Sistemas Operacionais

Escalonamento

Profª Drª Thaína Aparecida Azevedo Tosta

tosta.thaina@unifesp.br

UNIFESP
Universidade Federal de São Paulo

São José dos Campos
2024/01

Aula passada

- O problema do jantar dos filósofos
- O problema dos leitores e escritores

Sumário

- Introdução ao escalonamento de processos
- Escalonamento em sistemas em lote
- Escalonamento em sistemas interativos
- Escalonamento em sistemas de tempo real
- Política *vs* mecanismo de escalonamento
- Escalonamento de threads

Objetivo: relacionar estratégias de escalonamento a diferentes tipos de sistemas.

Introdução ao escalonamento de processos

- Quando um computador é multiprogramado, ele frequentemente tem múltiplos processos ou threads prontos competindo pela CPU ao mesmo tempo;
- A parte do sistema operacional que faz a escolha de qual executar é chamada de escalonador, por um algoritmo de escalonamento;
- Muitas das mesmas questões que se aplicam ao escalonamento de processos também se aplicam ao escalonamento de threads.

Introdução ao escalonamento de processos

- Computadores de grande porte: combinam serviço em lote e de compartilhamento de tempo (usuário interativo em um terminal);
- Computadores pessoais: computadores tornaram-se tão rápidos que a CPU dificilmente ainda é escassa;
- Servidores em rede: múltiplos processos muitas vezes competem pela CPU;
- Smartphones: otimização do consumo de energia.

Introdução ao escalonamento de processos

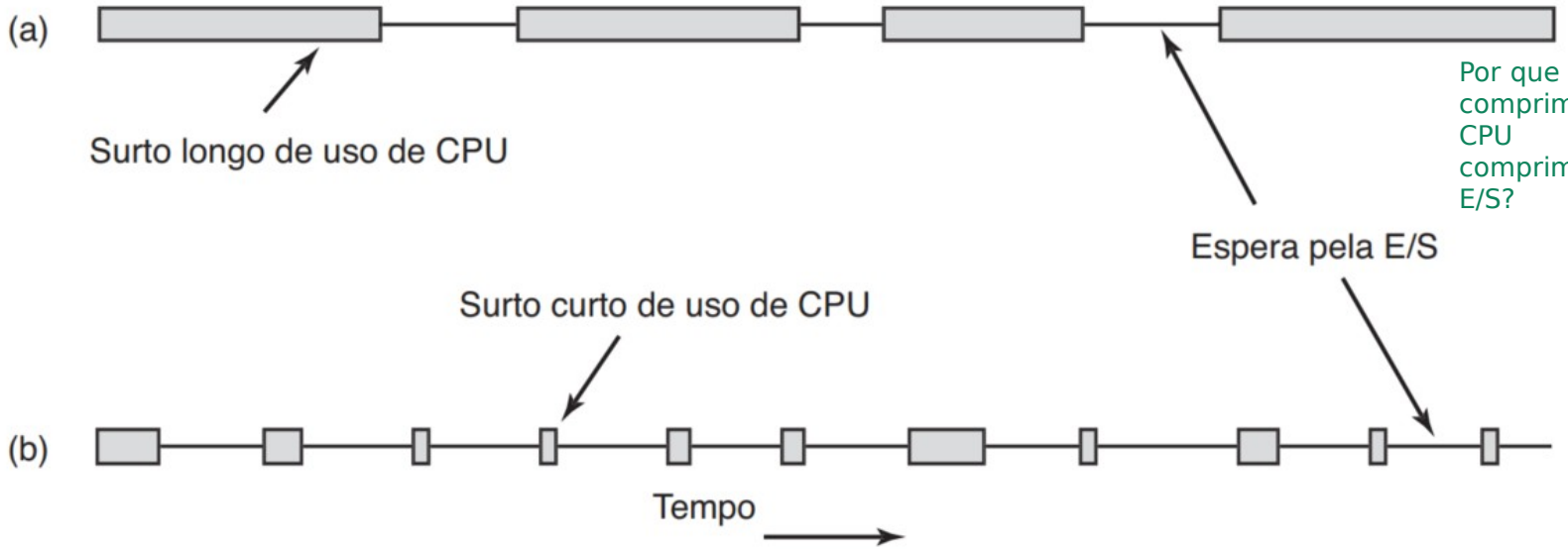
O que acontece no escalonamento de processos?

1. Troca do modo usuário para o modo núcleo;
2. Salva estado do processo atual (incluindo armazenar registradores na tabela de processos);
3. Em alguns sistemas, salva mapa de memória;
4. Executa algoritmo de escalonamento;
5. MMU é recarregada com o mapa de memória do novo processo;
6. Inicializa o novo processo.

E a cache? e as tabelas relacionadas?

Introdução ao escalonamento de processos

Surtos de uso da CPU alternam-se com períodos de espera por E/S. (a) Um processo limitado pela CPU. (b) Um processo limitado pela E/S.



Por que o fator chave é o comprimento do surto da CPU e não o comprimento do surto da E/S?

Introdução ao escalonamento de processos

- À medida que as CPUs ficam mais rápidas, os processos tendem a ficar mais limitados pela E/S (problema para o futuro);

- Se um processo limitado pela E/S quiser executar, ele deve receber uma chance rapidamente para que possa emitir sua solicitação de disco e manter o disco ocupado.

Introdução ao escalonamento de processos

O escalonamento é necessário quando:

1. Um novo processo é criado: executar pai ou filho?;
2. Um processo termina: escolher do conjunto de processos prontos (se nenhum? Processo ocioso);
3. Um processo bloqueia para E/S: a razão para bloquear pode ter um papel na escolha (se A é importante e espera por B? Deixa B executar);
4. Ocorre uma interrupção de E/S: executar o que ficou pronto há pouco, o que estava sendo executado no momento da interrupção, ou algum terceiro processo?;
5. Ocorre uma interrupção do relógio: a cada *k* interrupções periódicas do hardware de relógio, pode decidir executar ou não um outro processo pronto.

Introdução ao escalonamento de processos

Uma decisão de escalonamento pode ser feita a cada interrupção ou a cada *k*-ésima interrupção de relógio por:

- Algoritmo de escalonamento **não preemptivo**: escolhe um processo e o executa até que ele seja bloqueado (seja em E/S ou esperando por outro processo), ou libera voluntariamente a CPU;
- Algoritmo de escalonamento **preemptivo**: escolhe um processo e o deixa executar por no máximo um certo tempo fixo → suspensão do processo e escolha de novo pelo escalonador.

Introdução ao escalonamento de processos

Diferentes áreas de aplicação (e de tipos de sistemas operacionais) têm diferentes algoritmos de escalonamento:

- Lote: folhas de pagamento, estoques, contas a receber, contas a pagar, cálculos de juros (em bancos), processamento de pedidos de indenização (em companhias de seguro) e outras tarefas periódicas;
- Interativo: usuários interativos ou múltiplos usuários (servidores);
- Tempo real: processos sabem que não podem executar por longos períodos e em geral realizam o seu trabalho e bloqueiam rapidamente.

Preemptivo ou não preemptivo?

Introdução ao escalonamento de processos

Algumas metas do algoritmo de escalonamento

Todos os sistemas

- Justiça — dar a cada processo uma porção justa da CPU
- Aplicação da política — verificar se a política estabelecida é cumprida
- Equilíbrio — manter ocupadas todas as partes do sistema

Sistemas em lote

- Vazão (*throughput*) — maximizar o número de tarefas por hora
- Tempo de retorno — minimizar o tempo entre a submissão e o término
- Utilização de CPU — manter a CPU ocupada o tempo todo

Sistemas interativos

- Tempo de resposta — responder rapidamente às requisições
- Proporcionalidade — satisfazer às expectativas dos usuários

Sistemas de tempo real

- Cumprimento dos prazos — evitar a perda de dados
- Previsibilidade — evitar a degradação da qualidade em sistemas multimídia

Escalonamento em sistemas em lote

- Múltiplos processos (multiprogramação), mas não há usuários (impacientes) aguardando respostas (rápidas);
- Geralmente, não preemptivo.
  - Aplicações específicas (antigamente científicas);
  - Aceitável longos intervalos de execução de processos.

Objetivos comuns:

- Justiça: dar a cada processo uma porção justa da CPU;
- Aplicação da política: verificar se a política é cumprida;
- Equilíbrio: manter ocupada todas as partes do sistema.

Objetivos específicos:

- Vazão: maximizar a taxa de jobs por hora;
- Tempo de retorno: minimizar latência entre submissão e retorno de job;
- Utilização da CPU: manter a CPU ocupada o tempo todo.

Escalonamento em sistemas em lote

Primeiro a chegar, primeiro a ser servido (*first-come*, *first-served* - FCFS)

- Não preemptivo;
- Basicamente, há uma fila única de processos prontos;
- Fácil de compreender e fácil de implementar;
- Controle de todos os processos por uma única lista encadeada;
- Desvantagem: um processo limitado pela computação com posterior leitura de bloco *vs* muitos processos limitados pela E/S.

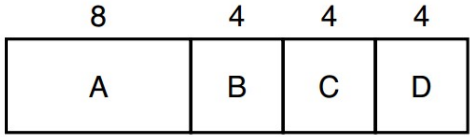
Escalonamento em sistemas em lote

Tarefa mais curta primeiro (*shortest job first* - SJF)

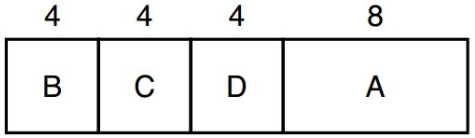
- Não preemptivo;
- Presume que os tempos de execução são conhecidos antecipadamente;
- Possível quando um trabalho similar é realizado todos os dias.

(a)	(b)
A = 8 min	A = 4 min
B = 12 min	B = 8 min
C = 16 min	C = 12 min
D = 20 min	D = 20 min

Um exemplo do escalonamento tarefa mais curta primeiro. (a) Executando quatro tarefas na ordem original. (b) Executando-as na ordem tarefa mais curta primeiro.



(a)



(b)

Escalonamento em sistemas em lote

Tempo restante mais curto em seguida (*shortest remaining time next* - SRTN)

- Preemptivo;
- O tempo de execução precisa ser conhecido antecipadamente;
- Quando uma nova tarefa chega, seu tempo total é comparado com o tempo restante do processo atual;
- Permite que tarefas curtas novas tenham um bom desempenho.

Escalonamento em sistemas interativos

- Múltiplos usuários, múltiplos processos (multiprogramação) e poucos recursos;
- Preempção é fundamental.
  - Evita que um processo apodere da CPU e deixe outros em inanição.

Objetivos comuns:

- Justiça: dar a cada processo uma porção justa da CPU;
- Aplicação da política: verificar se a política é cumprida;
- Equilíbrio: manter ocupada todas as partes do sistema.

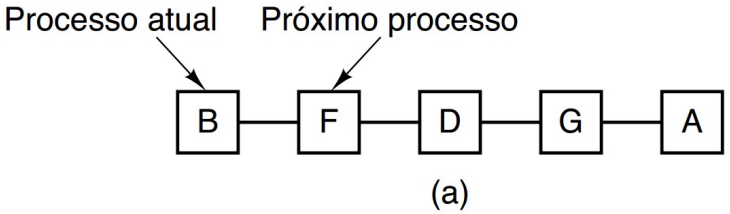
Objetivos específicos:

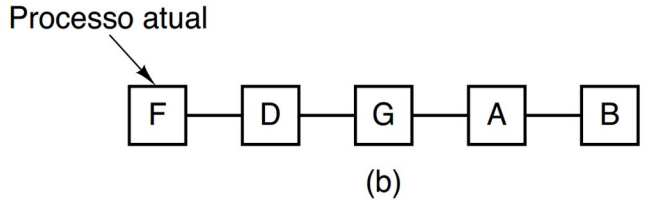
- Tempo de resposta (ações/requisições dos usuários);
- Proporcionalidade (satisfazer as expectativas dos usuários).

Escalonamento em sistemas interativos

Escalonamento por chaveamento circular (*round-robin* - RR)

- Preemptivo;
- Um dos mais antigos, simples, justos e amplamente usados;
- Fácil de implementar com manutenção de uma lista de processos executáveis;
- A cada processo é designado um quantum, durante o qual ele é deixado executar.





Escalonamento em sistemas interativos

Escalonamento por chaveamento circular (*round-robin* - RR)

- Chaveamento de processo ou chaveamento de contexto (salvar e carregar registradores e mapas de memória, atualizar várias tabelas e listas, carregar e descarregar memória cache, e assim por diante) exige tempo.

Chaveamento de 1ms	
Quantum curto	Quantum longo
4ms 20% do tempo da CPU é desperdiçado Muitos chaveamentos e menor eficiência da CPU	100ms 1% do tempo da CPU é desperdiçado Com 50 solicitações de tempos de CPU diversos, o último esperará 5s (ruim se ele levar poucos ms)

- Quantum > surto de CPU médio → ↓preempção → ↑operações de bloqueio antes do quantum → ↑chaveamento de processo
- ~20-50ms é razoável

Escalonamento em sistemas interativos

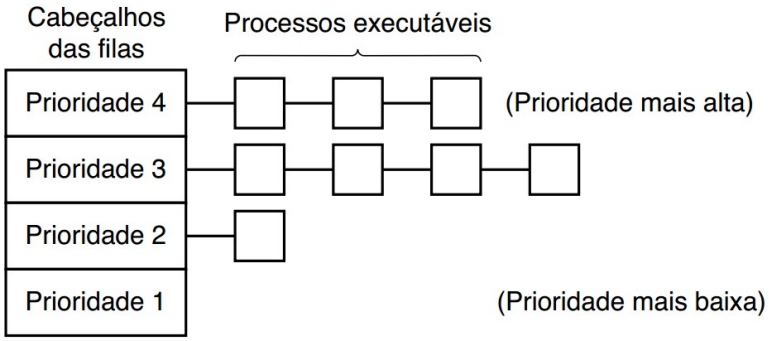
Escalonamento por prioridades

- A necessidade de levar em consideração fatores externos leva ao escalonamento por prioridades: a cada processo é designada uma prioridade, e o processo executável com a prioridade mais alta é autorizado a executar;
- Diminuir a prioridade do processo que está sendo executado em cada tique do relógio pode evitar que processos de prioridade mais alta executem indefinidamente;
- E um quantum máximo?

Escalonamento em sistemas interativos

Escalonamento por prioridades

- Prioridades podem ser designadas a processos estaticamente ou dinamicamente:
  - Estáticas: maior prioridade a hierarquias mais altas;
  - Dinâmicas: maior prioridade a processos altamente limitados pela E/S.
- Ainda é possível agrupar em classes de prioridade e usar o escalonamento de prioridades entre as classes, mas escalonamento circular dentro de cada classe.



Escalonamento em sistemas interativos

Múltiplas filas

- Uma possibilidade é estabelecer classes de prioridade;
- Exemplo: Processo que precisa computar por 100 quanta:

Classe 1	1 quantum
Classe 2	2 quanta
Classe 3	4 quanta
Classe 4	8 quanta
Classe 5	16 quanta
Classe 6	32 quanta
Classe 7	64 quanta (mas usa só 37)

Escalonamento em sistemas interativos

Processo mais curto em seguida (*shortest process next* - SPN)

- Podemos minimizar o tempo de resposta geral executando a tarefa mais curta primeiro;
- Como descobrir qual dos processos atualmente executáveis é o mais curto? Estimativas baseadas no comportamento passado (envelhecimento ou *aging*).

T_0

$T_0/2 + T_1/2$

$T_0/4 + T_1/4 + T_2/2$

$T_0/8 + T_1/8 + T_2/4 + T_3/2$

Escalonamento em sistemas interativos

Escalonamento por loteria

- A ideia básica é dar bilhetes de loteria aos processos para vários recursos do sistema, como o tempo da CPU;
- Sempre que uma decisão de escalonamento tiver de ser feita, um bilhete de loteria será escolhido ao acaso, e o processo com o bilhete fica com o recurso;
  - ↑ importante o processo ↑bilhetes ↑mais chances de acesso à CPU.
- Processos cooperativos podem trocar bilhetes se assim quiserem: processos cliente e servidor.

Escalonamento em sistemas interativos

Escalonamento por fração justa

- Até agora presumimos que cada processo é escalonado por si próprio, sem levar em consideração quem é o seu dono;

✘ Usuário 1 com 9 processos (90% de CPU) e usuário 2 com 1 processo (10% de CPU);

✓ Se dois usuários têm cada um 50% da CPU prometidos, cada um receberá isso, não importa quantos processos eles tenham em existência.

Escalonamento em sistemas de tempo real

- O tempo é essencial: CD player, monitoramento de pacientes em uma UTI, piloto automático em um avião, controle de robôs em uma fábrica automatizada, etc;
  - Sistemas de tempo real crítico: há prazos absolutos que devem ser cumpridos;
  - Sistemas de tempo real não crítico: descumprir um prazo é indesejável, mas mesmo assim tolerável.
- Processos geralmente têm vida curta e podem ser concluídos em bem menos de um segundo;
- Em caso de evento externo, o escalonador programa os processos para atender todos os prazos.

Escalonamento em sistemas de tempo real

- Os eventos a que um sistema de tempo real talvez tenha de responder podem ser categorizados como:
  - Periódicos: ocorrem em intervalos regulares;
  - Aperiódicos: ocorrem de maneira imprevisível.
- Um sistema de tempo real escalonável atende a:

P_i (período)	C_i (segundos de CPU)
100 ms	50 ms
200 ms	30 ms
500 ms	100 ms

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

Política *versus* mecanismo

- Além de podermos ter processo de diferentes usuários competindo pela CPU, podemos ter um processo com muitos filhos executando sob seu controle;
- Exemplo:
  - Pai: Um processo de sistema de gerenciamento de banco de dados
  - Filho 1: análise sintática de consultas;
  - Filho 2: acesso ao disco;
  - Filho 3: lidando com solicitação diferente;
  - Filho 4: outra função específica para realizar.
- Nenhum dos escalonadores discutidos aceita qualquer entrada dos processos do usuário sobre decisões de escalonamento, e agora?

Política *versus* mecanismo

- Mecanismo de escalonamento é **DIFERENTE** de política de escalonamento;
- O algoritmo de escalonamento é parametrizado de alguma maneira, mas os parâmetros podem estar preenchidos pelos processos dos usuários;
- O núcleo oferece uma chamada de sistema pela qual um processo pode estabelecer (e mudar) as prioridades dos filhos: mecanismo no núcleo e política no processo do usuário.

Escalonamento de threads

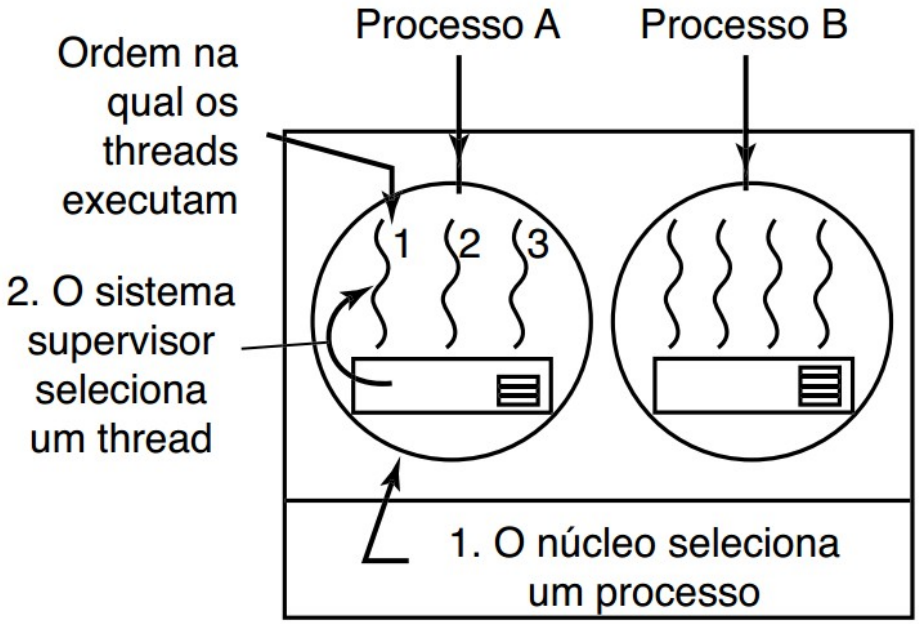
- Quando vários processos têm cada um múltiplos threads, temos dois níveis de paralelismo presentes: processos e threads;

- Escalonar nesses sistemas difere substancialmente, dependendo se os threads de usuário ou os threads de núcleo (ou ambos) recebem suporte.

Escalonamento de threads

Threads de usuário

- O escalonador de thread dentro de A decide qual thread executar por quanto tempo quiser;
- Se as threads têm pouco trabalho para fazer por surto de CPU? Cada um executa por um tempo, então cede a CPU de volta para o escalonador de threads.



Possível: A1, A2, A3, A1, A2, A3
Impossível: A1, B1, A2, B2, A3, B3

Escalonamento de threads

Threads de usuário

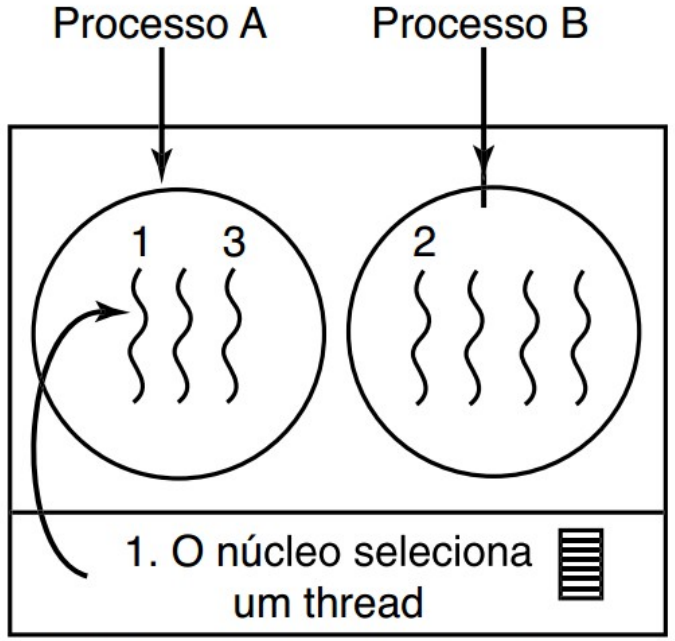
- Qual algoritmo usar? O escalonamento circular e o de prioridade são os mais comuns;

- A única restrição é a ausência de um relógio para interromper um thread que esteja sendo executado há tempo demais.
  - Isso é um problema?

Escalonamento de threads

Threads de núcleo

- O núcleo escolhe um thread em particular para executar e pode até considerar a qual processo ele pertence;
- Com um quantum de 50 ms, mas threads que são bloqueados após 5 ms, temos algo que não é possível com esses parâmetros e threads de usuário.



Possível: A1, A2, A3, A1, A2, A3
Também possível: A1, B1, A2, B2, A3, B3

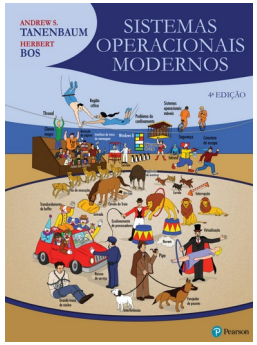
Escalonamento de threads

Características	Thread de usuário	Thread de núcleo
Chaveamento de thread	Punhado de instruções de máquina	Chaveamento de contexto completo, mudar o mapa de memória e invalidar o cache (ainda mais demorado)
Bloqueio de thread na E/S	Suspende todo o processo	Não suspende todo o processo
Escalonador de thread específico de uma aplicação	Possível (O sistema de tempo de execução sabe o que todos os threads fazem)	Impossível (o núcleo jamais saberia o que cada thread faz)

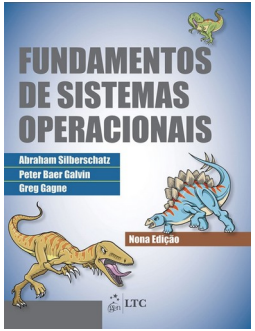
Relacionar estratégias de escalonamento a diferentes tipos de sistemas

a) Escalonamento por chaveamento circular
b) Escalonamento por fração justa
c) Escalonamento por loteria
d) Escalonamento por prioridades
e) Múltiplas filas
f) Primeiro a chegar, primeiro a ser servido
g) Processo mais curto em seguida
h) Tarefa mais curta primeiro
i) Tempo restante mais curto em seguida

Referências



TANENBAUM, Andrew S.; BOS, Herbert. Sistemas operacionais modernos. 4. edição. São Paulo: Pearson, 2016. xviii, 758 p. ISBN 9788543005676.



SILBERSCHATZ, Abraham.; GALVIN, Peter Baer.; GAGNE, Greg. Fundamentos de sistemas operacionais. 9. edição. Rio de Janeiro: LTC, 2015.

O modelo desta apresentação foi criado pelo Slidesgo.
Agradeço ao Prof. Bruno Kimura da Universidade Federal de São Paulo pelo material disponibilizado.