

## Cap. 11 - I/O Management and Disk Scheduling

- 11.1 - Dispositivos do Sistema de I/O
- 11.2 - Funções do Sistema de I/O
- 11.3 - Aspectos de Projeto do Sistema de I/O
- 11.4 - Bufferização do Sistema de I/O
- 11.5 - Escalonamento em Discos Rotativos
- 11.6 - Redundant Array of Independent Disks
- 11.7 - Memória Cache em Discos Rotativos
- 11.8 - Sistema de I/O no UNIX SVR4
- 11.9 - Sistema de I/O no Windows

## ... Cap. 11 - I/O Management and Disk Scheduling

- ☆ William STALLINGS; **Operating Systems: Internals and Design Principles**, New Jersey, Prentice-Hall, 1998, ISBN: 0-13-887407-7
- ☆ Eleri CARDOZO; Maurício MAGALHÃES; Luís F. FAINA; **Sistemas Operacionais**, Dep. de Eng. de Computação e Automação Industrial, Fac. de Engenharia Elétrica e de Computação, UNICAMP, 1996.

## 11.1 - Dispositivos do Sistema de I/O

- \* Dispositivos de I/O externos podem grosseiramente ser divididos em 03 categorias:
  - **human readable**: voltado para a comunicação com os usuários de computadores; p. ex.: monitor, *keyboard*, *mouse* e impressora;
  - **machine readable**: voltado para a comunicação com equipamentos eletrônicos; p. ex.: *drives* de fitas e discos, sensores, controladores e atuadores;
  - **communication**: voltado para a comunicação com dispositivos remotos; p. ex.: **drives** de linhas digitais e modems.

## ... 11.1 - Dispositivos do Sistema de I/O

- \* Exemplos de Dispositivos de I/O classificados segundo comportamento, parceiro e taxa de transferência de dados:

Device	Purpose	Partner	Data Rate (KB/s)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Voice Input	Input	Human	0.02
Scanner	Input	Human	200.00
Voice Output	Output	Human	0.60
Line Printer	Output	Human	1.00
Laser Printer	Output	Human	100.00
Graphics Display	Output	Human	30000.00
CPU to Frame Buffer	Output	Human	200.00
Network Terminal	Input / Output	Machine	0.05
Network – LAN	Input / Output	Machine	200.00
Optical Disk	Storage	Machine	500.00
Magnetic Tape	Storage	Machine	2000.00
Magnetic Disk	Storage	Machine	2000.00

## ... 11.1 - Dispositivos do Sistema de I/O

- \* Há muitas diferenças entre as classes e até mesmo dentro de uma mesma classe esta diferença pode ser grande. Dentre as várias diferenças, destacamos:
  - **taxa de transferência de dados:** pode haver diferenças de várias ordens de magnitude entre as taxas de transferências;
  - **tipo de aplicação:** o uso de um dispositivo para um dado fim tem influência no software e nas políticas do sistema operacional;
  - **complexidade de controle:** existe módulos de I/O para controle do dispositivo com um maior grau de complexidade e, portanto, pode exigir um maior esforço computacional na sua execução;
  - **unidade de transferência:** dados podem ser transferidos em *streams* de bytes ou caracteres (ex.: terminal de entrada/saída) ou em blocos (ex.: disco);
  - **representação dos dados:** diferentes tipos de codificação são utilizados por diferentes dispositivos, incluindo códigos de caracteres e convenções de paridade;
  - **condição de erros:** a natureza dos erros, a forma como são apresentados e quantidade de respostas disponíveis variam muito de dispositivo para dispositivo.

## ... 11.1 - Dispositivos do Sistema de I/O

- \* Exemplos de Dispositivos de I/O classificados segundo comportamento, parceiro e taxa de transferência de dados:

Device	Purpose	Partner	Data Rate (KB/s)
Keyboard	Input	Human	0.01
Mouse	Input	Human	0.02
Voice Input	Input	Human	0.02
Scanner	Input	Human	200.00
Voice Output	Output	Human	0.60
Line Printer	Output	Human	1.00
Laser Printer	Output	Human	100.00
Graphics Display	Output	Human	30000.00
CPU to Frame Buffer	Output	Human	200.00
Network Terminal	Input / Output	Machine	0.05
Network – LAN	Input / Output	Machine	200.00
Optical Disk	Storage	Machine	500.00
Magnetic Tape	Storage	Machine	2000.00
Magnetic Disk	Storage	Machine	2000.00

## 11.2 - Funções do Sistema de I/O

\* Três são as formas de prover I/O:

- **programada:** processo requisita um operação de entrada/saída para um módulo de entrada/saída e permanece em espera ocupada que a operação de entrada/saída requisitada se complete; quando então continua o seu processamento;
- **controlada por interrupção:** após requisitar uma operação de entrada/saída, o processo pode continuar a sua execução, entretanto o mesmo será interrompido mais a diante quando o módulo de entrada/saída completar a operação requisitada;
- **acesso direto a memória:** compõe-se de um módulo capaz de trocar dados entre a memória principal e o módulo de entrada/saída.

	No Interrupts	Use of Interrupts
I/O to Memory Transfer through processor	Programmed I/O	Interrupt Driven I/O
Direct I/O to Memory Transfer		Direct Memory Access (DMA)

## ... 11.2 - Funções do Sistema de I/O

\* Durante a evolução dos Sistemas Computacionais, em nenhuma outra área a evolução foi tão evidente quanto a das funções de entrada/saída, cujos passos foram:

- ❶ nos primeiros sistemas o processador controlava diretamente o dispositivo periférico, ou seja, **microprocessor-controlled devices**;
- ❷ com a adição de um controlador ou módulo de entrada/saída, o processador passa a utilizar entrada/saída programada sem interrupções;
- ❸ mesmo que o anterior, mas agora utilizando interrupções o processador não perde tempo esperando que uma operação de entrada/saída se complete;
- ❹ o módulo de entrada/saída passa a controlar diretamente a memória através do DMA e, assim, blocos de dados podem ser movidos de/para a memória sem a intermediação do processador exceto no início e fim da transferência;
- ❺ o módulo de entrada/saída sofre adicionamentos tornando-se um processador em separado, com um conjunto especializado de instruções voltadas para entrada/saída;
- ❻ adição de memória ao módulo de entrada/saída, tornando-o de fato um computador capaz de controlar um grande número de dispositivos de entrada/saída.

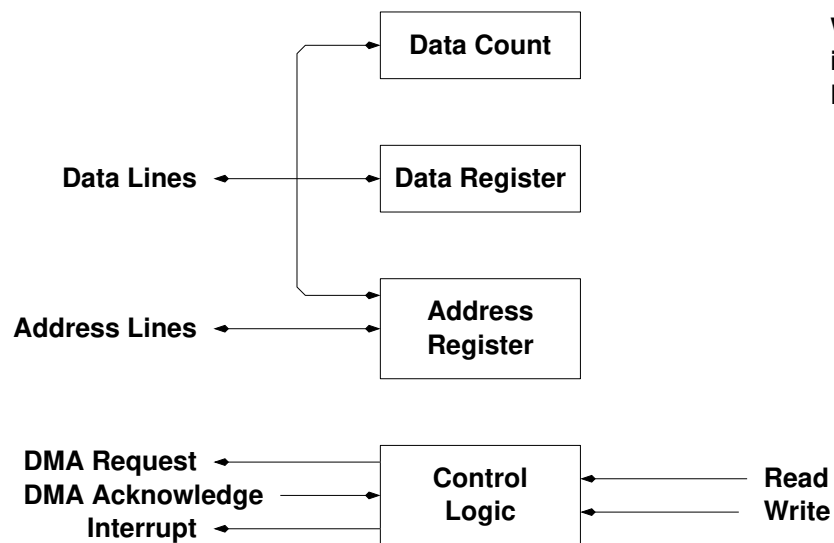


## ... 11.2 - Funções do Sistema de I/O

- \* Como mencionado, durante a evolução dos dispositivos, mais e mais de suas funções deixaram de ser executadas ou não mais receberam a participação do processador;
- ... para todos os módulos descritos desde o passo 4 até o 6, o termo acesso direto à memória é apropriado, posto que, todos envolvem controle direto da memória principal pelo módulo de entrada/saída.
- \* Tipicamente, o Módulo de DMA realiza a transferência de dados de/para a memória através do barramento do sistema, portanto, assume o papel do processador no controle do barramento do sistema;
- ... o barramento é utilizado somente quando o processador não está utilizando, embora possa forçar a suspensão temporária da operação pelo processador.

## ... 11.2 - Funções do Sistema de I/O

\* Em termos gerais, a lógica de um DMA pode ser descrita através do diagrama:

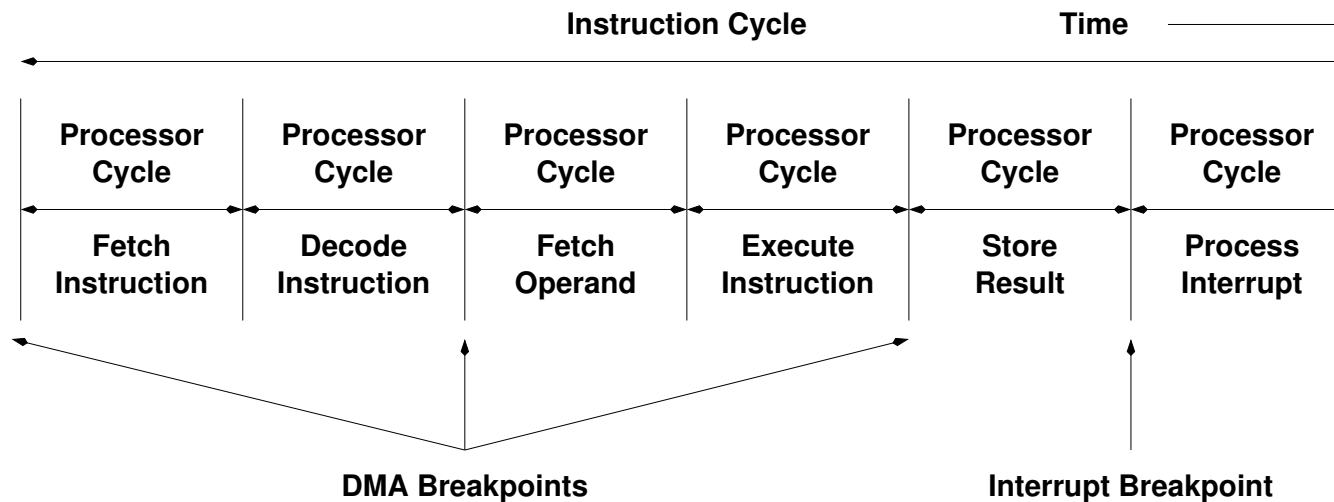


When the processor wishes to read or write a block of data, it issues a command to the DMA module by sending to the DMA module the following information:

- Whether a read or write is requested, using the read or write control line between the processor and the DMA module;
- The address of the I/O device involved, communicated on the data lines;
- The starting location in memory to read from or write to, communicated on the data lines and stored by the DMA module in its address register;
- The number of words to be read or written, again communicated via the data lines and stored in the data count register.

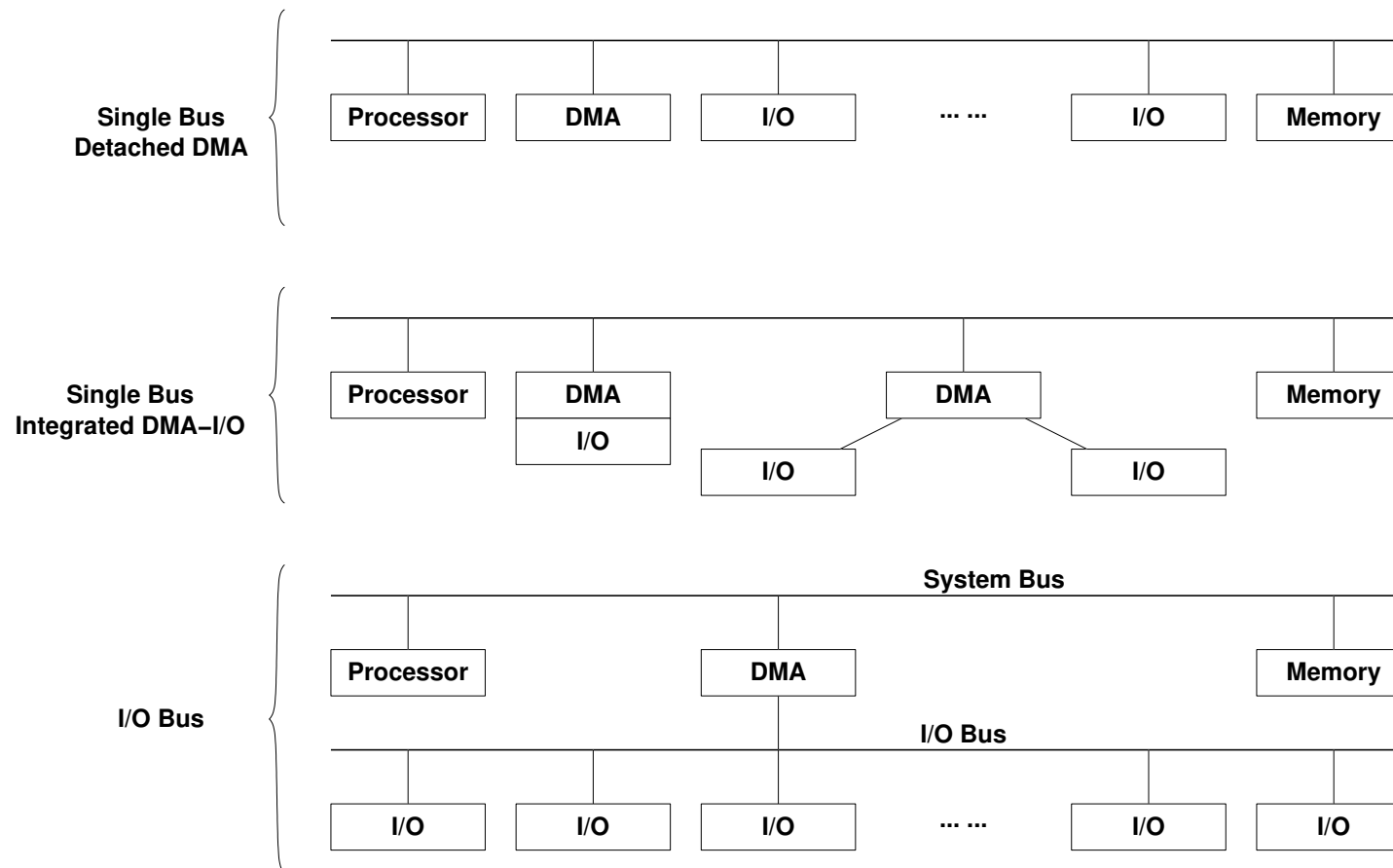
## ... 11.2 - Funções do Sistema de I/O

- \* O diagrama abaixo mostra onde durante o ciclo de instrução o processador pode ser suspenso pelo DMA, de modo que a transferência de dados possa ser feita:



## ... 11.2 - Funções do Sistema de I/O

\* Formas pelas quais o mecanismo de DMA pode ser configurado:



## 11.3 - Aspectos de Projeto do Sistema de I/O

\* No projeto de funcionalidades do sistema de entrada/saída, 02 aspectos devem invariavelmente ser contemplados: **eficiência** e **generalidade**.

❶ **eficiência:** é fundamental no projeto, posto que, operações de entrada/saída constituem o gargalo nos sistemas computacionais;

- ... dentre as várias maneiras de lidar com este problema, destacam-se a multiprogramação e a permutação de processos entre memórias (*swapping*).

❷ **generalidade:** é desejável que se possa tratar todos os dispositivos de maneira uniforme em nome da simplicidade e liberdade no tratamento de erros.

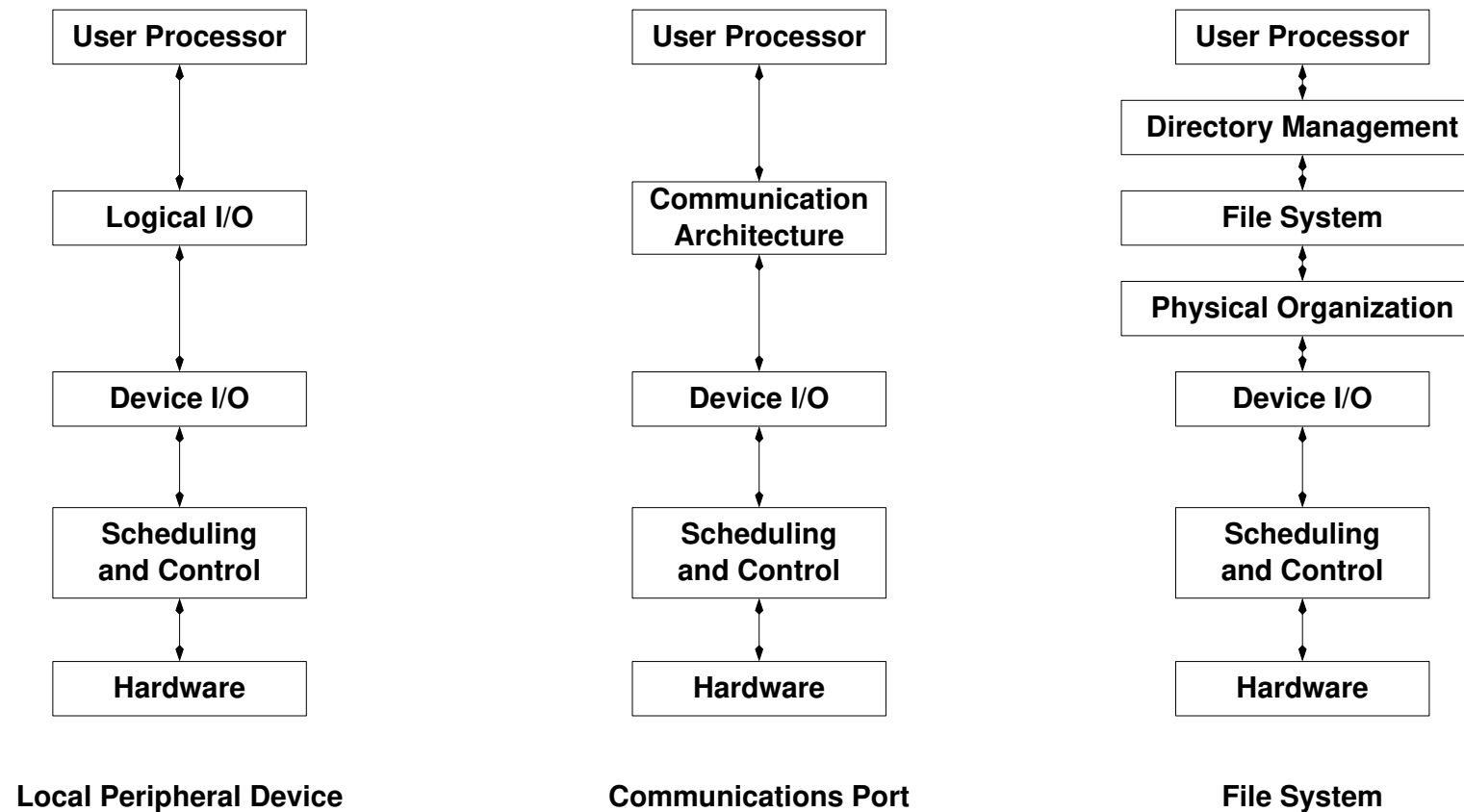
- ... isto se aplica tanto na forma como processos enxergam os dispositivos de entrada/saída quanto na forma como o sistema operacional gerencia os dispositivos de entrada/saída e as operações por eles suportadas.

## ... 11.3 - Aspectos de Projeto do Sistema de I/O

- \* Quanto à estrutura lógica do sistema, a filosofia hierárquica estabelece que as funções do sistema operacional devem ser separadas de acordo com a complexidade, com as características na escala de tempo e com os seus níveis de abstração;
- ... esta abordagem conduz a uma organização do sistema operacional em uma série de níveis, cada qual relacionado a um conjunto de funções do sistema operacional;
- ... de modo geral, as camadas inferiores com constantes de tempo menores, posto que devem interagir diretamente com o *hardware* do sistema computacional;
- ... do outro lado, próximo do usuário, constantes maiores na escala de tempo são tratadas, como p. ex., comandos emitidos pelo usuário.

## ... 11.3 - Aspectos de Projeto do Sistema de I/O

\* A aplicação desta filosofia especialmente nas funcionalidades do sistema de entrada/saída conduz a organização sugerida abaixo:



## ... 11.3 - Aspectos de Projeto do Sistema de I/O

- \* **Logical I/O:** trata o dispositivo como um recurso lógico, ou seja, é responsável pelo gerenciamento das funções de entrada/saída em benefício dos processos do usuário, possibilitando que os dispositivos de entrada/saída sejam operados por simples comandos: *open*, *close*, *read* e *write*.
- \* **Device I/O:** responsável por converter operações e dados (caracteres bufferizados, registros, etc.) requisitados em sequências apropriadas de instruções de entrada/saída, *channel commands* e sinais para o controlador do dispositivo.
- \* **Scheduling and Control:** responsável pelo enfileiramento e escalonamento de operações de entrada/saída, bem como pelo controle destas operações.
- \* **Communication Architecture:** substitui o módulo lógico de entrada/saída, e pode ser constituído de várias camadas (p.ex.: Modelo de Referência OSI).

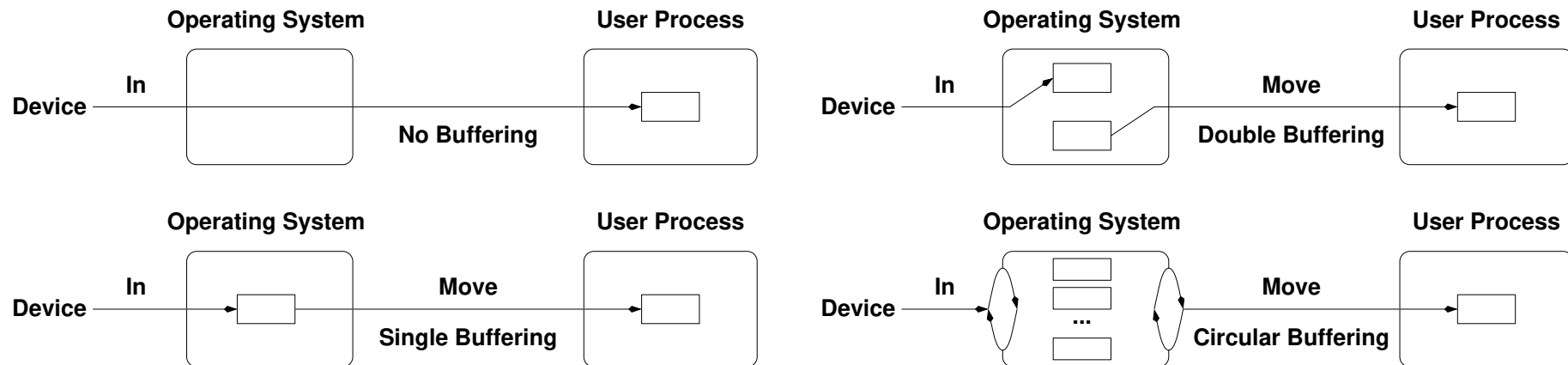


## 11.4 - Bufferização do Sistema de I/O

- \* Suponha que um processo do usuário leia blocos de dados, um por vez, de uma unidade de fita magnética e que cada bloco tenha um comprimento de 100 bytes;
  - ... os dados devem ser lidos para uma área do processo do usuário cuja localização na memória corresponde ao intervalo do endereço 1000 a 1099;
  - ... o forma mais simples seria executar o comando de entrada/saída para a unidade de fita e então esperar que o dado se torne disponível, podendo o processo de espera se dar por espera ocupada, por suspensão do processo ou por uma interrupção.
- \* Esta abordagem apresenta 02 problemas: espera pela operação de entrada/saída se completar e interferência nas decisões de permuta por parte do sistema operacional;
  - ... o que pode trazer como consequência o *deadlock* de todo o sistema.

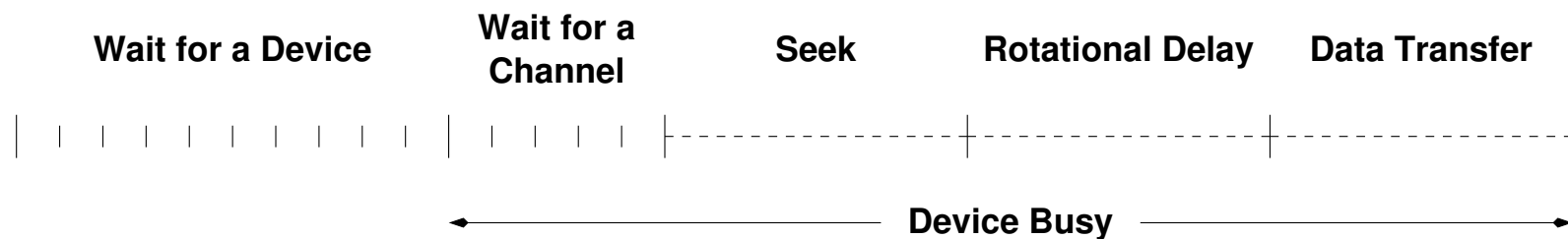
## ... 11.4 - Bufferização do Sistema de I/O

- \* Para evitar tais sobreposições e ineficiências, é às vezes conveniente efetuar transferência de entrada em avanço às requisições sendo processadas e transferência de saída após a requisição ter sido completada — **buffering**.



## 11.5 - Escalonamento de Discos Rotativos

- \* A performance dos sub-sistemas de armazenamento em disco é de vital importância e tem recebido especial atenção no que se refere a pesquisa, posto que, ainda hoje, apresentam taxas de transferência ao menos 4 vezes menos que a memória principal;
- ... operações de entrada/saída em disco dependem do sistema operacional, da natureza do canal de entrada/saída bem como do *hardware* do controlador de disco.



## ... 11.5 - Escalonamento de Discos Rotativos

- \* **seek time:** pode ser aproximado por  $T_s = m * n + s$ , onde  $T_s$  é o tempo estimado de *seek time*;  $n$  é o número de trilhas percorridas;  $m$  é um constante que depende do *drive* do disco e  $s$  é o tempo de partida.
- \* **rotational delay:** embora dependente das características rotacionais do disco, pode ser expresso como a metade do tempo de rotação, ou seja,  $\frac{T_r}{2}$
- \* **transfer time:**  $T = \frac{b}{r * N}$ , onde  $T$  é o tempo de transferência;  $b$  representa o número de bytes a serem transferidos;  $N$  é o número de bytes por trilha e  $r$  é a rotação em revoluções por segundo.
- ... assim, o tempo médio de acesso pode ser expresso por:  $T_a = T_s + \frac{1}{2 * r} + \frac{b}{r * N}$ , onde  $T_s$  é o tempo médio de *seek*.

## ... 11.5 - Escalonamento de Discos Rotativos

- \* Considere um disco típico com *seek time* médio de 20 ms, 3600 rpm, taxa de transferência de 1 MB/s e 512 bytes por setor com 32 setores por trilha;
- ... considere ainda um arquivo de 128 Kbytes (8 trilhas de 32 setores) armazenado na forma mais compacta possível, então o tempo de leitura será de:  $20ms + 8.3ms + 16.7ms + 7 * (8.3ms + 16.7ms) = 0.22s$ ;
- ... se o mesmo arquivo for armazenado aleatoriamente em setores não consecutivos em trilhas não adjacentes (256 setores), o tempo de leitura será de:  $256 * (20ms + 8.3ms + 16.7ms) = 7.37s$ .
- \* Conclui-se que a ordem em que os setores são lidos do disco tem um efeito muito grande na performance do sistema de entrada/saída, logo políticas de escalonamento em disco podem melhorar em muito a performance do sistema de entrada/saída.

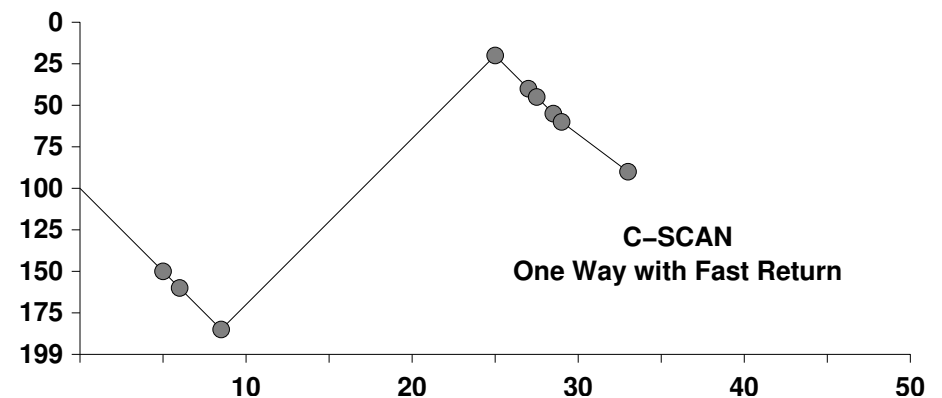
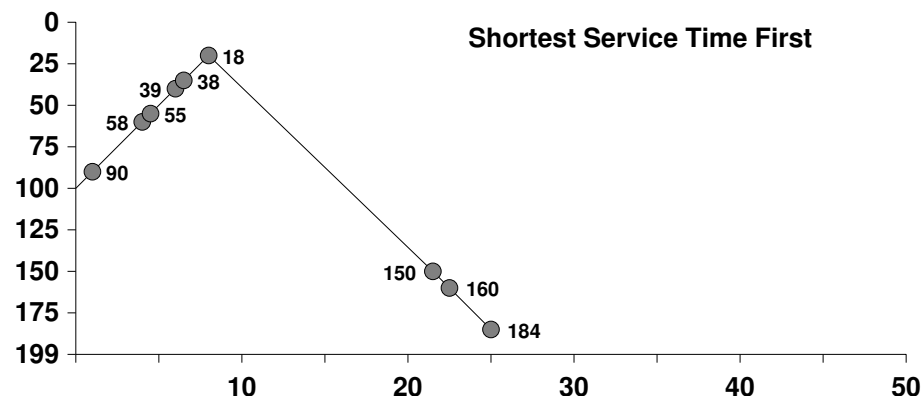
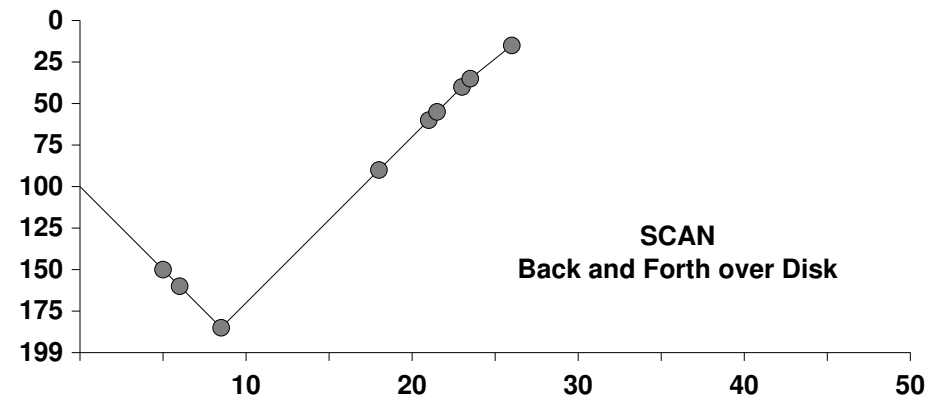
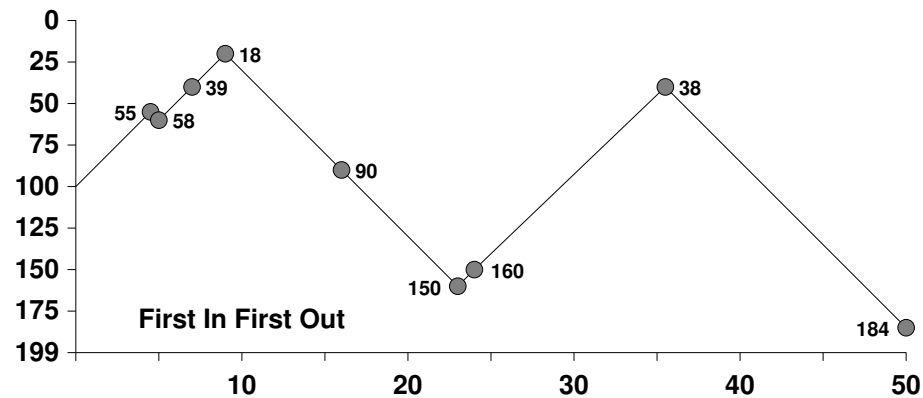
## ... 11.5 - Escalonamento de Discos Rotativos

\* Considere as seguintes Políticas de Escalonamento em Disco:

First In First Out (FIFO) (starting at track 100)		Shortest Service Time First (SSTF) (starting at track 100)		SCAN (starting at track 100, in the direction of increasing number)		C-SCAN (starting at track 100, in the direction of increasing number)	
Next Track Accessed	Number of Tracks Traversed	Next Track Accessed	Number of Tracks Traversed	Next Track Accessed	Number of Tracks Traversed	Next Track Accessed	Number of Tracks Traversed
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average Seek Length	55.3	Average Seek Length	27.5	Average Seek Length	27.8	Average Seek Length	35.8

## ... 11.5 - Escalonamento de Discos Rotativos

\* Movimento do braço do disco para as políticas anteriormente apresentadas:



## ... 11.5 - Escalonamento de Discos Rotativos

- \* **SCAN:** alternativa simples que previne o *starvation* das requisições, posto que o braço se desloca em uma única direção satisfazendo todas as requisições na rota;
  - ... não é difícil verificar que esta política favorece requisições para trilhas em cilindros mais internos ou mais externos, bem como favorece requisições que foram encaminhadas mais recentemente.
- \* **C-SCAN:** restringe a busca para uma única direção, assim, quando a última trilha for visitada em uma direção o braço retorna para a extremidades oposta do disco e recomeça a busca reduzindo o atraso máximo imposto por requisições recentes;
  - ... se o intervalo de tempo entre requisições para cilindros opostos e nas extremidades do disco é  $t$ , o intervalo de tempo do C-SCAN é da ordem de  $t + s_{max}$  onde  $s_{max}$  é o tempo máximo de *seek time*.



## 11.6 - Redundant Array of Independent Disks

- \* Com múltiplos discos, requisições de I/O podem ser tratadas em paralelo desde que os dados objetos da requisição estejam em discos diferentes;
- ... da mesma forma, uma requisição de I/O pode ser executada em paralelo se o bloco de dados objeto da requisição for distribuído entre os discos.
- \* Com múltiplos discos há várias maneiras de organizar os dados de modo que a redundância de informação seja convertida em aumento de confiabilidade;
- ... felizmente, um padrão foi estabelecido e aceito entre os fabricantes no tocante à organização de informação (p.ex., base de dados) em múltiplos discos – RAID (Redundancy Array of Independent Disks) em 06 diferentes níveis.

## ... 11.6 - Redundant Array of Independent Disks

- \* Estes 06 níveis não tem implicação hierárquica entre si, mas espelham diferentes projetos arquiteturais que compartilham 03 características comuns:
  - ... RAID consiste de um conjunto de *drivers* de discos físicos vistos pelo sistema operacional como um único *driver* lógico de disco;
  - ... dados são distribuídos entre os *drivers* físicos de uma lista de *drivers* (p.ex., *array* de *drivers*);
  - ... redundância na capacidade do disco é utilizada para armazenar informações de paridade, para possibilitar a recuperação de dados no caso de falhas no disco.

## ... 11.6 - Redundant Array of Independent Disks

- \* A proposta do RAID foi a de diminuir a distância entre a velocidade relativa dos processadores quando comparada a dos *drivers* de discos eletromecânicos;
- ... a estratégia consiste em substituir discos de alta capacidade por múltiplos discos de menor capacidade, bem como distribuir os dados de forma que possibilite o acesso simultâneo a partir de múltiplos *drivers*;
- \* Se por um lado a operação em paralelo de múltiplas cabeças de leitura/gravação e atuadores oferecem altas taxas de transferências, por outro lado aumenta a probabilidade de falhas;
- ... para compensar a perda de confiabilidade, faz-se uso de informações de paridade para possibilitar a recuperação de dados perdidos por falhas em discos.

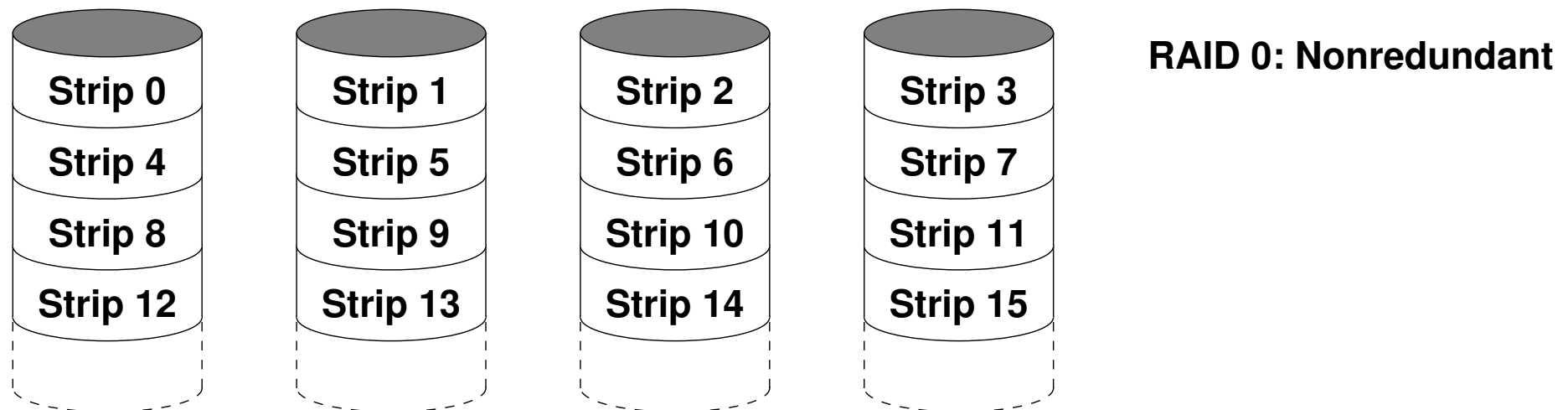
## ... 11.6 - Redundant Array of Independent Disks

\* Tabela abaixo resume os 06 níveis RAID, entretanto os níveis 2 e 4 não são comercialmente oferecidos e constituem padrões pouco aceitos pela indústria.

Category	Level	Description	I/O Request Rate (Read/Write)	Data Transfer Rate (Read/Write)	Typical Application
Striping	0	Nonredundant	Large strips: Excellent	Small strips: Excellent	Application requiring high performance for noncritical data.
Mirroring	1	Mirrored	Good/Fair	Fair/Fair	System drives: critical files.
Parallel Access	2	Redundant via Hamming Code	Poor	Excellent	Large I/O request size applications such as imaging, CAD.
	3	Bit-Interleaved Parity	Poor	Excellent	
Independent Access	4	Block-Interleaved Parity	Excellent/Fair	Fair/Poor	High request rate, read-intensive, data lookup.
	5	Block-Interleaved Distributed Parity	Excellent/Fair	Fair/Poor	

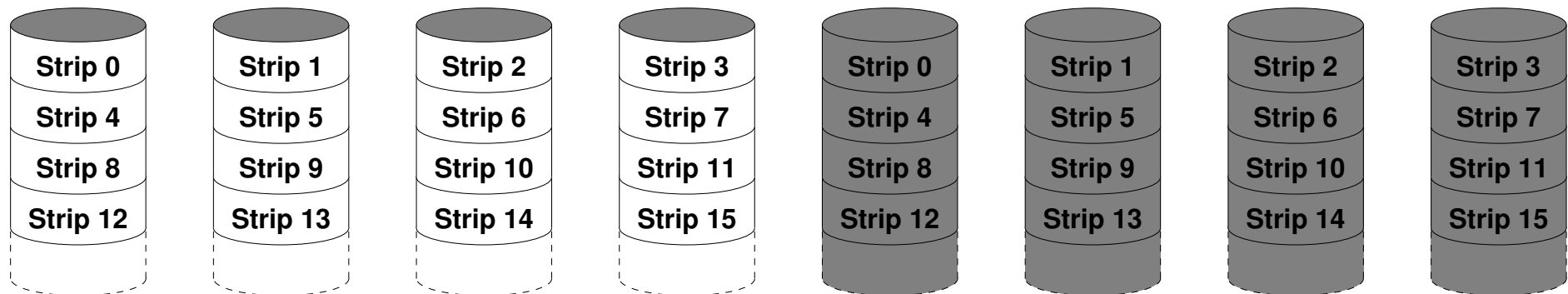
## ... 11.6 - Redundant Array of Independent Disks

- \* **RAID 0:** dados do sistema e do usuário são distribuídos nos discos, entretanto não utiliza redundância de informação para melhorar a performance.
- ... admitindo que haja requisições pendentes para blocos de dados distintos, há uma grande chance que os blocos requisitados estejam em diferentes discos — assim, as requisições podem ser disparadas em paralelo.



## ... 11.6 - Redundant Array of Independent Disks

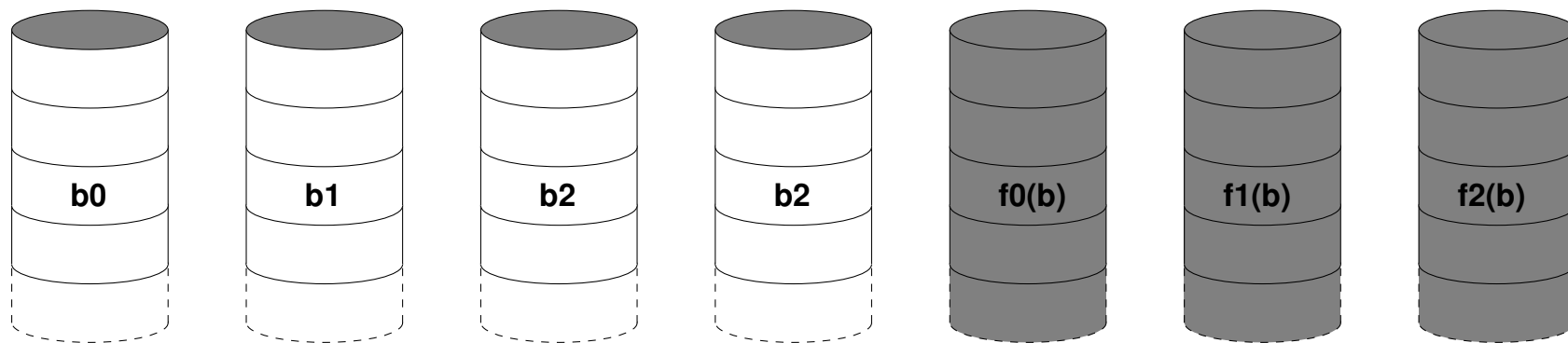
- \* **RAID 1:** contempla redundância na sua forma mais simples, replica integralmente os dados exigindo, assim, no mínimo o dobro da capacidade desejada para armazenamento — custo alto é a sua principal desvantagem.
- ... em situações onde o ambiente orientado a transações, a performance do RAID 1 pode dobrar em relação ao do RAID 0, entretanto, se uma fração substancial das requisições for de escrita então o ganho de performance será insignificante.



**RAID 1: Mirrored**

## ... 11.6 - Redundant Array of Independent Disks

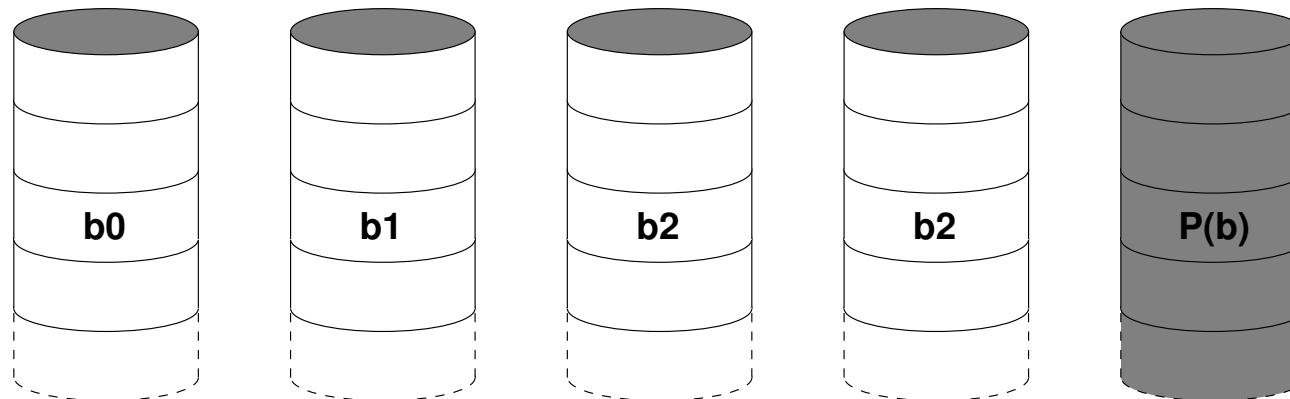
- \* **RAID 2:** utiliza técnica de acesso paralela, assim todos os discos participam da execução (normalmente síncrona) de uma requisição de I/O.
- ... no RAID 2 e 3 as *strips* são pequenas, frequentemente um único byte ou palavra e utiliza o Código de Hamming para correção e detecção de erro (Hamming Code for Error Checking and Correction - ECC).



**RAID 2: Redundancy through Hamming Code**

## ... 11.6 - Redundant Array of Independent Disks

- \* **RAID 3:** emprega acesso paralelo aos dados e está organizado de forma similar ao RAID 2, exceto que requer apenas um disco para armazenar informações redundantes não importando quão grande seja o *array* de discos.
- ... o bit de paridade para o  $i$ -ésimo bit é calculado pela equação  $X_4(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$ , assim, o conteúdo de qualquer faixa de dados em qualquer um dos discos pode ser regenerada a partir das informações dos demais.

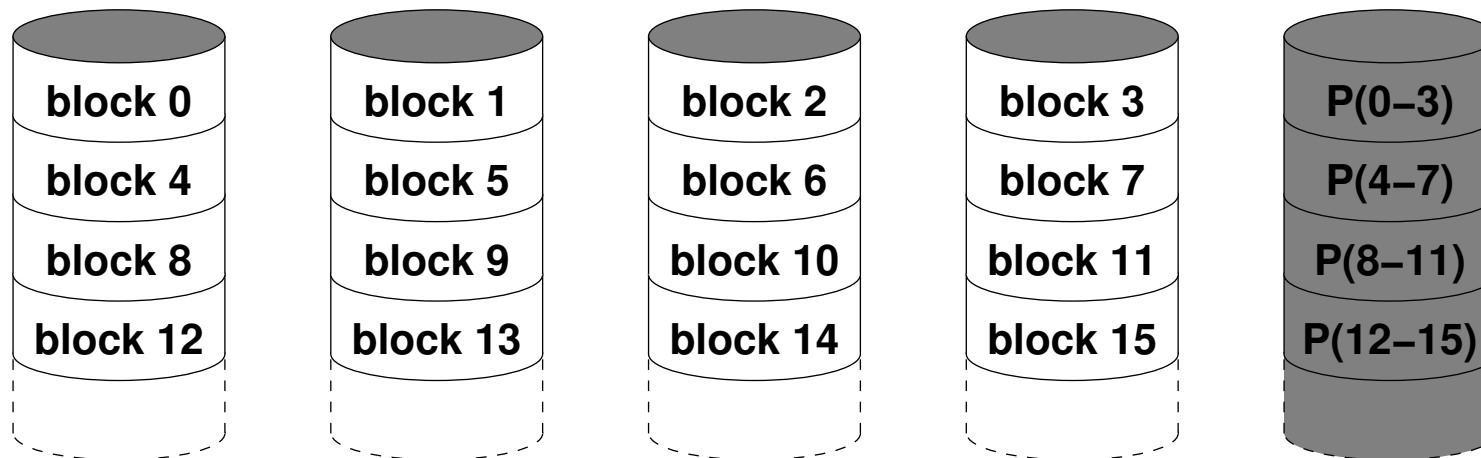


**RAID 3: Bit Interleaved Parity**



## ... 11.6 - Redundant Array of Independent Disks

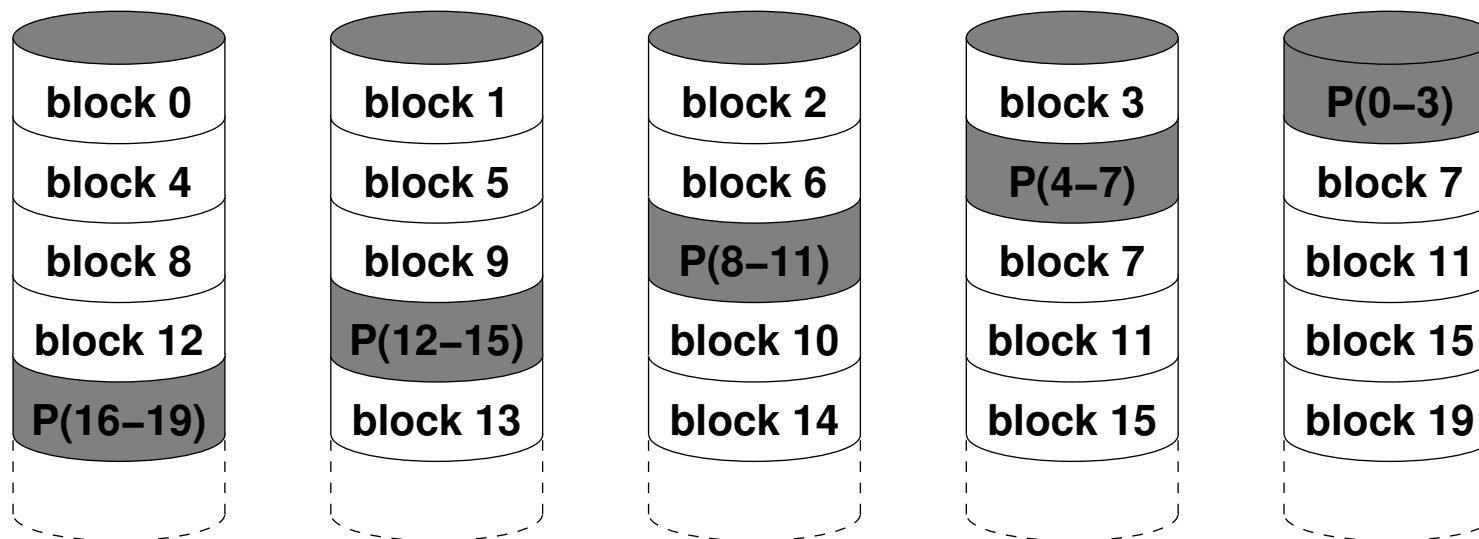
- \* **RAID 4:** opera com acessos independentes um do outro, assim, cada disco opera independentemente possibilitando que a separação de requisições se dê em paralelo.
- ... após a modificação de alguma faixa de dados (p.ex.,  $X_1$ ) e considerando que para cada bit  $i$  a seguinte relação é válida:  $X_4(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$  então  $X'_4(i) = X_4(i) \oplus X_1(i) \oplus X'_1(i)$



**RAID 4: Block Level Parity**

## ... 11.6 - Redundant Array of Independent Disks

- \* **RAID 5:** organizado de forma similar ao RAID 4, exceto que distribui os bits de paridade entre todos os discos — uma alocação típica é a de *round-robin*.
- ... para um *array* de  $n$  discos, a faixa de paridade é armazenada em um disco que não aquele que contenha as  $n$  faixas (*strips*) e, assim, o padrão se repete.



**RAID 5: Block Level Distributed Parity**

## 11.7 - Memória Cache de Discos Rotativos

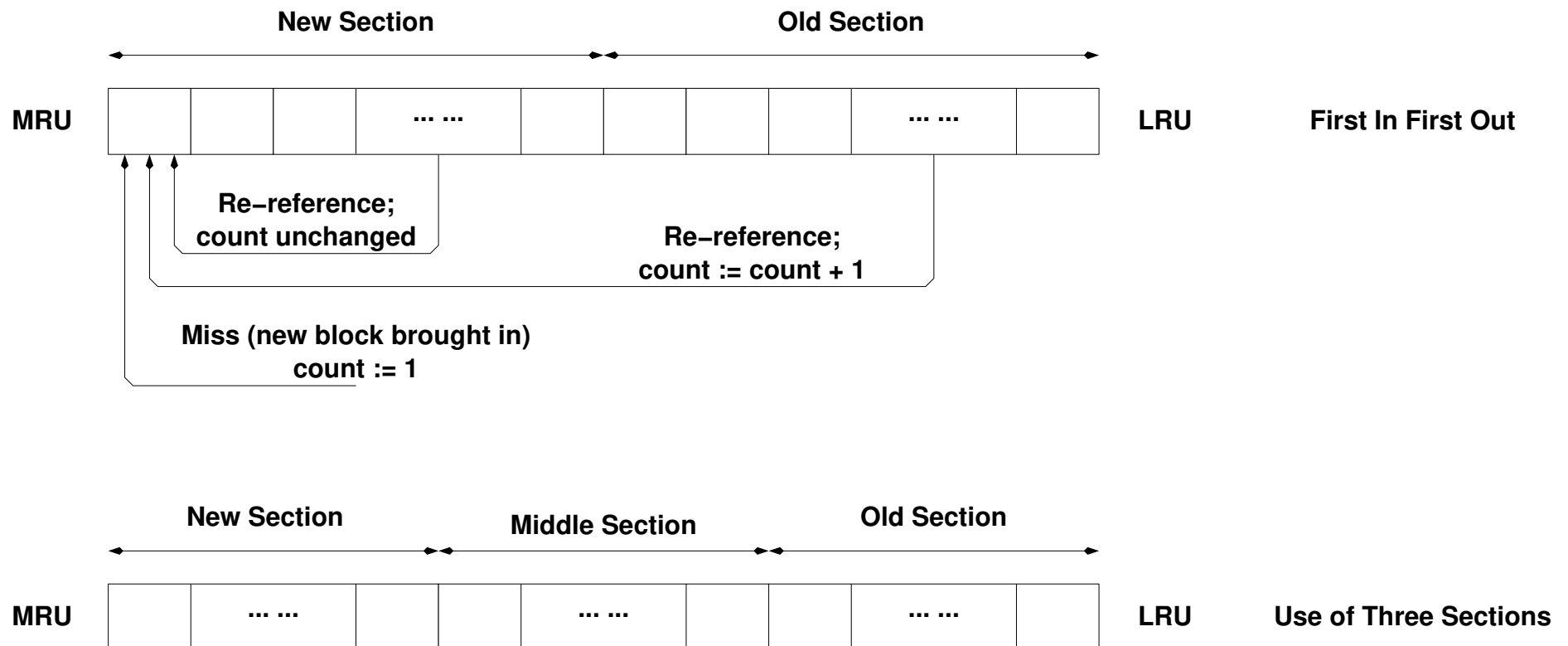
- \* **cache de disco:** termo normalmente aplicado a um *buffer* da memória principal reservado para setores do disco – contém cópia de alguns setores do disco.
- ... explora o mesmo princípio da localidade aplicado ao memória *cache*, reduzindo o tempo médio de acesso a memória principal.
- \* Por conta do princípio da localidade (para um bloco presente na *cache* de disco há a probabilidade de referências futuras ao mesmo bloco), as seguintes **considerações de projeto** se fazem necessárias:
  - ❶ quando uma requisição de I/O é atendida na *cache* do disco, o dado na *cache* deve ser disponibilizado para o processo que o requisitou (p.ex., transferência);
  - ❷ quando um novo setor é carregado na *cache* de disco, um dos blocos corrente será substituído, ou seja, vale aqui os mesmos algoritmos de substituição de páginas.

## ... 11.7 - Memória Cache de Discos Rotativos

- \* Uma possibilidade é mover o bloco que acabou de ser referenciado para o topo da pilha, assim, toda vez que um bloco for lido da memória secundária, remove-se o bloco da início da pilha e empilha-se o bloco que acabou ora referenciado.
- ... naturalmente não é necessário mover estes blocos pela memória, basta associar ponteiros de pilha à *cache* de disco.
- \* Uma outra possibilidade é o LFU (*Least Recently Used*), entretanto apresenta um problema quando blocos são referenciados infrequentemente, mas quando o são o seu contador atinge valores altos em decorrência de repetidas referências;
- ... neste cenário, o contador pode não refletir que o bloco será novamente e rapidamente referenciado e, assim, o efeito da localidade causa ao algoritmo degradação na escolhas dos blocos a serem substituídos.

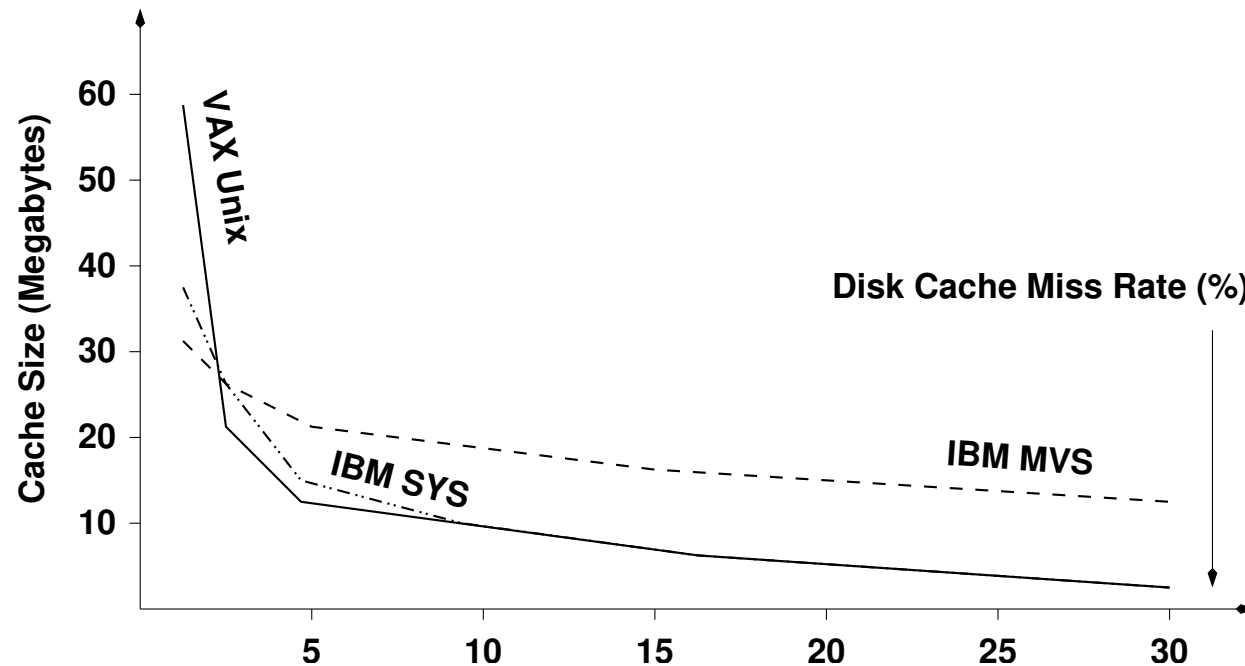
## ... 11.7 - Memória Cache de Discos Rotativos

- \* Para contornar este problema, utiliza-se a Substituição baseada em Frequência - porção do topo da pilha é mantida aparte como uma nova seção.



## ... 11.7 - Memória Cache de Discos Rotativos

- \* **Considerações de Performance:** converge para o questão de quando uma dada taxa de erro é alcançada, o que por sua vez, depende do comportamento das referências no disco, do algoritmo de substituição e de outros fatores de projeto;
- ... outro aspecto é que a taxa de erros é função do tamanho da *cache* de disco como pode ser comprovado pelos diagramas abaixo.



## 11.8 - Sistema de I/O no UNIX SVR4

\*

## ... 11.8 - Sistema de I/O no UNIX SVR4

\*



## 11.9 - Sistema de I/O no Windows

\*

## ... 11.9 - Sistema de I/O no Windows

\*