

IMPLEMENTAÇÃO DE UM SIMULADOR DE ESTRATÉGIAS DE ESCALONAMENTO DE DISCO RÍGIDO COM CABEÇOTE MÓVEL EM SISTEMAS DE MULTIPROGRAMAÇÃO

¹AGUIAR, Carlos Eduardo Pereira de; ¹ASSIS, Marcos Antônio de; ¹BARBOSA, Aline; ¹DOURADO, Luiz Antonio; ¹FERREIRA, Eliane; ¹GONÇALVES, Olicio Hipólito; ¹MARTINES, Rozimeire dos Santos;

²TAMAE, Rodrigo Yoshio, ²MUZZI, Fernando Augusto Garcia, ²ROSA, Adriano Justino
1-Discentes do Curso Sistemas de Informação – FAEG / Garça
2-Docentes do Curso Sistemas de Informação – FAEG / Garça
rytamae@yahoo.com.br

RESUMO

Uma aplicação desenvolvida em Linguagem Java apresenta as estratégias de escalonamento de disco rígido com cabeçote móvel de forma básica dentro do contexto de otimização e desempenho de disco mediante duas estratégias de escalonamento de disco. Uma vez que o usuário não pode visualizar o processo de busca de dados efetuada pelo sistema operacional na unidade de disco rígido (o que dificulta o seu entendimento), a fim de proporcionar uma melhor compreensão e ressaltar sua importância no contexto de otimização e desempenho de disco, foi desenvolvido um software cujo objetivo é simular o comportamento dos componentes do sistema de armazenamento em disco de cabeçote móvel em sistemas de multiprogramação ao iniciante.

Palavras-chave: estratégias de escalonamento de disco, otimização e desempenho.

ABSTRACT

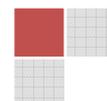
An application developed in Java language will be shown the strategies of hard disk scheduling along mobile head as of basic form within the context of optimization and performance under two scheduling strategies. The user can't preview the process on screen by the operating system on the hard disk unit (the one to arduousness its perception), for the purpose of provide a better understanding and distinct at the context hard disk's optimization and performance, was developed a software whose purpose is simulate the behavior of the system components into multiprogrammer's systems to the initiating.

Keywords: disk scheduling strategy, improvement and performance.

1. INTRODUÇÃO

Nos primeiros computadores os dados eram gravados em cartões perfurados onde a presença ou ausência de furos representava bits de dados. Desenvolver software e fazê-lo funcionar em um computador exigia muito trabalho e o processo era muito lento. Diante da necessidade de dispositivos de armazenamento barato e regraváveis os pesquisadores desenvolveram o armazenamento magnético.

Em 1951, surgiu o armazenamento em fita magnética, que é um tipo de armazenamento de acesso seqüencial, como as fitas de vídeo ou áudio. Bastante eficiente para aplicações nas quais os dados podem ser acessados seqüencialmente. No entanto, em casos que o sistema precisa ser capaz de localizar e atualizar qualquer registro em uma fração de segundo ou em sistemas de multiprogramação esse tipo de armazenamento não é apropriado devido sua lentidão.



Visando resolver esse problema a IBM lançou em 1951 a primeira unidade de disco rígido comercial o RAMAC (*Random Access Method of Accounting and Control* – Método de Contabilidade e Controle de Acesso Aleatório). Unidades de discos rígidos são dispositivos de acesso aleatório, também conhecido como acesso direto, porque não estão limitados a acessar dados seqüencialmente.

Embora discos rígidos resultem em melhor desempenho do que as fitas magnéticas, o alto custo limitava sua utilização. À medida que o tempo foi passando, a capacidade e desempenho do disco rígido aumentavam, enquanto os custos declinavam, as velocidades dos discos rígidos melhoraram lentamente do que suas capacidades. Conforme a velocidade dos processadores e da memória principal aumentarem mais rapidamente do que as dos dispositivos de armazenamento secundário, realizar desempenho ótimo tornou-se importante.

2. CARACTERÍSTICAS DO ARMAZENAMENTO EM DISCO DE CABEÇOTE MÓVEL

A maioria dos computadores modernos usa discos rígidos como armazenamento secundário. A figura 1 mostra uma vista simplificada de um disco de cabeçote móvel.

Os dados são gravados em discos magnéticos, ou pratos (*platters*) que são conectados por uma haste (*spindle*) os discos através dessa haste giram em alta velocidade, geralmente milhares de rotações por segundo, os dados dos discos são acessados por um cabeçote de leitura-escrita que paira sobre a superfície do disco a uma distância muito menor do que uma partícula de fumaça. Mas antes que o dados possam ser acessados a porção do disco no qual eles devem ser lidos deve girar até ficar imediatamente acima ou abaixo do cabeçote de leitura-escrita. O tempo que o dados leva pra girar da posição em que se encontra até o cabeçote de leitura-escrita é denominado tempo de latência rotacional, que é simplesmente a metade do tempo que o disco leva pra completar uma rotação. A maioria dos discos rígidos exibe um tempo de latência rotacional média na ordem de vários milissegundos (DEITELL, 2005).

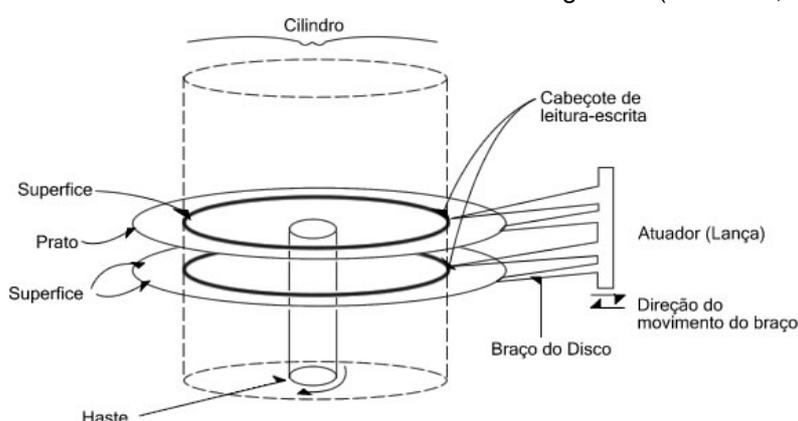
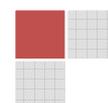


Figura 1 – vista lateral do esquema de um disco de cabeçote móvel.



Enquanto os pratos giram, cada cabeçote de leitura-escrita desenha uma trilha circular (ou cilindro) de dados sobre a superfície de um disco. Cada cabeçote de leitura-escrita é posicionado na extremidade de um braço de disco, ligado a um atuador, também conhecido como lança ou unidade de braço móvel. Esse braço move-se paralelamente à superfície do disco. Quando o braço do disco movimenta os cabeçotes de leitura-escrita para uma nova posição, um conjunto vertical diferente de trilhas circulares torna-se acessível. O processo de levar o braço do disco até um novo cilindro é denominado operação de busca. Para localizar pequenas unidades de dados, o disco divide as trilhas em diversos setores, quase sempre de 512 bits, como se pode observar na Figura 2.

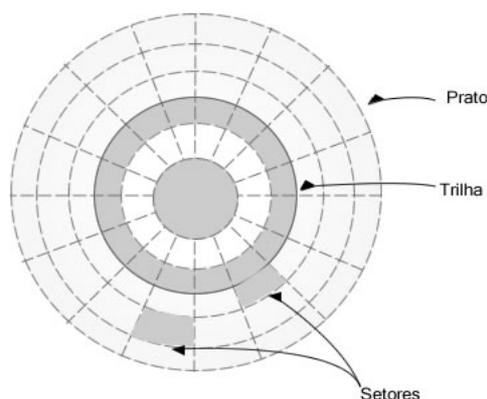
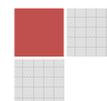


Figura 2 – vista esquemática do alto da superfície de um disco.

O armazenamento em disco de cabeçote móvel exibe velocidades de acesso variável que depende das posições relativas entre o cabeçote de leitura-escrita e o dados requisitado.

O sistema operacional pode localizar determinado dado especificando o cabeçote que indica qual superfície do disco deve ser lida, o cilindro que indica qual trilha a ser lida e o setor no qual o dados está localizado. Para localizar determinado dado, primeiramente o braço do disco deve ir até o cilindro apropriado executando a operação de busca, o tempo que o cabeçote leva pra ir do cilindro em que está até o cilindro que contém o registro de dados é denominado tempo de busca, a porção do disco onde está armazenado o dado deve girar até fiar acima ou abaixo do cabeçote de leitura-escrita, em seguida, o registro, cujo tamanho depende do tipo de dado, deve sofrer um giro por meio do cabeçote de leitura-escrita, o que é denominado tempo de transmissão, o que é visível na Figura 3. Para cada uma dessas operações envolve movimento mecânico, portanto o tempo total de acesso ao disco é freqüentemente equivalente a uma fração apreciável de um segundo, ou seja, diversos milissegundos, no mínimo (DEITELL, 2005).



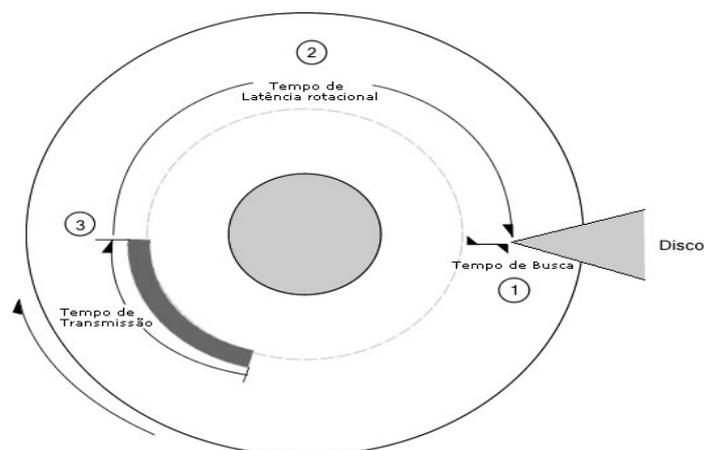


Figura 3 – componentes de acesso a um disco.

2.1 JUSTIFICANDO O ESCALONAMENTO DO DISCO

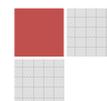
Muitos processos podem gerar requisições para ler e escrever dados em um disco simultaneamente. Pelo fato desses processos às vezes fazerem requisições mais rapidamente do que podem ser atendidas pelo disco, formam-se linhas ou filas para reter as requisições de disco.

Alguns sistemas de computadores mais antigos simplesmente atendiam essas requisições segundo o algoritmo primeira a chegar, primeira a ser atendida (*First-Come-First-Served* - FCFS), mas quando havia uma longa fila de requisições aguardando sua vez pra ser atendida, isso levava a longos tempos de espera, pois esse tipo de algoritmo exhibe um padrão de busca aleatório no qual requisições sucessivas podem causar buscas demoradas partindo dos cilindros mais internos para os mais externos. Para reduzir o tempo gasto na busca de registros, utiliza-se do processo denominado escalonamento de disco, que é uma forma de ordenar a fila de requisições de alguma maneira que não seja pelo algoritmo FCFS, melhorando significativamente o rendimento.

Escalonamento de disco envolve um exame cuidadoso de requisições pendentes para determinar o modo mais eficiente de atendê-las. Um escalonador de disco examina as relações entre as requisições que estão à espera e, então, reorganiza a fila de modo que as requisições sejam atendidas com o mínimo de movimento mecânico.

As estratégias de escalonamento de um disco dependem dos objetivos do sistema, mas a maioria das estratégias é avaliada pelos seguintes critérios:

- Rendimento – o número de requisições atendidas por unidade de tempo.
- Tempo médio de resposta – o tempo médio gasto a espera de que uma requisição seja tendida.



- Variância dos tempos de resposta – uma medida da previsibilidade dos tempos de resposta. Cada requisição de disco deve ser atendida dentro de um período de tempo aceitável, ou seja, a estratégia deve impedir que haja adiantamento indefinido.

A política de escalonamento deve sempre tentar maximizar o rendimento e minimizar o tempo médio de resposta. Muitas políticas de escalonamento tentam atingir esses objetivos minimizando o tempo gasto na execução de buscas demoradas. Quando o rendimento e o tempo de resposta são otimizados, o desempenho médio do sistema melhora, mas requisições individuais podem sofrer atrasos.

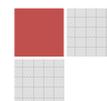
A variância mede como as requisições individuais são atendidas em relação o desempenho médio do sistema. Quanto menor a variância, maior as chances de as requisições serem atendidas após esperar um período de tempo semelhante. Portanto a variância deve ser vista como uma medida de justiça e de previsibilidade.

3. ESCALONAMENTO DE DISCO FCFS

O escalonamento FCFS usa uma fila, de modo que as requisições são atendidas na ordem que chegam, pode ser considerado como uma política justa, pois as requisições que já estão na fila não serão afetadas pelas que chegarem depois, isso garante que nenhuma requisição pode ser adiada indefinidamente, mas também significa que o FCFS poderia realizar uma longa operação de busca para atender à requisição seguinte à espera, mesmo que uma outra requisição da fila estivesse mais próxima e pudesse ser atendida mais rapidamente (figura 4). Embora essa técnica ocorra em baixa sobrecarga de tempo de execução, pode resultar em baixo rendimento devido às longas operações de busca. Exemplo supondo que disco possua 200 cilindros numerados de 0 a 199 e que o cabeçote de leitura-escrita está localizado inicialmente no cilindro 53, as requisições estão em uma fila e são representadas pelo número dos cilindros a serem percorridos pelo cabeçote de leitura-escrita, na ordem de chegada temos os cilindros 98, 183, 37, 122, 14, 124, 65 e 67. O cabeçote de leitura-escrita estará se deslocando através do movimento do braço do disco do cilindro 53 para o cilindro 98 depois para o cilindro 183 e assim sucessivamente até chegar ao cilindro 67, nesse processo de mudança de um cilindro pra outro podemos verificar uma certa demora, pois os braços exigem movimentação mecânica e ao sair do cilindro 183 e ir para o cilindro 37, uma fração apreciável de tempo é perdida durante essa movimentação.

4. O SIMULADOR

O simulador foi desenvolvido em Linguagem Java, por se tratar de uma linguagem de alto nível e também multiplataforma, foi utilizado em conjunto a ferramenta NetBeans IDE 5.5 por oferecer recursos para criação de um ambiente visual e também a fim de melhorar a aparência da aplicação (figura 6). Durante o processo de inicialização da aplicação, ou seja, quando



iniciar a geração dos arquivos a aplicação estará trazendo mensagens na tela das operações efetuadas.

A aplicação estará simulando um disco rígido composto por três pratos, conseqüentemente haverá seis superfícies, onde os cabeçotes de leitura-escrita estarão percorrendo ao efetuar a busca do dado especificado pelo usuário, possui um vetor (lista), onde serão gerados os nomes dos arquivos, essa lista foi definida para gerar arquivos de numeração seqüencial e a quantidade de arquivos poderá ser definida pelo usuário, mas possui um limite mínimo de cem arquivos. Logo que os arquivos são gerados é alimentado também um outro vetor – disco virtual com informações referentes a cada arquivo, essas informações contêm, por exemplo, o endereçamento dos arquivos, a superfície do disco onde foram gravados, o cilindro e o setor, e serão apresentadas posteriormente quando a busca dos arquivos for executada.

Para iniciar o processo de pesquisa é necessário informar a quantidade de arquivos a serem gerados e em seguida informar o nome do arquivo que se deseja pesquisar. Ao iniciar a busca a aplicação executa a estratégia de escalonamento e acesso a disco por meio do algoritmo implementado. Durante todo o processo de pesquisa, são exibidas mensagens na tela informando qual operação está sendo executada. Como exemplo, o tempo gasto para percorrer as trilhas, tempo total da busca, setor localizado. Quando o arquivo é encontrado as informações referentes a ele serão exibidas na tela. Exemplo: nome do arquivo, endereço, superfície do disco onde o arquivo está gravado, cilindro e setor. Caso o usuário digite um nome de arquivo que não esteja na lista de arquivos gerados, os escalonadores farão a busca pelos cilindros e não encontrando o arquivo o tempo total de busca é mostrado juntamente com a mensagem “Arquivo não encontrado”, de acordo com a Figura 10.

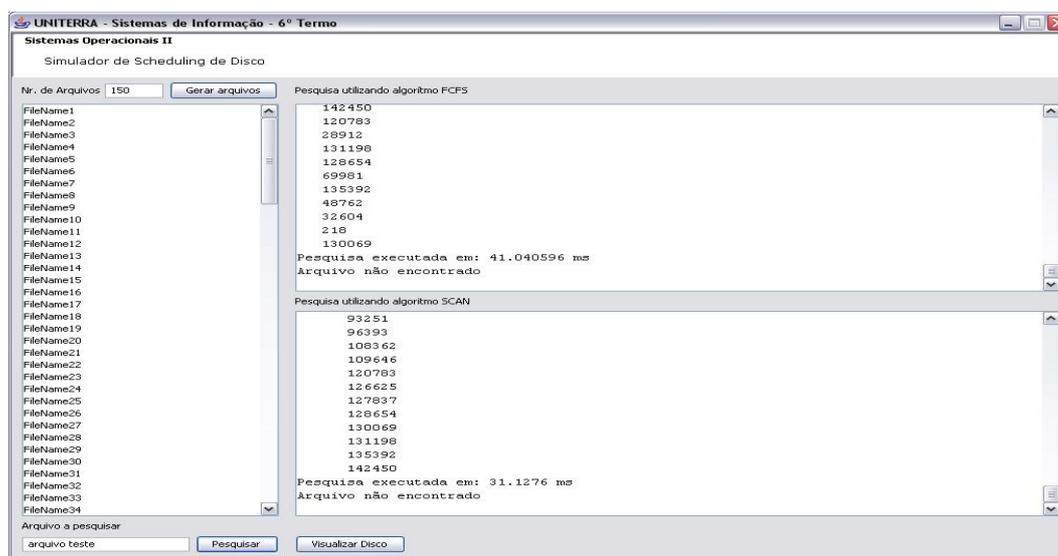
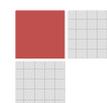


Figura 10 – pesquisa executada, arquivo não encontrado.



Caso o usuário queria realizar outra pesquisa, a aplicação deve ser reiniciada, pois há uma pequena limitação no código da aplicação que não oferece nova busca com os arquivos gerados anteriormente.

5. CONCLUSÃO

Existem vários tipos de estratégias de escalonamento de disco, que possibilitam uma melhor forma de se obter um desempenho ótimo do disco rígido em relação ao armazenamento e recuperação de dados. Quando se conhece a maneira que o sistema operacional atua nesse gerenciamento, fica mais fácil de se optar pela melhor estratégia, implementando-a de acordo com as necessidades das aplicações e dos usuários. A aplicação demonstra de forma simplificada a idéia de como o acesso a um disco rígido é efetuado pelo sistema operacional mediando o algoritmo FCFS, facilitando assim o entendimento e compreensão por parte dos alunos e iniciantes.

6. REFERÊNCIA BIBLIOGRÁFICA

DEITELL, H. M., DEITEL, P. J., CHOFFNES, D. R. **Sistemas Operacionais**. Editora Prentice Hall, 3ª Edição. SP. 2005.

SILBERCHATZ, A.; Galvin, P.; Gane, G. **Applied Operating System Concepts**. John Wiley & Sons Inc., first edition. 2000.

OLIVEIRA, R.; Carissimi, A.; Toscani, S. **Sistemas Operacionais**. Série de Livros Didáticos do Instituto de Informática da UFRGS (nro. 11). Sagra-Luzzato. Primeira edição. 2001.

