

# **Exercícios: Fragmentação e Reutilização de Espaço**

Organização e Recuperação de Dados

Profa. Valéria

UEM – CTC – DIN

# Exercício PED

Os registros têm tamanho fixo e o cabeçalho do arquivo ocupa 4 bytes.

- Suponha o arquivo abaixo:

PED = -1				
0	1	2	3	4
... SOUZA   ...	LOPES   ...	CASTRO   ...	SILVEIRA   ...	ALVES   ...

- Supondo o uso de uma **PED** para gerenciar os espaços disponíveis, mostre como fica o arquivo depois das seguintes operações, nesta ordem:
  - a) Remoção do registro de chave CASTRO
  - b) Remoção do registro de chave SOUZA
  - c) Remoção do registro de chave LOPES
  - d) Remoção do registro de chave ALVES
  - e) Inserção do registro de chave BARROS
  - f) Inserção do registro de chave CRUZ

# PED – Pilha de Espaços Disponíveis

PED = -1

0

1

2

3

4

... SOUZA | ... | LOPES | ... | CASTRO | ... | SILVEIRA | ... | ALVES | ...

## a) Remoção CASTRO

PED = 2

0

1

2

3

4

... SOUZA | ... | LOPES | ... | \*-1TRO | ... | SILVEIRA | ... | ALVES | ...

## b) Remoção SOUZA

PED = 0

0

1

2

3

4

... \*2UZA | ... | LOPES | ... | \*-1TRO | ... | SILVEIRA | ... | ALVES | ...

# PED – Pilha de Espaços Disponíveis

PED = 0

0	1	2	3	4
... *2UZA   ...	LOPES   ...	*-1TRO   ...	SILVEIRA   ...	ALVES   ...

## c) Remoção LOPES

PED = 1

0	1	2	3	4
... *2UZA   ...	*0PES   ...	*-1TRO   ...	SILVEIRA   ...	ALVES   ...

## d) Remoção ALVES

PED = 4

0	1	2	3	4
... *2UZA   ...	*0PES   ...	*-1TRO   ...	SILVEIRA   ...	*1VES   ...

# PED – Pilha de Espaços Disponíveis

PED = 4

0	1	2	3	4
... *2UZA   ...	*0PES   ...	*-1TRO   ...	SILVEIRA   ...	*1VES   ...

## e) Inserção do registro BARROS

PED = 1

0	1	2	3	4
... *2UZA   ...	*0PES   ...	*-1TRO   ...	SILVEIRA   ...	<b>BARROS</b>   ...

## f) Inserção do registro CRUZ

PED = 0

0	1	2	3	4
... *2UZA   ...	<b>CRUZ</b>   ...	*-1TRO   ...	SILVEIRA   ...	<b>BARROS</b>   ...

# Exercício LED

Cada registro é precedido por um campo de **2 bytes** que armazena o seu tamanho. O cabeçalho do arquivo ocupa **4 bytes**.

- Suponha o arquivo abaixo:

```
LED = -1
0    4          34          56          81          115
... 28SOUZA|... |20LOPES|... |23CASTRO|... |32SILVEIRA|... |22ALVES|...
```

- Usando a estratégia worst fit para gerenciar a **LED**, mostre como fica o arquivo depois das seguintes operações, nesta ordem:
  - a) Remoção do registro de chave CASTRO
  - b) Remoção do registro de chave SOUZA
  - c) Remoção do registro de chave LOPES
  - d) Remoção do registro de chave ALVES
  - e) Inserção do registro de chave BARROS com 36 bytes
  - f) Inserção do registro de chave CRUZ com 15 bytes

**Obs.:** as sobras produzidas após a reutilização do espaço devem retornar à LED.

# WORST FIT -- LED em ordem decrescente dos tamanhos dos espaços disponíveis

LED = -1

0	4	34	56	81	115										
...	28	SOUZA	...	20	LOPES	...	23	CASTRO	...	32	SILVEIRA	...	22	ALVES	...

## a) Remoção CASTRO

LED = 56

0	4	34	56	81	115										
...	28	SOUZA	...	20	LOPES	...	23*-1	TRO	...	32	SILVEIRA	...	22	ALVES	...

## b) Remoção SOUZA

LED = 4

0	4	34	56	81	115										
...	28*56	ZA	...	20	LOPES	...	23*-1	TRO	...	32	SILVEIRA	...	22	ALVES	...

# WORST FIT -- LED em ordem decrescente dos tamanhos dos espaços disponíveis

LED = 4

0	4	34	56	81	115					
...	28*56	ZA ...	20	LOPES ...	23*-1	TRO ...	32	SILVEIRA ...	22	ALVES ...

## c) Remoção LOPES

LED = 4

0	4	34	56	81	115					
....	28*56	ZA ...	20*-1	ES ...	23*34	TRO ...	32	SILVEIRA ...	22	ALVES ...

## d) Remoção ALVES

LED = 4

0	4	34	56	81	115					
....	28*56	ZA ...	20*-1	ES ...	23*115	RO ...	32	SILVEIRA ...	22*34	ES ...



## WORST FIT -- LED em ordem decrescente dos tamanhos dos espaços disponíveis

LED = 4

0	4	34	56	81	115
...	28*56ZA ...	20*-1ES ...	23*115RO ...	32SILVEIRA ...	22*34ES ...

e) Inserção do registro de chave BARROS com 36 bytes

LED = 4					
4	34	56	81	115	139
...	28*56ZA ...	20*-1ES ...	23*115RO ...	32SILVEIRA ...	22*34ES ...  36BARROS ...

## WORST FIT -- LED em ordem decrescente dos tamanhos dos espaços disponíveis

LED = 4

4	34	56	81	115	139
... 28*56ZA   ...	20*-1ES   ...	23*115RO   ...	32SILVEIRA   ...	22*34ES   ...	36BARROS   ...

f) Inserção do registro de chave CRUZ com 15 bytes (com retorno da sobra para LED)

- **Opção 1:** Deixar a sobra no começo do espaço → não altera o byte-offset que o espaço tinha antes da inserção

LED = 56

4	17	34	56	81	115	139
.. 11*-1.. 15CRUZ   ..	20*04ES   ..	23*115RO   ..	32SILVEIRA   ..	22*34ES   ..	36BARROS   ..	

## WORST FIT -- LED em ordem decrescente dos tamanhos dos espaços disponíveis

LED = 4

4	34	56	81	115	139
... 28*56ZA   ...	20*-1ES   ...	23*115RO   ...	32SILVEIRA   ...	22*34ES   ...	36BARROS   ...

f) Inserção do registro de chave CRUZ com 15 bytes (com retorno da sobra para LED)

— **Opção 2:** Deixar a sobra no final do espaço → a sobra terá um novo byte-offset (21 nesse caso)

LED = 56

4	21	34	56	81	115	139
.. 15CRUZ   ..	11*-1..	20*21ES   ..	23*115RO   ..	32SILVEIRA   ..	22*34ES   ..	36BARROS   ..