

# Estrutura de Dados II

---

## Aula 19 – Ordenação Externa

Giovanni Comarela

Departamento de Informática

Universidade Federal do Espírito Santo

# Processamento de dados em memória externa

---

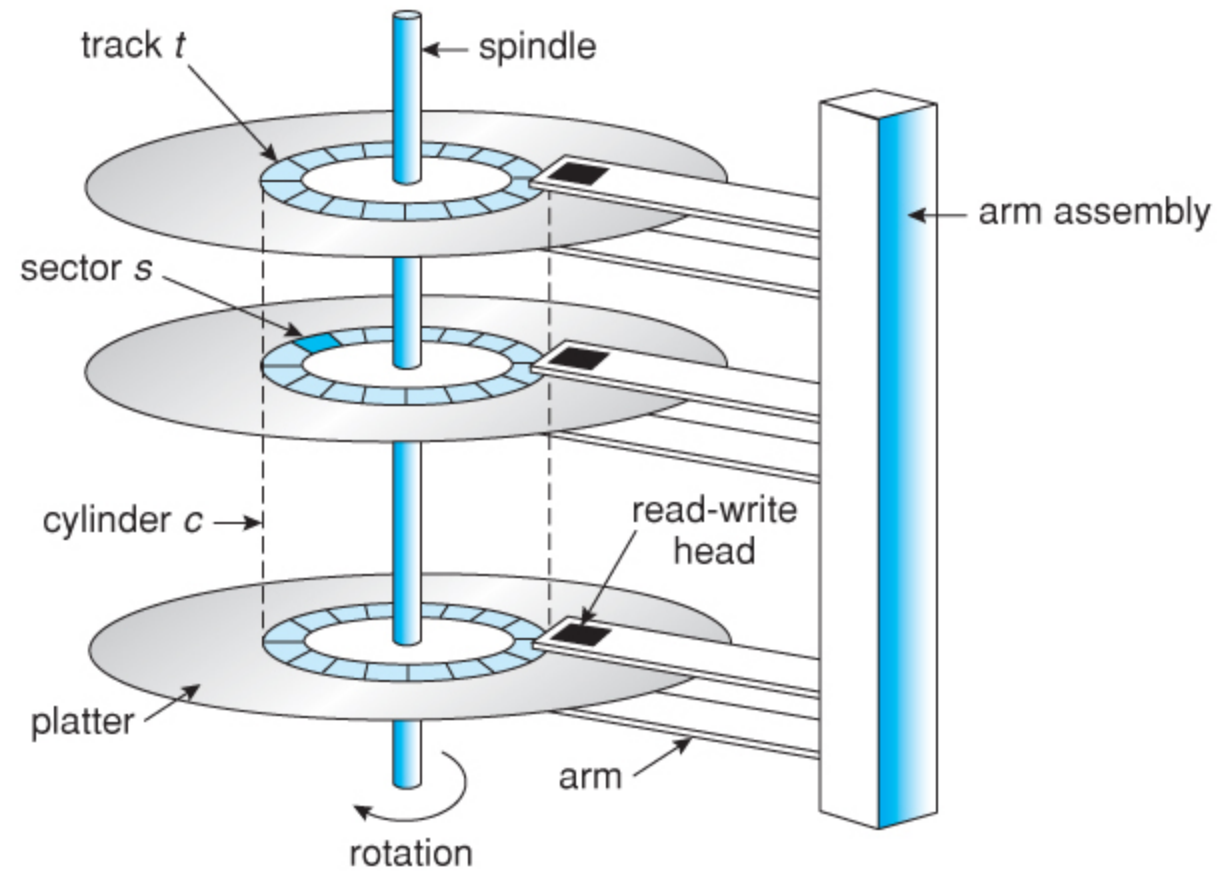
- Muitas vezes, informações não podem ser armazenadas e processadas inteiramente na memória principal dos computadores.
- **Ex:** bancos de dados, imagens de satélite em alta resolução, vídeos, etc.

# Processamento de dados em memória externa

---

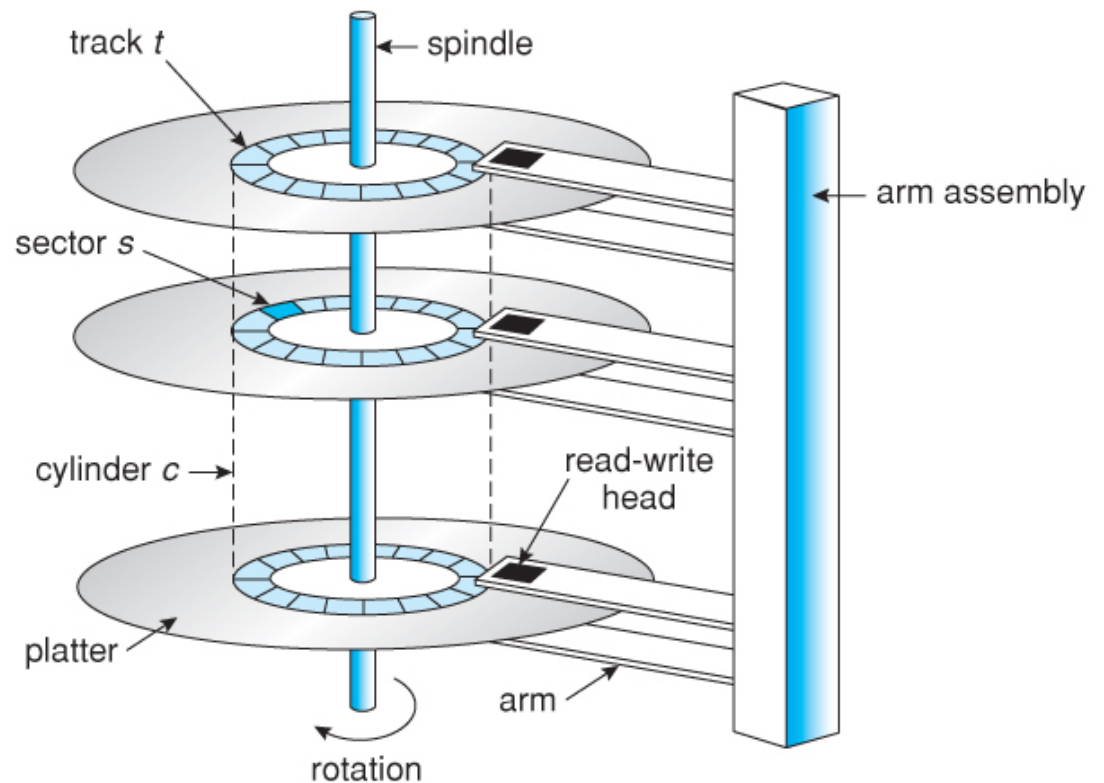
- Tais dados precisam ser processados na **memória externa** dos computadores.
- Exemplos de memória externa: *discos, fitas, DVDs, SSDs*, etc.
- Problemas com a memória externa.
  - Alto custo de acesso aos dados.
  - Vários custos diferentes podem ser associados ao acesso (*e.g.*, custo de ler dados, custo de gravar, custo de posicionar a cabeça de gravação, etc.)
  - Os custos dependem da tecnologia: Disco vs. SSD, Fita vs. Disco.

# Anatomia de um disco rígido



# Quanto tempo gasta para ler/escrever no disco?

- Depende primariamente de duas quantidades:
  - Tempo de *Seek*
    - Rotação completa
    - Posicionamento do braço
  - Tempo de transferência



# Exemplos

- Tempo típico de *seek* em discos: **8ms**
- Suponha que um disco tenha taxa de transferência interna de **100MB/s**.
- Vamos supor que o tamanho de bloco seja **4KB**.
- Quanto tempo gastaríamos para fazer um *seek* e ler um inteiro (com 4 bytes) do disco?
- Quanto tempo gastaríamos para fazer um *seek* e ler 10000 inteiros do disco?

# Conta de Padaria 1

- 
- Tempo de *seek*: **8ms**
  - Taxa de transferência: **100MB/s**
  - Tamanho de bloco: **4KB**
  - Tempo para ler 4 bytes do disco: ?

# Conta de Padaria 1

- Tempo de *seek*: **8ms**
- Taxa de transferência: **100MB/s**
- Tamanho de bloco: **4KB**
- Tempo para ler 4 bytes do disco: **8.04ms**

*Para ler um bloco: 0.04ms*

*Tempo total: 8ms + 0.04ms*



# Conta de Padaria 2

- 
- Tempo de *seek*: **8ms**
  - Taxa de transferência: **100MB/s**
  - Tamanho de bloco: **4KB**
  - Tempo para ler 40 mil bytes do disco: ?

# Conta de Padaria 2

- Tempo de *seek*: **8ms**
- Taxa de transferência: **100MB/s**
- Tamanho de bloco: **4KB**
- Tempo para ler 40 mil bytes do disco: **8.4ms**

*40 mil bytes usam aproximadamente 10 blocos*

*Para ler 10 blocos: 0.4ms*

*Tempo Total: 8ms + 0.4ms*

# Processamento de dados em memória externa

---

- Note que o tempo de *seek* é **dominante** em ambos os casos
- Assim, o acesso **aleatório** ao disco é muito ineficiente
- Imagine como seria o acesso a uma fita... Quanto tempo custaria fazer um acesso na última posição da fita e, a seguir, acessar o primeiro elemento gravado nela?
- Na memória principal, esse problema é menos grave
  - Lembrando RAM = *Random Access Memory*

# Processamento de dados em memória externa

- Veja uma tabela com tempos típicos (e aproximados) de algumas operações em computadores

<b>execute typical instruction</b>	1 nanosec
fetch from L1 cache memory	0.5 nanosec
fetch from L2 cache memory	7 nanosec
<b>fetch from main memory</b>	100 nanosec
send 2K bytes over 1Gbps network	20,000 nanosec
read 1MB sequentially from memory	250,000 nanosec
<b>fetch from new disk location (seek)</b>	8,000,000 nanosec
read 1MB sequentially from disk	20,000,000 nanosec
send packet US to Europe and back	150,000,000 nanosec

# Processamento de dados em memória externa

---

- Devido a essa ENORME diferença entre o custo de acesso ao disco e a execução de operações na CPU, normalmente os algoritmos projetados para trabalhar com dados na memória externa são avaliados com base no acesso a disco
- Assim, normalmente é preferível utilizar um algoritmo com complexidade de tempo de CPU mais alta mas que faça menos acesso a disco a usar um algoritmo muito eficiente em termos de CPU mas que realiza muitos acessos ao disco.
- Métrica de interesse: **acessos ao disco**

# Ordenação em memória externa

- 
- A seguir, vamos apresentar um exemplo de algoritmo para ordenar dados em memória externa.

# Ordenação em memória externa

- O método de ordenação externa mais conhecido é o método de ordenação por **intercalação**.
  - Hoje veremos **Intercalação Balanceada**
- Assim como no algoritmo *Merge Sort*, a intercalação é utilizada como operação auxiliar do algoritmo de ordenação.

# Modelo de Abstração

- 
- Queremos ordenar  **$N$**  registros
  - Apenas  **$M$**  cabem na memória principal ( **$M < N$** )
  - O Computador possui  **$2P$**  dispositivos de armazenamento
  - Os dados a serem ordenados estão no dispositivo  **$0$**



# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: INTERCALACAOBALANCEADA

Disp. 1:

Disp. 2:

Disp. 3:

Disp. 4:

Disp. 5:

# Ordenação em memória externa

## Intercalação Balanceada

---

- 1) É realizada uma primeira passada sobre o arquivo, quebrando-o em blocos do tamanho  **$M$** . Cada bloco é ordenado com um método para memória interna
  - Os blocos são salvos nos dispositivos  **$P, P+1, \dots, 2P-1$**

# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: INTERCALACAOBALANCEADA

Disp. 1:

Disp. 2:

Disp. 3:

Disp. 4:

Disp. 5:

# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: INTERCALCAOBALANCEADA

Disp. 1:

Disp. 2:

Disp. 3: INT ACO ADE

Disp. 4: CER ABL A

Disp. 5: AAL ACN

# Ordenação em memória externa

## Intercalação Balanceada

---

- 2) Os  $P$  primeiros blocos ordenados são intercalados
- Os resultados são blocos maiores, distribuídos nos dispositivos  $0, 1, \dots, P-1$

# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0:

Disp. 1:

Disp. 2:

Disp. 3: 

INT	ACO	ADE
-----	-----	-----

Disp. 4: 

CER	ABL	A
-----	-----	---

Disp. 5: 

AAL	ACN	
-----	-----	--

# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: **AACEILNRT**

Disp. 1: **AAABCCLNO**

Disp. 2: **AADE**

INT	ACO	ADE
CER	ABL	A
AAL	ACN	

# Ordenação em memória externa

## Intercalação Balanceada

---

- 3) Os  $P$  primeiros blocos ordenados são intercalados
- Os resultados são blocos maiores, distribuídos nos dispositivos  $P, P+1, \dots, 2P-1$



# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: **AACEILNRT**

Disp. 1: **AAABCCLNO**

Disp. 2: **AADE**

Disp. 3:

Disp. 4:

Disp. 5:

# Modelo de Abstração - Exemplo

- $N = 22, M = 3, P = 3$

Disp. 0: **AACEILNRT**

Disp. 1: **AAABCCLNO**

Disp. 2: **AADE**

Disp. 3: **AAAAAAAAABCCDEEILLNNORT**

Disp. 4:

Disp. 5:

# Complexidade

- No nosso modelo de abstração, a intercalação balanceada faz  $1 + \lceil \log_P(N/M) \rceil$  passadas pelos dados
- Isso é muito? Suponha
  - $N = 1$  bilhão
  - $M = 1$  milhão
  - $P = 3$
  - Precisaremos de 8 passadas pelos dados

# Complexidade – “Prova”

- 
- Após a passada **1**, teremos  $N/M$  blocos de tamanho **M**
  - Após a passada **2**, teremos  $N/PM$  blocos de tamanho **PM**
  - Após a passada **3**, teremos  $N/P^2M$  blocos de tamanho  **$P^2M$**
  - ...
  - Após a passada **k**, teremos  $N/P^{k-1}M$  blocos de tamanho  **$N/P^{k-1}M$**

# Complexidade – “Prova”

- 
- Após a passada **1**, teremos  **$N/M$**  blocos de tamanho  **$M$**
  - Após a passada **2**, teremos  **$N/PM$**  blocos de tamanho  **$PM$**
  - Após a passada **3**, teremos  **$N/P^2M$**  blocos de tamanho  **$P^2M$**
  - ...
  - Após a passada  **$k$** , teremos  **$N/P^{k-1}M$**  blocos de tamanho  **$P^{k-1}M$**
  - Queremos  **$k$** , tal que  **$N = P^{k-1}M$**
  - Resolvendo para  **$k$** , teremos  **$k = 1 + \log_P(N/M)$**

# Muito além disso

---



UFES  
Informática

