

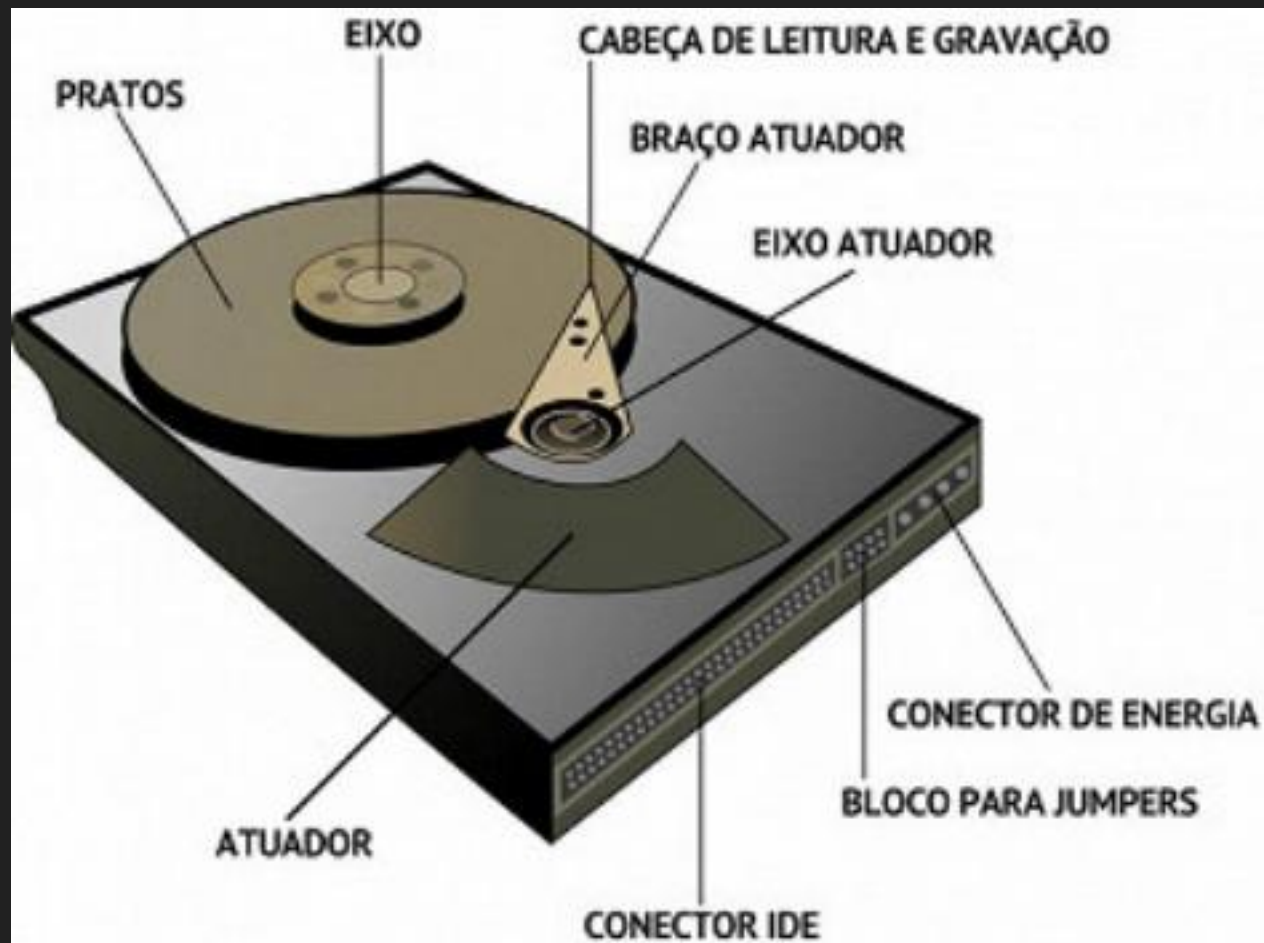
Sistemas de arquivos

Nome:Guilherme Londero, Guilherme Moraes, João Victor

Discos Magnéticos -ou HD(disco rígido)

É a parte do computador onde são armazenadas as informações. É caracterizado como memória física, não-volátil, O disco rígido é um sistema lacrado contendo discos de metal recobertos por material magnético onde os dados são gravados através de cabeças, e revestido externamente por uma proteção. É nele que normalmente gravamos dados (informações) e a partir dele que o sistema operacional é salvo (a não ser que tenha um ssd) e carregamos os dados dos programas que utilizamos.

Para que um computador comece a funcionar — por exemplo, quando é ligado ou reinicializado — ele deve ter um programa inicial para executar. Esse programa bootstrap inicial tende a ser simples. Ele inicializa todos os aspectos do sistema, dos registradores da CPU aos controladores de dispositivos e conteúdos da memória principal e, então, inicia o sistema operacional. Para realizar sua tarefa, o programa bootstrap encontra o kernel do sistema operacional no disco, carrega o kernel na memória e salta para um endereço inicial de modo a começar a execução do sistema operacional.



*Conceitos importantes

Em computação (especificamente em transmissão de dados e armazenamento de dados), um bloco, algumas vezes chamado de registro físico, **é uma sequência de bytes ou bits, normalmente contendo algum número completo de registros, que possui um tamanho máximo, chamado de tamanho de bloco.**

Na maioria dos computadores, o bootstrap é armazenado em memória somente-de-leitura (ROM — read-only memory). Essa locação é conveniente porque a ROM não precisa de inicialização e está em uma locação fixa em que o processador pode começar a execução quando ligado ou reinicializado. Além disso, já que a ROM é somente-de-leitura, ela não pode ser infectada por um vírus de computador. O problema é que a alteração desse código de bootstrap requer a alteração dos chips de hardware da ROM

Este sistema é necessário porque o conteúdo da memória RAM é apagado quando o computador é desligado. Desta forma, temos um meio de executar novamente programas e carregar arquivos contendo os dados da próxima vez que o computador for ligado. O disco rígido é também chamado de memória de massa ou ainda de memória secundária. Nos sistemas operacionais o disco rígido é também utilizado para expandir a memória RAM, através da gestão de memória virtual (quando a memória ram está cheia o sistema operacional tenta criar essa memória virtual para ainda continuar executando suas tarefas).

Discos de Estado Sólido - ou SSD

Mesma função que o HD, porém, é muito mais rápido que o HD, ou seja velocidade de gravar e ler dados é maior, conseqüentemente por ser uma tecnologia melhor o preço do SSD é mais caro.

Conexão do Disco

Os computadores acessam a memória em disco de duas formas. Uma forma é por meio de portas de I/O (ou armazenamento conectado ao hospedeiro, O armazenamento conectado ao hospedeiro é aquele acessado por portas de I/O locais.); isso é comum em sistemas pequenos. A outra forma é por meio de um hospedeiro remoto em um sistema de arquivos distribuído; isso é chamado armazenamento conectado à rede.

Armazenamento conectado à rede é um sistema de armazenamento de uso específico que é acessado remotamente por meio de uma rede de dados. Os clientes acessam o armazenamento conectado à rede por meio de uma interface de chamadas de procedimento remotas, como o NFS em sistemas UNIX ou o CIFS em máquinas Windows.

O armazenamento conectado à rede fornece um meio conveniente para que todos os computadores em uma LAN compartilhem um pool de espaços de armazenamento, com a mesma facilidade de nomeação e acesso inerentes ao armazenamento local conectado ao hospedeiro. No entanto, ele tende a ser menos eficiente e apresentar pior desempenho do que algumas opções de armazenamento de conexão direta. Também tem a desvantagem que as operações I/O de armazenamento consomem largura de banda da rede de dados, aumentando assim a latência(ping) da comunicação de rede. Esse problema pode ser particularmente grave em grandes instalações cliente-servidor — a comunicação entre servidores e clientes compete por largura de banda com a comunicação entre servidores e dispositivos de armazenamento.

Scheduling de Disco

Scheduling FCFS

o algoritmo primeiro-a-chegar, primeiro-a-ser-atendido (FCFS). Esse algoritmo é intrinsecamente justo, mas ele geralmente não fornece o serviço mais rápido porque o cabeçote pode estar muito longe de onde ele precisa ser operado, aí o tempo de deslocamento dele pode ser grande.

Scheduling SSTF

algoritmo do tempo-de-busca-mais-curto-primeiro (SSTF — shortest-seek-time-first). O algoritmo SSTF seleciona a solicitação com o menor tempo de busca a partir da posição corrente do cabeçote. O SSTF seleciona a solicitação pendente mais próxima da posição corrente do cabeçote. É melhor que o FCFS mas não é o melhor.

Scheduling SCAN

No algoritmo de escalonamento de disco SCAN, o cabeçote começa em uma extremidade do disco e se move em direção à outra extremidade, atendendo às requests entre uma a uma e chegando à outra extremidade. Então, a direção da cabeça é invertida e o processo continua enquanto a cabeça faz a varredura continuamente para frente e para trás para acessar o disco. Portanto, esse algoritmo funciona como um elevador e, portanto, também é conhecido como algoritmo de elevador . Como resultado, as requests no midrange são mais atendidas e aqueles que chegam atrás do braço do disco terão que esperar.

Desvantagens do algoritmo SCAN (elevador)

Algoritmo mais complexo para implementar.

Este algoritmo não é justo porque causa um longo tempo de espera para os cilindros que acabam de ser visitados pelo cabeçote.

Isso faz com que a cabeça se mova até o final do disco, dessa forma, as requests que chegam antes da posição do braço receberão serviço imediato, mas algumas outras requests que chegarem atrás da posição do braço terão que esperar a conclusão da solicitação.

Scheduling C-SCAN

É uma variante do algoritmo SCAN projetada para fornecer um tempo de espera mais uniforme. Como o SCAN, o scheduling C-SCAN move o cabeçote de uma extremidade à outra do disco, atendendo às solicitações ao longo do caminho. No entanto, quando o cabeçote alcança a outra extremidade, ele retorna imediatamente ao começo do disco sem atender a nenhuma solicitação na viagem de volta. O algoritmo de scheduling C-SCAN trata os cilindros, essencialmente, como uma lista circular que é percorrida até o último cilindro e retorna ao primeiro.

Conceito de Arquivo

Um arquivo é uma coleção nomeada de informações relacionadas que são gravadas em memória secundária. Da perspectiva do usuário, um arquivo é a menor unidade de armazenamento lógico secundário; isto é, os dados não podem ser gravados na memória secundária se não estiverem em um arquivo. Normalmente, os arquivos representam programas (nas formas fonte e objeto) e dados. Os arquivos de dados podem ser numéricos, alfabéticos, alfanuméricos ou binários. Os arquivos podem ter forma livre, como os arquivos de texto, ou podem ser formatados rigidamente. Em geral, um arquivo é uma sequência de bits, bytes, linhas ou registros, cujo significado é definido pelo criador e usuário do arquivo. Assim, o conceito de arquivo é extremamente geral.

As informações em um arquivo são definidas por seu criador. Muitos tipos diferentes de informações podem ser armazenados em um arquivo — Um arquivo tem uma estrutura específica definida, que depende de seu tipo. Um arquivo de texto é uma sequência de caracteres organizada em linhas. Um arquivo-fonte é uma sequência de funções, todas organizadas como declarações seguidas por comandos executáveis. Um arquivo executável é uma série de seções de código que o carregador pode trazer para a memória e executar.

De forma muito objetiva podemos dizer que um arquivo possui pelo menos três características que os definem:

Nome: E rótulo do arquivo, ao se escolher um nome para o arquivo, é recomendável evitar o uso de caracteres especiais como !@#\$%*+?:><>[]|. Alguns Sistemas Operacionais permitem o uso de tais símbolos, mas ao se veicular estes arquivos na Internet ou por pendrives nunca sabemos o destino final e nem mesmo se sua compatibilidade será satisfeita em outros ambientes operacionais.

Tipo (extensão): Devido a importância desta característica, destacamos um tópico abaixo somente para tratar este item.

Tamanho: Esta informação é mensurada dentro dos padrões da informática, este é um tema que já tratamos em postagens anteriores, para maiores informações veja:

Operações de Arquivo

Um arquivo é um tipo de dado abstrato. Para definir um arquivo apropriadamente, precisamos considerar as operações que podem ser executadas sobre arquivos. O sistema operacional pode fornecer chamadas de sistema para criar, gravar, ler, reposicionar, excluir e truncar arquivos

Criação de um arquivo. Deve ser encontrado espaço para o arquivo no sistema de arquivos. Em segundo lugar, deve ser criada uma entrada para o novo arquivo no diretório.

Gravação em um arquivo. Para gravar em um arquivo, fazemos uma chamada de sistema especificando tanto o nome do arquivo quanto as informações a serem gravadas no arquivo. Dado o nome do arquivo, o sistema pesquisa o diretório para determinar a localização do arquivo.

- Leitura de um arquivo. Para ler a partir de um arquivo, usamos uma chamada de sistema que especifica o nome do arquivo e onde (na memória) o próximo bloco do arquivo deve ser inserido. Novamente, o diretório é pesquisado em busca da entrada associada, e o sistema precisa manter um ponteiro de leitura para a localização no arquivo em que a próxima leitura deve ocorrer. Uma vez que a leitura tenha ocorrido, o ponteiro de leitura é atualizado. Já que um processo usualmente tanto faz leituras de um arquivo quanto gravações em um arquivo, a localização da operação corrente pode ser mantida como um ponteiro da posição corrente do arquivo por processo. Tanto as operações de leitura quanto as de gravação usam esse mesmo ponteiro, economizando espaço e reduzindo a complexidade do sistema.

- Reposicionamento dentro de um arquivo. O diretório é pesquisado em busca da entrada apropriada, e o ponteiro da posição corrente do arquivo é reposicionado para determinado valor. O reposicionamento dentro de um arquivo não precisa envolver nenhum I/O real. Essa operação de arquivo também é conhecida como busca em arquivo.
- Exclusão de um arquivo. Para excluir um arquivo, pesquisamos o diretório em busca do arquivo nomeado. Tendo encontrado a entrada associada no diretório, liberamos todo o espaço do arquivo, para que ele possa ser reutilizado por outros arquivos, e apagamos a entrada no diretório.
- Truncamento de um arquivo. O usuário pode querer apagar o conteúdo de um arquivo, mas manter seus atributos. Em vez de forçar o usuário a excluir o arquivo e então recriá-lo, essa função permite que todos os atributos permaneçam inalterados — exceto o tamanho — mas deixa que o arquivo seja redefinido com o tamanho zero e seu espaço seja liberado.

Tipos de Arquivo

Ao projetar um sistema de arquivos — na verdade, um sistema operacional inteiro — sempre consideramos se o sistema operacional deve reconhecer e suportar tipos de arquivo. Se um sistema operacional reconhece o tipo de um arquivo, pode então operar sobre o arquivo de formas racionais. Por exemplo, um erro comum ocorre quando um usuário tenta dar saída na forma objeto binária de um programa. Normalmente essa tentativa produz lixo; no entanto, a tentativa pode ser bem-sucedida se o sistema operacional tiver sido informado de que o arquivo é um programa-objeto binário.

Uma técnica comum para a implementação dos tipos de arquivo é incluir o tipo como parte do nome do arquivo. O nome é dividido em duas partes — um nome e uma extensão, usualmente separados por um ponto. Dessa forma, o usuário e o sistema operacional podem identificar, somente pelo nome, qual é o tipo de um arquivo.

Diretório

Modo como o sistema organiza os diferentes arquivos contidos num disco. É a estrutura de dados que contém entradas associadas aos arquivos onde estão informações como localização física, nome, organização e demais atributos.

Quando um arquivo é aberto, o sistema operacional procura a sua entrada na estrutura de diretórios, armazenando as informações do arquivo em uma tabela mantida na memória principal. Esta tabela contém todos os arquivos abertos, sendo fundamental para aumentar o desempenho das operações com arquivos.

Nível Único (single-level directory): é a implementação mais simples, existe apenas um único diretório contendo todos os arquivos do disco. É bastante limitado já que não permite que usuários criem arquivos com o mesmo nome

User File Directory (UFD): para cada usuário existe um diretório particular e assim poderia criar arquivos com qualquer nome, que á um nível de diretório adicional para controle chamado de Master File Directory (MFD) que é indexado pelo nome do usuário e cada entrada aponta para o diretório pessoal.

Estrutura de diretórios em Árvore (Tree Structured Directory), Adotado pela maioria dos sistemas operacionais e é logicamente melhor organizado.

É possível criar quantos diretórios quiser, podendo um diretório conter arquivos e outros diretórios (chamados subdiretórios).

Cada arquivo possui um único path, descreve todos os diretórios da raiz até o diretório onde o arquivo está ligado e na maioria dos sistemas os diretórios são tratados como arquivos tendo atributos e identificação.

Métodos de Acesso

Arquivos armazenam informações. Para serem usadas, essas informações devem ser acessadas e lidas para a memória do computador. As informações no arquivo podem ser acessadas de várias maneiras. Alguns sistemas fornecem apenas um método de acesso a arquivos, enquanto outros suportam muitos métodos de acesso.

Acesso Sequencial

O método de acesso mais simples é o acesso sequencial. As informações no arquivo são processadas em ordem, um registro após o outro. Essa modalidade de acesso é sem dúvida a mais comum; por exemplo, editores e compiladores usualmente acessam arquivos dessa forma.

Acesso Direto

Outro método é o acesso direto (ou acesso relativo). Aqui, o arquivo é composto por registros lógicos de tamanho fixo que permitem que os programas leiam e gravem registros rapidamente sem uma ordem específica. O método de acesso direto é baseado em um modelo de arquivo em disco, já que os discos permitem acesso randômico a qualquer bloco do arquivo. No acesso direto, o arquivo é considerado como uma sequência numerada de blocos ou registros.

Outros Métodos de Acesso

Outros métodos de acesso podem ser construídos no topo de um método de acesso direto. Esses métodos envolvem, em geral, a construção de um índice para o arquivo. O índice, semelhante a um índice da parte final de um livro, contém ponteiros para os vários blocos. Para encontrar um registro em um arquivo, primeiro pesquisamos o índice e, então, usamos o ponteiro para acessar o arquivo diretamente e encontrar o registro desejado.

Estrutura do Sistema de Arquivos

Os discos oferecem a maior parte da memória secundária(HD, SSD, etc) em que os sistemas de arquivos são mantidos. Duas características tornam os discos convenientes para esse fim:

- 1-Um disco pode ser regravado in loco; é possível ler um bloco do disco, modificar o bloco e gravá-lo de volta no mesmo local.
- 2.Um disco pode acessar diretamente qualquer bloco de informações que ele contém. Portanto, é simples acessar qualquer arquivo sequencial ou randomicamente, e a permutação de um arquivo para outro requer apenas a movimentação dos cabeçotes de leitura-gravação e a espera pela rotação do disco.

Os sistemas de arquivos fornecem acesso eficiente e conveniente ao disco permitindo que dados sejam armazenados, localizados e recuperados facilmente. Um sistema de arquivos apresenta dois problemas de projeto bem diferentes. O primeiro problema é a definição da aparência que o sistema de arquivos deve ter para o usuário.

Essa tarefa envolve a definição de um arquivo e seus atributos, as operações permitidas sobre um arquivo e a estrutura de diretórios para a organização de arquivos. O segundo problema é a criação de algoritmos e estruturas de dados para mapear o sistema de arquivos lógico para os dispositivos físicos de memória secundária.

O próprio sistema de arquivos é, geralmente, composto por vários níveis diferentes.

Cada nível do projeto usa os recursos dos níveis mais baixos para criar novos recursos a serem usados pelos níveis mais altos.

O nível de controle de I/O é composto por drivers de dispositivos e manipuladores de interrupções para transferir informações entre a memória principal e o sistema de discos. Sua entrada é composta por comandos de alto nível como “recuperar o bloco 123”. A saída é composta por instruções de baixo nível específicas de hardware que são usadas pelo controlador de hardware que interliga o dispositivo de I/O ao resto do sistema.

O sistema de arquivos básico precisa apenas emitir comandos genéricos para o driver de dispositivo apropriado para ler e gravar blocos físicos em disco.

Cada bloco físico é identificado por seu endereço de disco numérico (por exemplo, drive 1, cilindro 73, trilha 2, setor 10). Essa camada também gerencia os buffers e caches da memória que contêm vários sistemas de arquivos, diretórios e blocos de dados.

Um bloco no buffer é alocado antes que possa ocorrer a transferência de um bloco de disco. Quando o buffer está cheio, o seu gerenciador tem que encontrar mais memória no buffer ou liberar espaço no buffer para permitir que o I/O solicitado seja concluído.

Os caches são usados para armazenar metadados do sistema de arquivos que são frequentemente utilizados, a fim de melhorar o desempenho; portanto, o gerenciamento do seu conteúdo é crítico para o desempenho ótimo do sistema.

Várias estruturas em disco e em memória são usadas para implementar um sistema de arquivos. Essas estruturas variam, dependendo do sistema operacional e do sistema de arquivos, mas alguns princípios gerais são aplicáveis.

Em disco, o sistema de arquivos pode conter informações sobre como inicializar um sistema operacional armazenado nesse local, o número total de blocos, o número e a localização de blocos livres, a estrutura do diretório e arquivos individuais.

Uma estrutura de diretório (por sistema de arquivos) é usada para organizar arquivos. No UFS, ela inclui os nomes de arquivo e os números associados. No NTFS, a estrutura é armazenada na tabela de arquivos mestre.

As informações em memória são usadas tanto no gerenciamento do sistema de arquivos quanto na melhoria do desempenho por meio do armazenamento em cache. Os dados são carregados em tempo de montagem, atualizados durante operações sobre o sistema de arquivos e descartados na desmontagem. Vários tipos de estruturas podem ser incluídos:

- Uma tabela de montagens em memória contém informações sobre cada volume montado.
- Um cache em memória da estrutura de diretórios mantém as informações referentes aos diretórios acessados recentemente. (Para diretórios nos quais há volumes montados, ele pode conter um ponteiro para a tabela de volumes.)
- A tabela de arquivos abertos em todo o sistema contém uma cópia do FCB de cada arquivo aberto, assim como outras informações.
- A tabela de arquivos abertos por processo contém um ponteiro para a entrada apropriada na tabela de arquivos abertos em todo o sistema, assim como outras informações.x
- Buffers mantêm blocos do sistema de arquivos quando eles estão sendo lidos de disco ou gravados em disco.

Lista Linear

O método mais simples de implementação de um diretório é utilizar uma lista linear de nomes de arquivos com ponteiros para os blocos de dados. Esse método é simples de programar, mas de execução demorada. Para criar um novo arquivo, primeiro devemos pesquisar o diretório para termos certeza de que nenhum arquivo existente tenha o mesmo nome. Em seguida, adicionamos uma nova entrada ao final do diretório. Para excluir um arquivo, pesquisamos o diretório em busca do arquivo nomeado e, então, liberamos o espaço a ele alocado.

A desvantagem real de uma lista linear de entradas de diretório é que o encontro de um arquivo exige uma busca linear. Informações de diretório são usadas com frequência, e os usuários notarão que o acesso a elas é lento.

Tabela de Dispersão (Tabela Hash)

Uma tabela hash, é uma estrutura de dados utilizada para tornar o processo de busca mais eficiente. Assim, esta estrutura de dados é muito utilizada não para inserções ou remoções, mas quando há a necessidade de realizar muitas buscas e com rápido tempo de resposta.

Para descobrir se um elemento está ou não no vetor precisamos percorrer todo o vetor procurando pelo elemento. Isso pode ser muito demorado dependendo do tamanho do vetor. Em uma tabela hash podemos acessar diretamente a posição da tabela onde o elemento deve estar caso exista.

Métodos de Alocação

A natureza de acesso direto dos discos oferece-nos flexibilidade na implementação de arquivos. Em quase todos os casos, muitos arquivos são armazenados no mesmo disco. O principal problema é como alocar espaço para esses arquivos de modo que o espaço em disco seja eficientemente utilizado e os arquivos possam ser rapidamente acessados. Três métodos principais de alocação de espaço em disco estão sendo muito usados: contíguo, encadeado e indexado. Cada método apresenta vantagens e desvantagens. Embora alguns sistemas suportem todos os três, o mais comum é que um sistema utilize um método para todos os arquivos de um tipo de sistema de arquivos.

Alocação Contígua

A alocação contígua consiste em armazenar um arquivo em blocos sequencialmente dispostos, permitindo ao sistema localizar um arquivo através do endereço do primeiro bloco e da sua extensão em blocos. O acesso é feito de maneira simples, tanto para a forma sequencial quanto para a direta.

Um problema desse tipo de alocação é que quando um arquivo é criado com n blocos, é necessário que exista uma cadeia de n blocos livres disposto sequencialmente. Nesse tipo de alocação, o disco é visto como um grande vetor, com segmentos ocupados e livres.

Alocação Encadeada

A alocação encadeada consiste em manter os arquivos, cada um, como uma lista encadeada de blocos de disco. Dessa forma uma parte de cada bloco é usada como ponteiro para o próximo bloco. O restante do bloco é usado para dados, diferente da alocação contígua os blocos de disco podem estar espalhados.

Uma vantagem desse tipo de alocação é que o tamanho do arquivo não precisa ser conhecido antes de sua criação, já que cada bloco terá um ponteiro para o próximo bloco. Assim o fato de o disco estar fragmentado não ocasiona problemas para a criação do arquivo

O problema deste tipo de alocação é o tempo de leitura extremamente lento. O acesso deverá ser sempre sequencial. Assim, para acessar um bloco intermediário será necessário percorrer o arquivo desde o início.

Alocação Indexada

Alocação indexada é uma forma de resolver o problema do tempo de leitura da alocação por lista encadeada. Isso é feito por meio de uma tabela de endereços dos blocos ocupados pelo arquivo. Para cada novo bloco alocado, inclui-se mais um novo item na tabela.

Usando alocação indexada, o acesso aleatório fica mais fácil, pois não é necessário percorrer todos os blocos de forma sequencial, o endereço de cada bloco fica armazenado na tabela de alocação.

A desvantagem desta abordagem é que a tabela alocação deverá ser mantida na memória principal e dependendo do tamanho dos blocos do disco ocupará muito espaço de memória o tempo todo.