maior valor, Maior prioridade
    - Favorece threads interativas



FIM