

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

MIZAEL AGUIAR SILVA

GABRIEL LEAL COSTA PARK

BRENO MENDES BERNARDO

LUCAS ZIMERER COSTA

DAVI VITOR GOMES ORNELAS

**ESCALONAMENTO DE PROCESSOS**

**TEOFILO OTONI – MG**

**19/11/2023**

MIZAEL AGUIAR SILVA

GABRIEL LEAL COSTA PARK

BRENO MENDES BERNARDO

LUCAS ZIMERER COSTA

DAVI VITOR GOMES ORNELAS

**Escalonamento de Processos**

Trabalho apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Federal do Norte de Minas Gerais Campus Teófilo Otoni, para fins de obtenção de nota na disciplina de Sistemas Operacionais, ministrada pelo professor Audrey Teles dos Santos

TEOFILO OTONI – MG

06/11/2023

**RESUMO**

**O escalonamento de processos é uma peça essencial na gestão eficiente de recursos em sistemas computacionais. Este trabalho oferece uma visão abrangente dessa prática, abordando desde definições fundamentais até a análise de algoritmos. São explorados objetivos como equidade, produtividade e previsibilidade, considerando as complexidades inerentes ao design de algoritmos de escalonamento. Além disso, são examinados os níveis de escalonamento, desde a admissão de processos até a alocação efetiva do processador. Diversos algoritmos são apresentados, destacando suas características e aplicações específicas, proporcionando uma compreensão aprofundada do papel crítico do escalonamento na otimização do desempenho de sistemas computacionais.**

**Palavras-chave:** escalonamento de processos, otimização do desempenho, sistemas computacionais.

**ABSTRACT**

**Process scheduling is a pivotal component in the efficient management of resources in computational systems. This work provides a comprehensive overview of this practice, addressing fundamental definitions and algorithm analysis. Objectives such as fairness, productivity, and predictability are explored, considering the inherent complexities in designing scheduling algorithms. Additionally, scheduling levels, from process admission to effective processor allocation, are examined. Various algorithms are presented, highlighting their characteristics and specific applications, offering an in-depth understanding of the critical role scheduling plays in optimizing the performance of computational systems.**

**Keywords:** Process scheduling, optimizing the performance, computational systems.

**SUMARIO**

[**1 INTRODUÇÃO** 5](#_Toc151302248)

[**2.1.2 Funções Principais de um Sistema Operacional** 6](#_Toc151302249)

[**2.1.3 Papel do Escalonamento de Processos no Sistema Operacional** 7](#_Toc151302250)

[**2.2 Definição de Escalonamento de Processos** 7](#_Toc151302251)

[**2.2.1 O que é Escalonamento de Processos** 7](#_Toc151302252)

[**2.2.2 Importância do Escalonamento** 7](#_Toc151302253)

[**2.2.3 Escalonamento como Política de Tratamento de Processos** 7](#_Toc151302254)

[**2.3 Objetivos do Escalonamento** 8](#_Toc151302255)

[**2.4 Níveis de escalonamento** 8](#_Toc151302256)

[**2.4.1 Escalonamento de Alto Nível** 9](#_Toc151302257)

[**2.4.2 Escalonamento de Nível Intermediário** 9](#_Toc151302258)

[**2.4.3 Escalonamento de Baixo Nível** 9](#_Toc151302259)

[**2.5 Escalonamento Preemptivo e Não Preemptivo** 9](#_Toc151302260)

[**2.5.1 Algoritmos Preemptivos** 10](#_Toc151302261)

[**2.5.2 Algoritmos Não Preemptivos** 10](#_Toc151302262)

[**2.5.3 Cooperação e Preempção Voluntária** 10](#_Toc151302263)

[**2.6 Algoritmos de Escalonamento** 10](#_Toc151302264)

[**2.6.1 Escalonamento FIFO (First In First Out)** 10](#_Toc151302265)

[**2.6.2 Escalonamento HPF (Highest Priority First)** 10](#_Toc151302266)

[**2.6.3 Escalonamento SJF (Shortest Job First)** 10](#_Toc151302267)

[**2.6.4 Escalonamento HRN (Highest Response-Ratio Next)** 11](#_Toc151302268)

[**2.6.5 Escalonamento SRT (Shortest Remaining Time)** 11](#_Toc151302269)

[**3 CONCLUSÃO** 11](#_Toc151302270)

[**4 REFERENCIAS** 12](#_Toc151302271)

# 1 INTRODUÇÃO

No cenário complexo dos sistemas computacionais, o escalonamento de processos emerge como um componente crucial para otimizar a utilização dos recursos do processador e, consequentemente, a eficiência do sistema operacional. Este trabalho explora os fundamentos e os desafios associados ao escalonamento de processos, fornecendo uma análise aprofundada dos objetivos, níveis e algoritmos envolvidos nesse processo vital.

O escalonamento de processadores, definido como a alocação de processos aos recursos de processamento, é uma tarefa complexa executada pelo sistema operacional. O escalonador, encarregado dessa decisão, desempenha um papel crucial na determinação da ordem em que os processos serão executados, influenciando diretamente a produtividade e eficiência globais de um sistema computacional.

Os objetivos fundamentais do escalonamento são variados, buscando equilibrar a justiça no tratamento dos processos, maximizar a produtividade, garantir previsibilidade, minimizar tempos de resposta para usuários interativos e otimizar a utilização de recursos, entre outros. No entanto, esses objetivos muitas vezes entram em conflito, exigindo uma cuidadosa consideração no design de algoritmos de escalonamento.

O trabalho também explora os diferentes níveis de escalonamento, desde o alto nível, que lida com a admissão de processos, até o baixo nível, responsável pela determinação do próximo processo a utilizar efetivamente o processador. Essa hierarquia reflete a complexidade inerente à gestão de processos em um ambiente computacional.

Além disso, são apresentados diversos algoritmos de escalonamento, cada um com suas características e aplicações específicas. Desde o clássico First In First Out (FIFO) até abordagens mais avançadas como o Multilevel Feedback Queues (MFQ), o trabalho explora como esses algoritmos lidam com desafios como tempo de execução imprevisível, uso intensivo de E/S e a necessidade de equilibrar o tratamento de processos curtos e longos.

Dessa forma, esta análise proporciona uma compreensão abrangente dos princípios fundamentais do escalonamento de processos, destacando sua importância na otimização do desempenho dos sistemas computacionais. Ao explorar os objetivos, níveis e algoritmos, este trabalho contribui para a compreensão crítica e aprimoramento contínuo das estratégias de escalonamento em ambientes computacionais cada vez mais complexos.

**2 DESENVOLVIMENTO**

**2.1 Fundamentos de Sistemas Operacionais**

**2.1.1 Definição de Sistema Operacional**

O sistema operacional é como o cérebro do computador. Ele atua como um intermediário crucial, proporcionando uma camada de abstração entre os recursos físicos do computador e os programas em execução. Essencialmente, o sistema operacional coordena e alocava recursos como CPU, memória, dispositivos de entrada/saída e rede. No contexto do escalonamento de processos, o sistema operacional desempenha um papel fundamental ao determinar quando e como os processos são executados, contribuindo diretamente para a eficiência e o equilíbrio do sistema.

# **2.1.2 Funções Principais de um Sistema Operacional**

As funções essenciais de um sistema operacional abrangem uma variedade de tarefas. A alocação de memória, por exemplo, garante que os programas tenham o espaço necessário para serem executados, enquanto o controle de dispositivos gerencia a comunicação eficiente entre software e hardware. O fornecimento de uma interface de usuário, seja por meio de linha de comando ou interfaces gráficas, é uma função crucial para facilitar a interação do usuário com o sistema. Neste contexto, o escalonamento de processos destaca-se como uma função crítica, influenciando diretamente a capacidade do sistema operacional de atender às demandas concorrentes de processamento.

# **2.1.3 Papel do Escalonamento de Processos no Sistema Operacional**

O escalonamento de processos desempenha um papel central no sistema operacional, decidindo a ordem de execução dos processos na CPU. Ao otimizar a utilização dos recursos do sistema, o escalonamento busca minimizar o tempo de espera dos processos, maximizar o throughput e proporcionar uma experiência de usuário mais responsiva. Age como um árbitro imparcial, distribuindo justamente os recursos escassos entre os processos concorrentes. Assim, o escalonamento se torna um componente vital para garantir a eficiência operacional do sistema operacional, tornando-se um aspecto inextricável de seu funcionamento.

# **2.2 Definição de Escalonamento de Processos**

# **2.2.1 O que é Escalonamento de Processos**

Vamos simplificar: em um sistema com apenas um processador, o escalonamento é como decidir a ordem em que os programas serão executados. É como organizar uma fila para que cada processo tenha sua vez. Tanenbaum e Deitel propõe a seguintes definições:

I. Quando mais de um processo é executável, o sistema operacional deve decidir qual será executado primeiro. A parte do sistema operacional dedicada a esta decisão é chamada escalonador (scheduler) e o algoritmo utilizado é chamado algoritmo de escalonamento (scheduling algorithm). [TAN92, p. 62]

II. A designação de processadores físicos para processos permite aos processos a realização de trabalho. Esta designação é uma tarefa complexa realizada pelo sistema operacional. Isto é chamado escalonamento do processador (scheduling). **[DEI92,p.287]**

# **2.2.2 Importância do Escalonamento**

Como o escalonamento é feito influencia bastante o quão bem o sistema funciona. É como o diretor de uma peça teatral, determinando como os atores entram e saem do palco. O escalonamento é responsável por garantir que o sistema seja produtivo e eficiente.

# **2.2.3 Escalonamento como Política de Tratamento de Processos**

Não é apenas uma regra mecânica. O escalonamento é como uma política, uma estratégia que guia como os processos são tratados para obter os melhores resultados no sistema. É um jeito de garantir que tudo funcione da melhor maneira possível.

# **2.3 Objetivos do Escalonamento**

Quando pensamos em criar um escalonador eficiente, é importante considerar várias necessidades diferentes. Em termos simples, a elaboração da política de escalonamento deve focar em alguns objetivos essenciais:

1. Justiça: Certificar-se de que todos os processos têm uma oportunidade justa de usar o processador. É como garantir que todos na fila tenham sua vez.

2. Eficiência: Fazer com que as coisas aconteçam da maneira mais rápida e eficiente possível. É como planejar o caminho mais curto para alcançar um destino.

3. Minimizar a Espera: Evitar que os processos fiquem esperando por muito tempo para serem executados. É como tentar reduzir o tempo que passamos na fila do supermercado.

4. Maximizar a Utilização da CPU: Garantir que o processador esteja sempre fazendo algo útil. É como manter seu computador ocupado, realizando tarefas importantes.

5. Responder às Necessidades Específicas: Adaptar o escalonador para atender às demandas específicas do sistema. É como ajustar a configuração do seu carro para diferentes tipos de viagem.

6. Evitar Conflitos: Certificar-se de que os processos não estão brigando por recursos. É como garantir que todos em uma sala compartilhem os brinquedos sem confusão.

Portanto, criar um bom escalonador não é só uma questão técnica, mas também sobre garantir que as coisas aconteçam de maneira justa, rápida e eficiente para todos os processos no sistema.

O escalonamento de processos é uma prática essencial em sistemas operacionais, com objetivos específicos que moldam seu desempenho.

# **2.4 Níveis de escalonamento**

O escalonamento de processos em um sistema computacional é uma prática complexa, e seus diferentes níveis desempenham papéis específicos na gestão eficiente dos recursos. Analisaremos os três níveis distintos de escalonamento, considerando a frequência e complexidade das operações envolvidas.

# **2.4.1 Escalonamento de Alto Nível**

Também conhecido como escalonamento de tarefas, este nível lida com a admissão de processos. Ele decide quais tarefas serão admitidas para competir pelos recursos do sistema. Uma vez admitidas, essas tarefas se transformam em processos. O escalonamento de alto nível corresponde a rotinas de alto nível fornecidas pelas APIs do sistema operacional, facilitando a interação inicial com o sistema.

# **2.4.2 Escalonamento de Nível Intermediário**

Neste nível, determina-se quais processos existentes competirão pelo uso da CPU (processos ativos). O escalonamento intermediário gerencia a carga do sistema, utilizando primitivas de suspensão (suspend) e ativação (resume ou activate). Essas operações visam otimizar a utilização dos recursos, e as rotinas correspondentes são internas ao sistema operacional, garantindo uma gestão eficaz dos processos em execução.

# **2.4.3 Escalonamento de Baixo Nível**

As rotinas de escalonamento de baixo nível são responsáveis por determinar qual processo ativo será o próximo a utilizar efetivamente a CPU. Essas operações são executadas pelo dispatcher, uma rotina geralmente escrita diretamente em linguagem de máquina e permanentemente armazenada na memória principal. O escalonamento de baixo nível, ou de curto prazo, ocorre constantemente e representa a troca de contexto e o chaveamento do processador entre os processos ativos.

Esses níveis de escalonamento, também conhecidos como escalonamento de longo prazo, médio prazo e curto prazo, desempenha\*\*Capítulo 3: Escalonamento de Processos - Relatório sobre Escalonamento Preemptivo e Não Preemptivo\*\*

O terceiro capítulo aborda a distinção crucial entre algoritmos de escalonamento preemptivo e não preemptivo, delineando suas características e aplicabilidades em diversos contextos operacionais.

# **2.5 Escalonamento Preemptivo e Não Preemptivo**

O escalonamento preemptivo e não preemptivo difere na maneira como um processo é tratado pelo processador. Em algoritmos não preemptivos, o processador é dedicado a um processo até que este seja concluído. Em contrapartida, os algoritmos preemptivos permitem a retirada do processador de um processo em execução para atender a outro.

# **2.5.1 Algoritmos Preemptivos**

Algoritmos preemptivos destacam-se em ambientes com múltiplos processos que requerem atenção simultânea, como sistemas multiusuário interativos e sistemas de tempo real. A capacidade de interromper um processo em execução é essencial para lidar com a imprevisibilidade desses ambientes, embora essa capacidade venha acompanhada da complexidade na implementação. A troca de contexto necessária para interromper e retomar um processo implica sobrecarga computacional, tornando os algoritmos preemptivos mais desafiadores de serem desenvolvidos e gerenciados.

# **2.5.2 Algoritmos Não Preemptivos**

Os algoritmos não preemptivos são mais diretos e adequados para processamento não interativo, assemelhando-se aos esquemas de processamento em lote de sistemas batch. Embora ofereçam menos interatividade, esses algoritmos são geralmente mais eficientes e previsíveis em termos de tempo de conclusão de tarefas.

# **2.5.3 Cooperação e Preempção Voluntária**

Além dos modelos puramente preemptivos e não preemptivos, existem abordagens cooperativas, onde a preempção ocorre apenas em situações específicas, como operações de I/O ou a conclusão de um processo. Alguns sistemas permitem preempção voluntária, onde um processo educado pode ceder o processador voluntariamente para beneficiar outros processos.

# **2.6 Algoritmos de Escalonamento**

# **2.6.1 Escalonamento FIFO (First In First Out)**

- Conhecido como FCFS (First Come First Served).

- Os processos são organizados em uma fila com base na ordem de chegada.

- Não preemptivo, pois os processos em execução não são interrompidos.

# **2.6.2 Escalonamento HPF (Highest Priority First)**

- Os processos são organizados por prioridade.

- Variante do escalonamento FIFO.

- Não preemptivo, priorizando processos mais importantes.

# **2.6.3 Escalonamento SJF (Shortest Job First)**

- Prioriza processos com menor tempo de execução.

- Uma forma de HPF, onde o tempo de serviço é a prioridade.

- Não preemptivo, favorecendo processos mais curtos.

# **2.6.4 Escalonamento HRN (Highest Response-Ratio Next)**

- Balanceia a duração do job e seu tempo de espera.

- Oferece dinamismo na organização da fila com base na taxa de resposta.

- Não preemptivo, proporcionando um equilíbrio positivo na seleção de processos.

# **2.6.5 Escalonamento SRT (Shortest Remaining Time)**

- Versão preemptivo do SJF. (Shortest Job First):

- Organiza a fila com base no tempo estimado de execução.

- Oferece preempção de processos em execução conforme novos processos chegam.

# **3 CONCLUSÃO**

Em resumo, a análise abrangente dos princípios, metas, níveis e algoritmos associados ao escalonamento de processos destaca sua importância crucial nos sistemas operacionais e na eficiência global dos ambientes computacionais. O sistema operacional, funcionando como o cérebro do computador, depende do escalonamento para otimizar a distribuição de recursos, assegurando equidade, eficiência e resposta rápida aos usuários.

Os objetivos do escalonamento, como equidade, eficiência, minimização do tempo de espera e maximização da utilização da CPU, representam desafios complexos enfrentados pelos algoritmos de escalonamento. A análise dos diferentes níveis de escalonamento, desde o alto até o baixo nível, ilustra a complexidade inerente à gestão de processos em ambientes computacionais.

Além disso, a diferenciação entre escalonamento preemptivo e não preemptivo destaca a importância da flexibilidade na abordagem do tratamento de processos, considerando a diversidade das demandas do sistema. Os algoritmos mencionados, como FIFO, HPF, SJF, HRN e SRT, oferecem estratégias diversas para lidar com desafios específicos, desde processos interativos até tarefas em lote.

Em última análise, a prática do escalonamento de processos é uma peça fundamental na melhoria contínua do desempenho dos sistemas computacionais. A compreensão crítica desses conceitos e a escolha apropriada de algoritmos contribuem para a eficiência operacional, a resposta eficaz às demandas dos usuários e a otimização dos recursos do sistema. À medida que os ambientes computacionais evoluem, a evolução e adaptação contínuas dos métodos de escalonamento são cruciais para enfrentar os desafios em constante mudança e proporcionar experiências computacionais mais eficientes e ágeis.

# **4 REFERENCIAS**

MAZIERO, Carlos. Sistemas Operacionais: Conceitos e Mecanismos. 2019. Disponível em: <https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-16.pdf>. Acesso em: 11 nov. 2023.

JANDL, Peter, Jr. Notas sobre Sistemas Operacionais. 2004. Disponível em: <https://docente.ifrn.edu.br/rodrigotertulino/livros/notas-sobre-sistemas-operacionais>. Acesso em: 15 nov. 2023.

[TAN92] TANENBAUM, Andrew S. Modern Operating Systems. Prentice Hall, Upper Saddle River, NJ, 1992.

DEI92] DEITEL, Harvey M. An Introduction to Operating Systems, 2nd Edition. Addison-Wesley, Reading, MA, 1992.