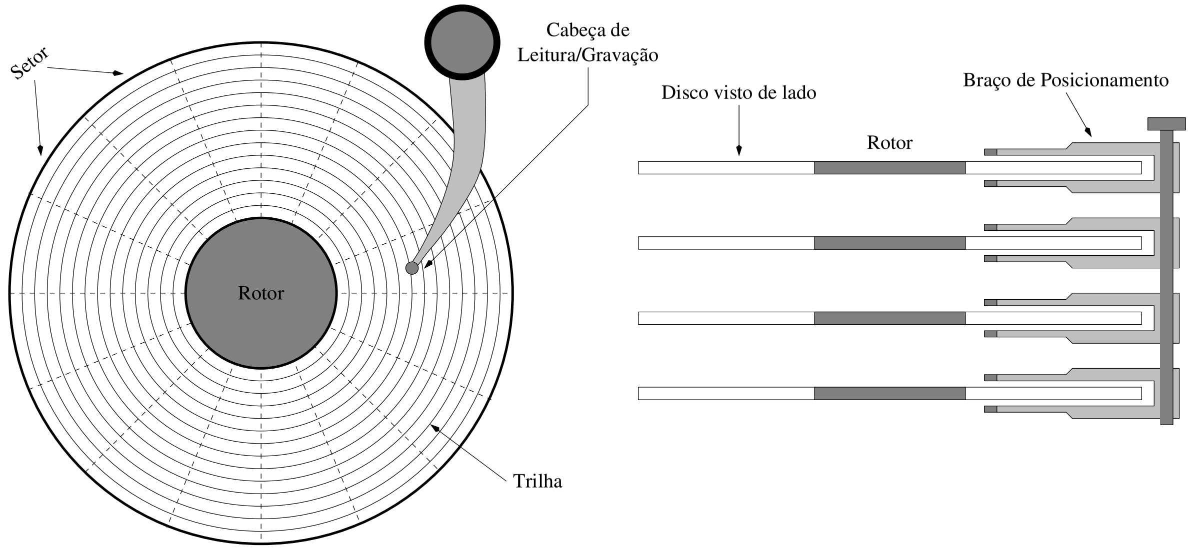
5.4 - Discos Rotativos

★ elementos básicos de um disco rotacional: cabeça de leitura/gravação, rotor, braço de posicionamento, cilindro, trilha e setor.



5.4.1 - Hardware do Disco

★ todos os discos rotativos são organizados em cilindros, cada qual contendo tantas trilhas quanto cabeças empilhadas verticalmente;

- ... as trilhas são divididas em setores, com um número de setores na circunferência, tipicamente entre 8 a 32 setores por trilha;

- ... todos os setores contém o mesmo número de bytes, embora setores no anel externo do disco sejam fisicamente maiores que aqueles no anel interno, mas isto não permite que o espaço extra seja aproveitado.

★ um aspecto que tem importante implicações no *disk driver* é a possibilidade do controlador fazer buscas *(seek)* em dois ou mais dispositivos ao mesmo tempo, que por sua vez são conhecidas como busca entrelaçada *(overlapped seek)*;

- ... enquanto o controlador e o *software* estão esperando uma busca se completar em um dispositivo, o controlador pode iniciar uma busca em outro;

... 5.4.1 - Hardware do Disco

- ... muitos controladores podem também ler ou escrever em um dispositivo enquanto executam uma busca em um ou mais de um dispositivos, mas nenhum pode ler ou escrever em dois dispositivos no mesmo tempo;

- ... isto decorre do fato de que ler ou escrever requer que o controlador mova bits na faixa de μs, assim uma transferência usa muito de sua capacidade computacional;

- ... a habilidade de realizar duas ou mais buscas ao mesmo tempo pode reduzir o tempo médio de acesso de modo considerável.

★ **IBM PC** - são parâmetros padrões para unidades de disquetes de 5.25", dupla-face/dupla-densidade e capacidade para 360 KB.

... 5.4.1 - Hardware do Disco

★ os parâmetros de dispositios para discos flexíveis do IBM PC são mostrados abaixo:

- ... esses são os parâmetros padrões para unidades de disquetes de 5.25", dupla-face/dupla-densidade e capacidade para 360 Kbytes.

número de cilindros	40	tempo de busca (cilindros adjacentes)	6 ms
trilhas por cilindro	2	tempo de busca (médio)	77 ms
setores por trilha	9	tempo de rotação	200 ms
total de setores	720	tempo de partida/parada	250 ms
bytes por setor	512	tempo para transferir um setor	22 ms
bytes por disquete	368640		

5.4.2 - Software do Disco

★ dentre as características genéricas relacionadas com os *disk drivers*, o tempo de leitura ou escrita de um bloco do disco é um fator determinante:

- ... tempo de *seek* - tempo para mover o braço para o cilindro desejado;
- ... atraso rotacinal - tempo para o setor desejado ficar sob a cabeça de leitura/escrita;
- ... tempo de transferência do bloco.

★ para muitos discos, o tempo de *seek* domina, assim reduzindo-se o tempo de *seek*, podemos melhorar substancialmente o desempenho do sistema.

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

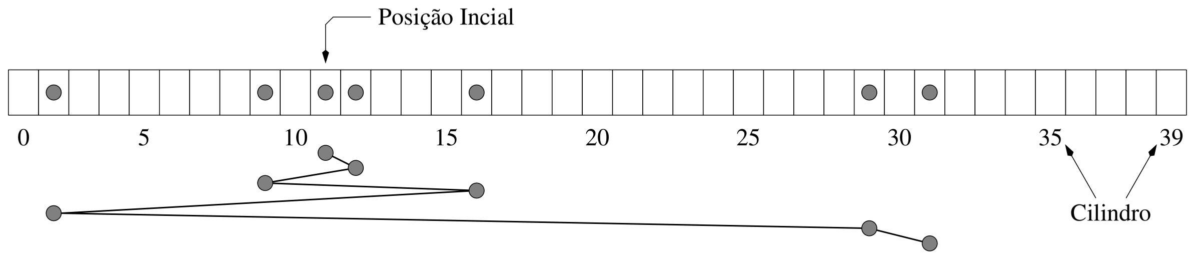
★ se o *disk driver* aceita uma requisição por vez e a executa nesta ordem, isto é, *First-Come, First-Served (FCFS)*, pouco pode ser feito para otimizar o tempo de *seek* uma vez não há a possibilidade de pré-processamento;

- ... **estratégia alternativa** - é provável que enquanto o braço está executando um *seek* na metade de uma requisição, uma outra requisição de disco pode ter sido gerada por outro processo;
- ... muitos *disk drivers* mantém uma tabela, indexada pelo número do cilindro, com todas as requisições pendentes para cada cilindro, encadeadas juntas numa lista;
- ... para este tipo de estrutura de dados (fila de requisições), podemos melhorar o algoritmo de escalonamento *First-Come, First-Served.*

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ em um disco de 40 cilindros, uma requisição chega para ler um bloco no cilindro 11. Enquanto a busca para o cilindro 11 está em progresso, novas requisições chegam para os cilindros 1, 36, 16, 34, 9, e 12, nesta ordem;

- ... elas são inseridas na tabela de requisições pendentes, tendo cada cilindro um lista separada conforme apresentado na figura abaixo.



... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ quando a requisição corrente termina (cilindro 11), o *disk driver* tem que escolher qual será a próxima requisição a ser atendida;

- ... usando *first come, first served*, ele irá para o cilindro 1, então para o 36, e assim por diante — este algoritmo requer movimentos do braço de 10, 35, 20, 18, 25, e 3, respectivamente, num total de 111 cilindros.

★ alternativamente, a próxima requisição pode ser manuseada a fim de minimizar o tempo de *seek*, ou seja, com a sequência 12, 9, 16, 1, 34, e 36, os movimentos do braço são 1, 3, 7, 15, 33, e 2, num total de 61 cilindros;

- ... este algoritmo, *Small Seek First* (SSF), diminuiu o total de movimentos do braço pela metade, comparado com o FCFS (*First Come, First Served*).

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

- **Problema no SSF** - suponha que mais requisições estão chegando enquanto as requisições 12, 9, 16, 1, 34 e 36 estão sendo processadas:
- ... por exemplo, se, após chegar ao cilindro 16, uma nova requisição para o cilindro 8 está presente e, assim, terá prioridade sobre o cilindro 1;
- ... se a requisição for para o cilindro 13, o braço irá atender esta requisição (vai p/ o cilindro 13) ao invés de ir para o cilindro 1.

★ com disco muito carregado, o braço tende a permanecer no meio do disco a maior parte do tempo, prejudicando assim as requisições das extremidades;

- ... requisições distantes do meio são em média mais demoradas, colocando o objetivo de mínima resposta no tempo e equitatividade em conflito.

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ **alternativa** - reconciliar os objetivos conflitantes entre a eficiência e equitatividade constitui-se em manter o movimento do braço na mesma direção até não haver mais requisições pendentes naquela direção, e então o movimento do braço é mudado.

★ este algoritmo (*algoritmo do elevador*), requer que o *software* mantenha 1 bit para indicar a direção corrente, *UP* ou *DOWN*;

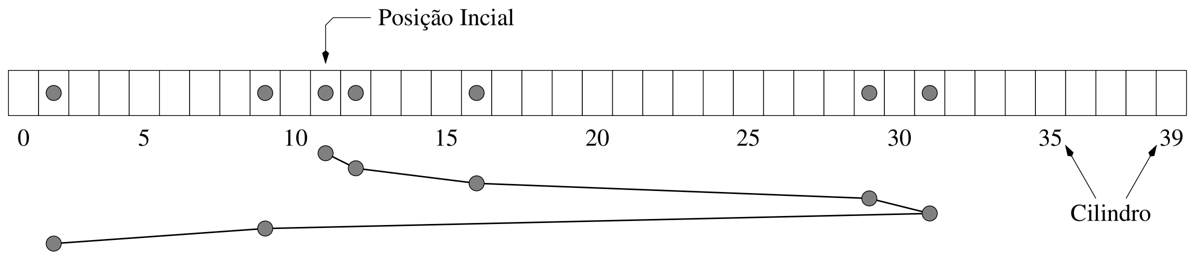
- ... quando a requisição termina, o *disk driver* testa o bit, se é *UP*, o braço é movido para a próxima requisição pendente de posições mais altas, se houver;
- se não há requisições pendentes para posições mais altas, o bit de direção é revertido (mudando o sentido) para a próxima requisição de posição mais baixa, se houver.

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ para ilustrar o algoritmo do elevador, assuma que o bit de direção esteja inicialmente em UP e que a ordem na qual os cilindros são servidos é 12, 16, 34, 36, 9, e 1;

- ... esta ordem gera movimentos do braço de 1, 4, 18, 2, 27, e 8, num total de 60 cilindro, ou seja, ligeiramente melhor que SSF, embora seja usualmente pior;

- ... uma propriedade interessante do algoritmo do elevador é que dada uma coleção de requisições, o limite superior para o total de movimentos é fixado: ele é apenas duas vezes o número de cilindros.



... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ alguns controladores de disco provêem um modo do *software* para inspecionar o número de setores correntes sob a cabeça, o que possibilita uma outra otimização;

- ... se duas ou mais requisições para o mesmo cilindro estão pendentes, o *driver* pode emitir a requisição para o setor que passará sob a cabeça do próximo cilindro;

- ... note que quando trilhas múltiplas estão presentes num cilindro, requisições consecutivas podem ser conduzidas para diferentes trilhas com nenhuma penalidade;

- ... o controlador pode selecionar alguma das cabeças instantâneamente, uma vez que seleção de cabeça não involve movimento dos braços nem atraso rotacional.

... 5.4.2 - Software do Disco - Algoritmos de Escalonamento

★ quando existe vários dispositivos, uma tabela de requisições pendentes deve ser mantida para cada dispositivo separadamente;

- ... quando algum dispositivo está desocupado, um *seek* deve ser emitido para mover os seus braços para o cilindro onde será necessário (assumindo que o controlador permita *seeks* sobrepostos);

- ... quando a transferência corrente termina, um teste pode ser feito para verificar se algum dos dispositivos está posicionado no cilindro correto;

- ... se um ou mais estão, a próxima transferência pode ser iniciada no dispositivo que já está no cilindro correto (otimização);

- ... se nenhum está posicionado, o *driver* deve emitir novos *seeks* sobre os dispositivos que já completaram a transferência, e esperar a próxima interrupção para examinar em qual dispositivo o posicionamento do braço se completou.

5.4.2 - Software do Disco - Manipulação de Erros

★ Discos RAM não têm preocupação sobre *seek* ou na otimização rotacional: em algum instante todos os blocos podem ser lidos ou escritos sem nenhum movimento físico;

★ uma outra área na qual discos RAM são consideravelmente mais simples que discos rotativos é a manipulação de erros. Discos rotativos são submetidos por uma larga variedade de erros. Alguns dos mais comuns são:

- erros de programação — requisição para setor não existente;

- erro de *checksum* transiente — causado por sujeira na cabeça;

- erro de *checksum* permantente — bloco do disco fisicamente danificado;

- erro de *seek* — enviar o braço para o cilindro 6, mas ele vai para o 7;

- erro de controlador —recusa do controlador em aceitar comandos.

... 5.4.2 - Software do Disco - Manipulação de Erros

★ **erros de programação** - ocorrem quando o *driver* diz ao controlador para executar uma operação de *seek* em um cilindro não existente;

- ... ler de um setor não existente, usar uma cabeça não existente, ou transferir de ou para uma posição de memória inexistente.

★ **erros checksum** - transientes são causados por poeira no ar entre a cabeça e a superfície do disco provocando erros de leitura;

- ... em muitos casos eles podem ser eliminados pela repetição da operação algumas vezes. Se o erro persiste, o bloco deve ser marcado como defeituoso.

... 5.4.2 - Software do Disco - Manipulação de Erros

★ **evitando blocos defeituosos** - uso de um programa especial que toma a relação destes blocos como entrada, e cria um arquivo contendo todos os blocos defeituosos;

- ... uma vez que este arquivo tenha sido criado, para o alocador do disco parecerá que estes blocos estão ocupados, não os alocando para outros arquivos;

- ... como o arquivo de blocos defeituosos nunca é lido, os blocos defeituosos permanecerão inertes no disco.

★ **tarefa árdua** - alguns controladores "inteligentes" reservam algumas poucas trilhas, não normalmente disponíveis para programas do usuário;

- ... quando um disco é formatado, o controlador determina quais blocos são defeituosos e automaticamente substitui por uma trilha de reserva;

- ... como a tabela contendo esse mapeamento é mantida na memória interna do controlador e no disco, a substituição é transparente para o *driver*.

... 5.4.2 - Software do Disco - Manipulação de Erros

★ **erros de seek** - são causados por problemas mecânicos no braço;

- ... mantendo o rastreamento da posição do braço internamente, para realizar um *seek*, o controlador emite uma série de pulsos para o motor do braço, um pulso por cilindro, para mover o braço para o novo cilindro;
- ... quando o braço chega no destino, o controlador lê o número do cilindro (escrito quando da formatação) gerando um erro de *seek* caso o braço esteja no lugar errado.

★ alguns computadores corrigem o erro de *seek* automaticamente, mas outros incluindo os IBM PC's apenas atribuem um bit de erro e abandonam o resto para o *driver*;

- ... o *driver* manipula este erro pela emissão de um comando **recalibrate**, que move o braço tão longe ele possa ir, e reinicia o controlador interno do cilindro corrente em 0, o que usualmente soluciona o problema;
- ... caso contrário, o dispositivo deve ser reparado.

... 5.4.2 - Software do Disco - Manipulação de Erros

★ resumidamente, o controlador é um pequeno computador especializado, contendo *software*, variáveis, *buffers*, e ocassionalmente *bugs*;

- ... algumas vezes uma sequência não usual de eventos, tal como uma interrupção sobre um dispositivo juntamente com um comando **recalibrate** em outro, dispara o *bug* e causa o controlador entrar em *loop* ou perder-se do estava fazendo.
- ... projetistas de controladores usualmente consideram a pior situação e provêem um pino no *chip*, o qual, quando em nível alto, força o controlador esquecer sua tarefa corrente, reiniciando a si mesmo;
- ... se tudo isso falhar, o *disk driver* pode reiniciar o controlador. Se isto também for em vão, o *driver* imprime uma mensagem e termina sua operação.

5.4.2 - Software do Disco - Cache de Rastreamento

- ✶ o tempo requerido para uma operação de *seek* para um novo cilindro é usualmente muito maior que o tempo de transferência ou rotação;

- ... em outras palavras, uma vez que o *driver* tenha posicionado o braço em algum lugar, pouco importa o tempo gasto para ler um setor ou uma trilha inteira.

- ✶ alguns *disk drivers* tiram proveito desta propriedade mantendo secretamente uma cache do rastreamento *(track-at-a-time cache)*, o que não é conhecido pelo *software* independente do dispositivo;

- ... se um setor é demandado e o mesmo encontra-se no cache, nenhuma transferência do disco é requerida.

- ✘ **cache de rastreamento** - complexidade do *software* e espaço req. de buffer;

- ✘ **cache de rastreamento** - transferência da cache para o programa que requisitou a operação deve ser feita pela CPU, usando um *loop* programado, ao invés de DMA.

... 5.4.2 - Software do Disco - Cache de Rastreamento

- ✶ alguns controladores aprimoram este processo, e fazem o *track-at-a-time caching* na sua memória interna, transparente para o *driver*;

- ... assim a transferência entre o controlador e a memória pode usar DMA.

- ✍ Note que ambos controlador e *driver* podem ler ou escrever trilhas inteiras em um único comando, mas que o *software* independente do dispositivo não pode;

- ... a razão para isto é que ele considera o disco como uma sequência linear de blocos, sem considerar como eles são divididos em trilhas e cilindros.