

ACH2024

Organização de arquivos: alocação sequencial (parte 1)

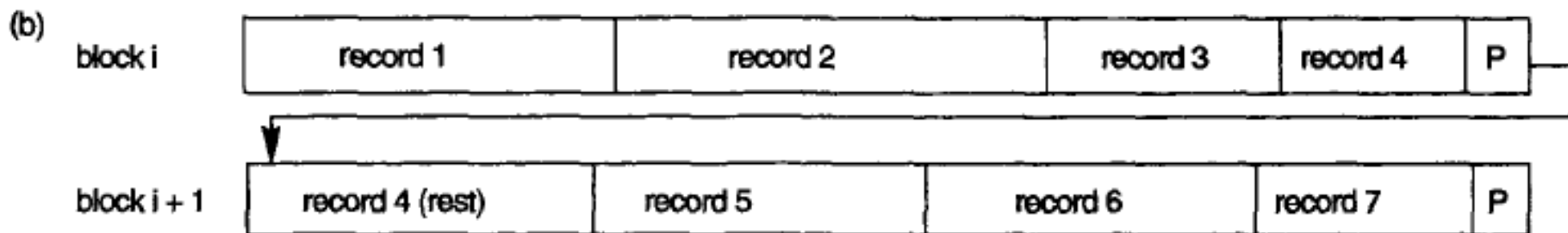
Prof Helton Hideraldo Bís caro

Organização de arquivos na memória secundária

- É comum considerar que em cada bloco pode haver vários registros
- Se R (tamanho fixo do registro, para simplificar) e B (tamanho do bloco) e $R \leq B$:
fator de blocagem $fb = \text{floor}(B/R)$
= número de registros inteiros que cabem em um bloco

Organização de arquivos na memória secundária

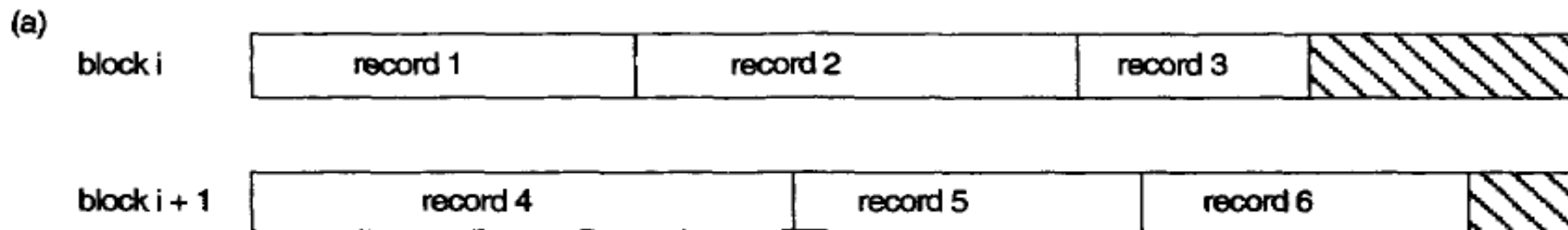
- **Organização *espalhada*:** os blocos são totalmente preenchidos; se um registro não cabe inteiramente na parte vazia do bloco, coloca o que couber e um ponteiro para o próximo bloco



(ELSMARI & NATATHE)

Organização de arquivos na memória secundária

- **Organização não *espalhada*:** registros não podem ser divididos. Cada bloco pode conter até fb registros.



(ELSMARI & NATATHE)

Organização de arquivos na memória secundária

- **Organização não *espalhada*:** registros não podem ser divididos. Cada bloco pode conter até fb registros.
 - Se os registros tiverem tamanho fixo = R , blocos de tamanho B e taxa de blocagem = fb :
Perda de espaço em cada bloco:

Organização de arquivos na memória secundária

- **Organização não *espalhada*:** registros não podem ser divididos. Cada bloco pode conter até fb registros.
 - Se os registros tiverem tamanho fixo = R , blocos de tamanho B e taxa de blocagem = fb :
Perda de espaço em cada bloco: $B - (fb * R)$

Alocação de blocos

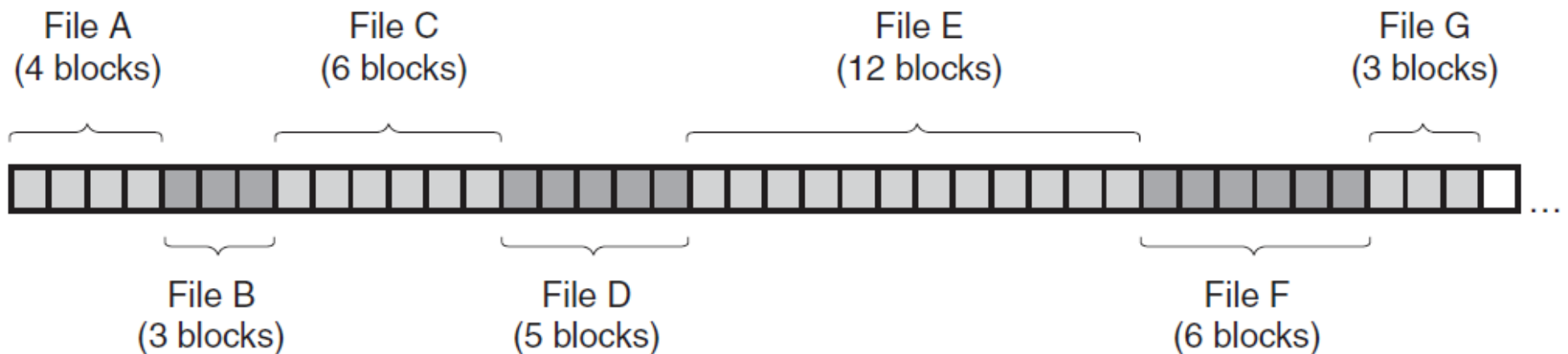
- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem
- Estratégias de alocação de blocos no disco devem considerar esse fato
- Vamos estudar várias estratégias até o fim do semestre
- Para cada estratégia analisaremos complexidade de leitura sequencial, leitura aleatória (busca), inserção e remoção de registros
 - Complexidade em termos de número de ...

Alocação de blocos

- Leitura, escrita, buscas, etc., são realizadas por blocos.
- Os arquivos não são estáticos, eles crescem e diminuem
- Estratégias de alocação de blocos no disco devem considerar esse fato
- Vamos estudar várias estratégias até o fim do semestre
- Para cada estratégia analisaremos complexidade de leitura sequencial, leitura aleatória (busca), inserção e remoção de registros
 - Complexidade em termos de número de seeks (estimado no pior caso pelo número de blocos a serem lidos)

Alocação sequencial

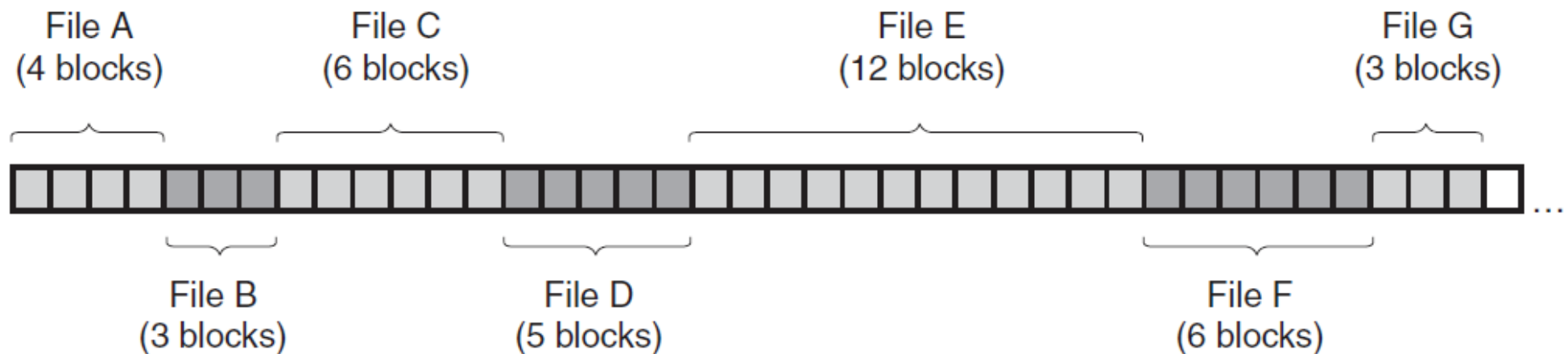
- Blocos alocados sequencialmente no disco (pelos cilindros)



- Vantagens e desvantagens?

Alocação sequencial

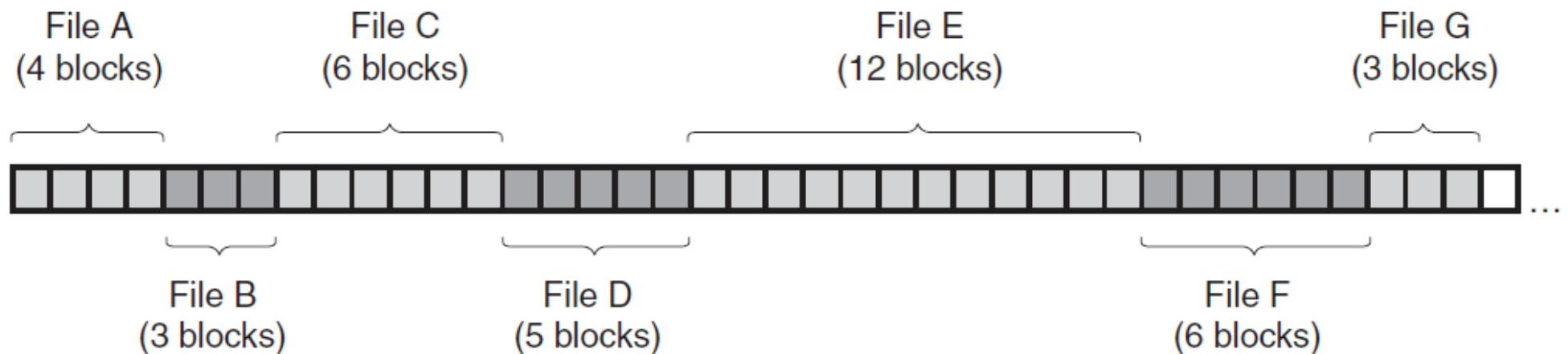
- Blocos alocados sequencialmente no disco (pelos cilindros)



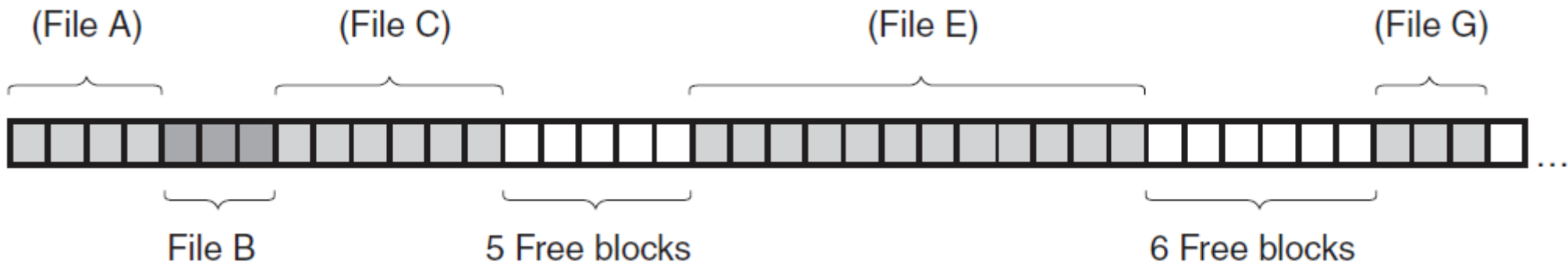
- Vantagens e desvantagens?
 - Leitura fácil (leitura sequencial é ótima, e na leitura aleatória depende da facilidade de localização do deslocamento do registro dentro do arquivo)
 - Expansão complicada: se não houver espaço disponível até o próximo arquivo tem que ser removido para outro local
 - Fragmentação externa (buracos entre os arquivos): maior ou menor dependendo da política de alocação

Alocação sequencial

- Blocos alocados sequencialmente no disco (pelos cilindros)



- Após algumas remoções



Alocação sequencial

Fragmentação externa:

- Com o tempo (após alocações/desalocações sucessivas), o disco pode ficar fragmentado, isto é, com vários trechos disponíveis intercortados por trechos utilizados

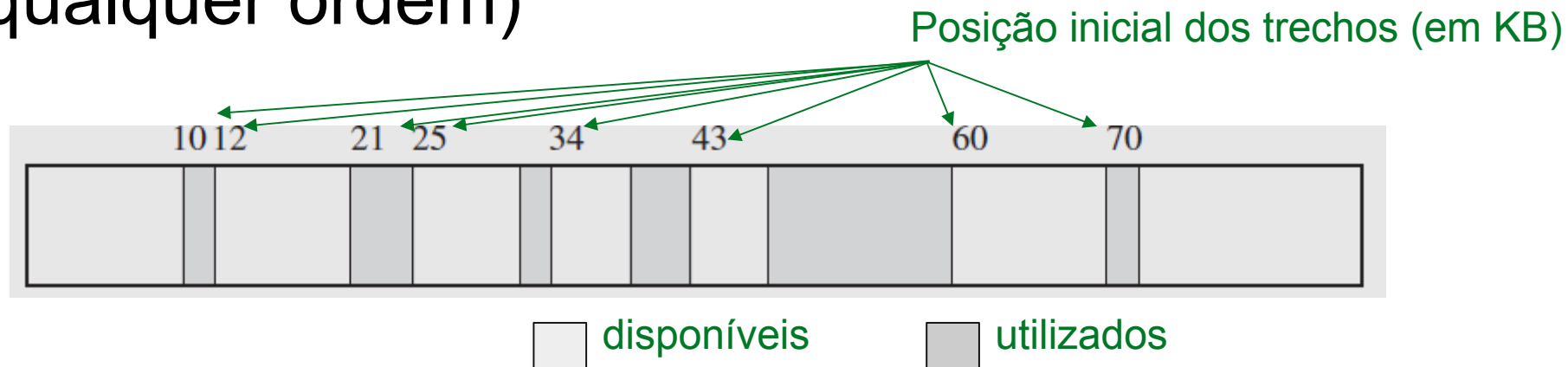


Allocation request



Alocação sequencial - Métodos de ajuste sequencial

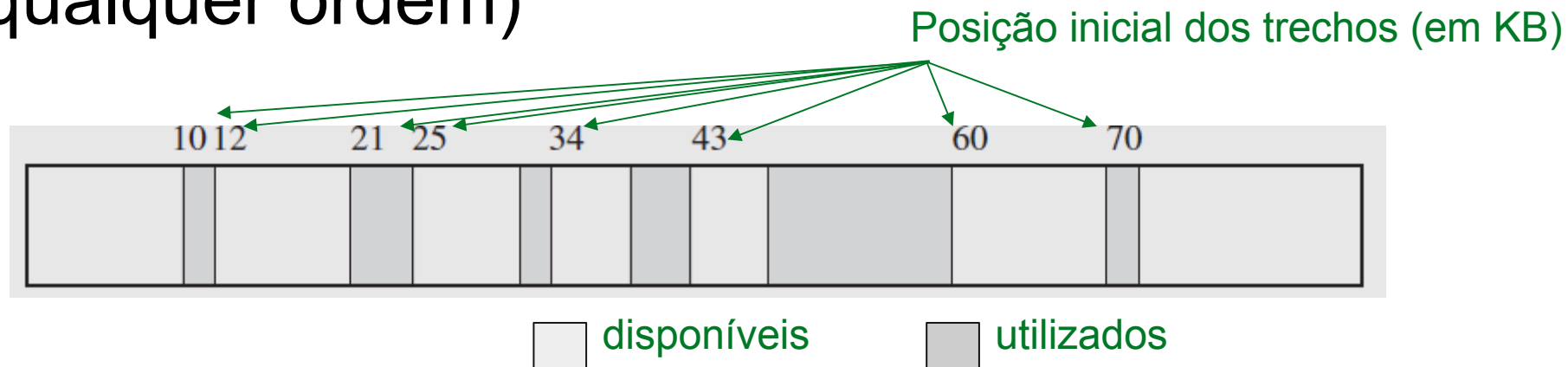
Vetor sequencial contendo trechos contíguos de memória utilizada e disponível (intercalados em qualquer ordem)



Se um trecho de tamanho R for requisitado (por exemplo 8 KB), onde ele será alocado?

Alocação sequencial - Métodos de ajuste sequencial

Vetor sequencial contendo trechos contíguos de memória utilizada e disponível (intercalados em qualquer ordem)



Se um trecho de tamanho R for requisitado (por exemplo 8 KB), onde ele será alocado?

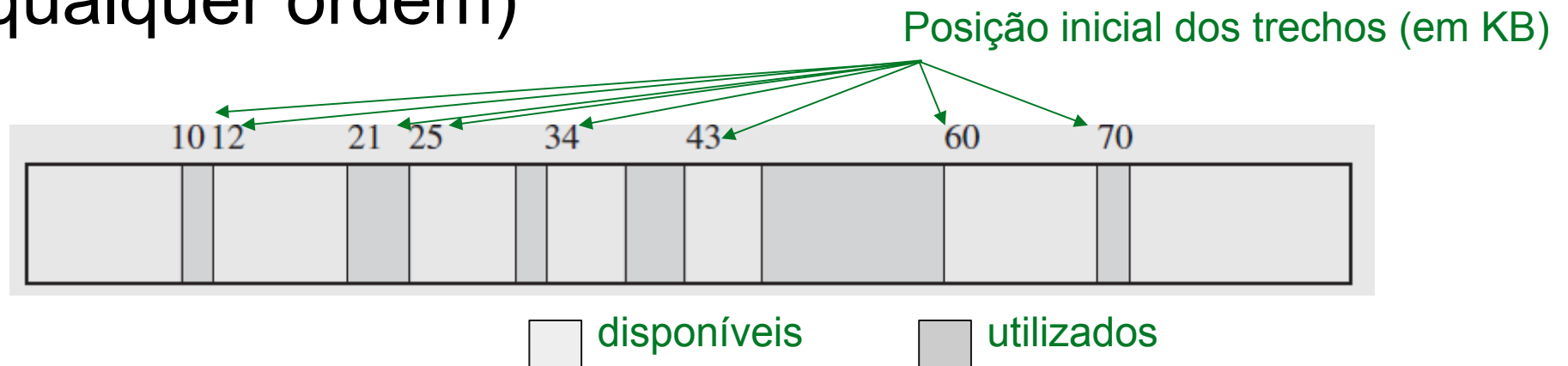
Depende de qual variação do método de ajuste sequencial está sendo utilizado...

Métodos de ajuste sequencial

- Primeiro ajuste: seleciona o primeiro trecho encontrado (a partir do início da lista) grande o suficiente
- Próximo ajuste: seleciona o próximo trecho grande o suficiente (a partir do índice “corrente”, ajustado após a última alocação)
- Melhor ajuste: seleciona o menor trecho dentre os trechos grandes o suficiente
- Pior ajuste: seleciona o maior trecho de todos

Métodos de ajuste sequencial

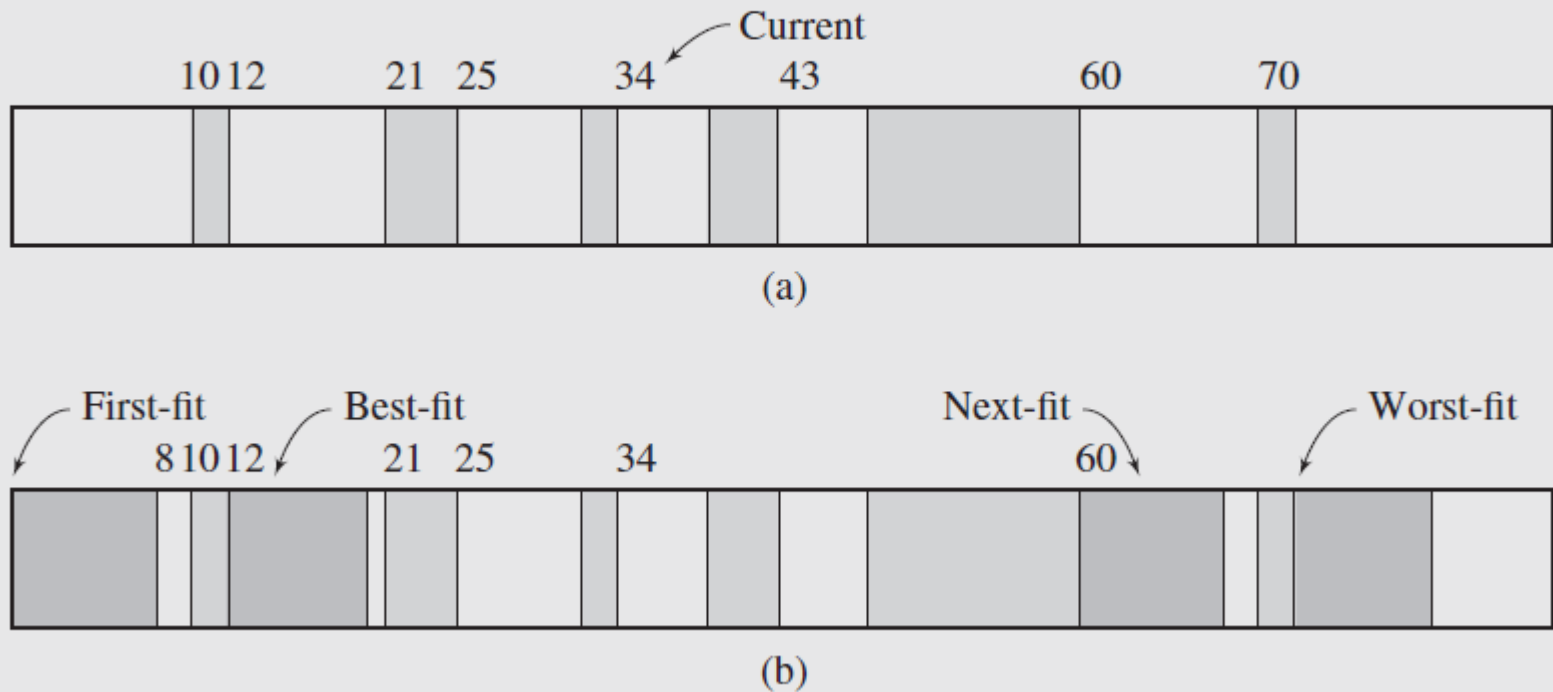
Vetor sequencial contendo trechos contíguos de memória utilizada e disponível (intercalados em qualquer ordem)



Se um trecho de tamanho R for requisitado (por exemplo 8 KB), onde ele será alocado?

Métodos de ajuste sequencial variações

FIGURE 12.1 Memory allocation using sequential-fit methods.



Métodos de ajuste sequencial: variações

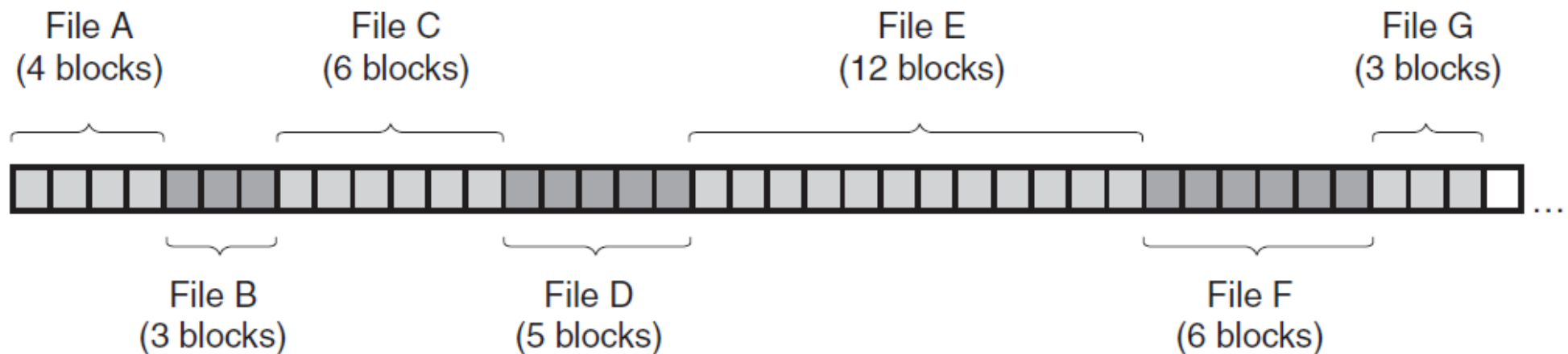
- Qual vocês acham que é melhor?

Métodos de ajuste sequencial: variações

- Qual vocês acham que é melhor?
- Melhor ajuste pode ser o pior: gasta tempo analisando tudo e, a menos que o ajuste seja perfeito, deixa sobrar normalmente trechos pequenos que não podem ser reutilizados
- Pior ajuste tenta evitar esse desperdício, deixando sobrar trechos maiores que podem ser ainda utilizados, e assim posterga a criação de blocos pequenos
- Primeiro ajuste é o mais eficiente: balanço entre tempo de achar um bloco de tamanho suficiente (retorna assim que achar o primeiro) e fragmentação (não deixa sobrar sistematicamente o menor ou maior trecho)
- Próximo ajuste, similar ao primeiro ajuste, mas chega mais rápido ao fim do heap. Nesse caso:
 - Ou gastará mais tempo tendo que rever todo o heap para ver se há trecho disponível
 - Ou considera-se que acabou a memória, e portanto pode ter fragmentado mais.

Alocação sequencial

- Blocos alocados sequencialmente no disco (pelos cilindros)



- Pode ter seus registros ordenados por uma chave de ordenação (sorted files) ou não (heap files)
- Vantagens e desvantagens de cada um?

Alocação sequencial (não ordenado)

- O arquivo, de r registros espalhados em b blocos, não está ordenado nem indexado
- **Inserção** : no final do arquivo. Há espaço disponível (dentro do último bloco ou após ele até o próximo arquivo)?
 - SIM: Eficiente: $O(1)$
 - Copia último bloco no buffer de memória
 - Insere registro
 - Reescreve bloco no disco
 - NÃO: Ineficiente: $O(b)$ – tem que realocar todo o arquivo em outro lugar no disco

Alocação sequencial (não ordenado)

- **Busca:** sequencial não ordenada
 - Tenho que olhar todos os blocos – $O(b)$
 - Lembrando que a busca é por registros, não por blocos
- **Remoção:**
 - Precisa achar onde está o registro – $O(b)$
 - Exclui registro do bloco (que está no buffer) ou resetar bit para inválido
 - Reescrever bloco de volta ao disco (com um espaço vazio) – $O(1)$
 - => Demanda uma reorganização periódica

Alocação sequencial (não ordenado)

- **Modificação de registro de tamanho variável:**
 - Pode exigir a remoção de outros registros a serem inseridos em outros blocos
- **Leitura ordenada: MUITO INEFICIENTE**
 - Exige uma ordenação primeiro! (exigirá ordenação externa se o arquivo inteiro não couber na memória)

Referências

- Slides da Profa. Graça (ICMC) - [http://wiki.icmc.usp.br/index.php/SCC-203_\(gracan\)](http://wiki.icmc.usp.br/index.php/SCC-203_(gracan)) (Arquivos 8, 9 e 12)
- Slides do cap 6 do Ziviani
- GOODRICH et al, **Data Structures and Algorithms in C++**. Ed. John Wiley & Sons, Inc. 2nd ed. 2011. Seção 14.2
- ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. 4 ed. Ed. Pearson-Addison Wesley. Cap 13 (até a seção 13.7).
- TANEMBAUM, A. S. & BOS, H. **Modern Operating Systems**. Pearson, 4th ed. 2015. Cap 4
- RAMAKRISHNAN & GEHRKE. **Data Management Systems**. 3ed. McGrawHill. 2003 Cap 8 e 9.