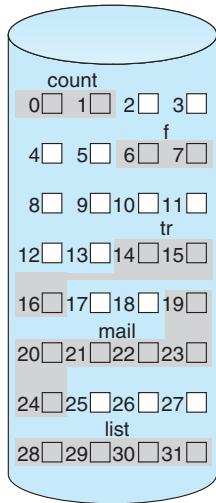


EXERCÍCIO 6.1

Considere a seguinte representação de um disco com 32 blocos de armazenamento, cada bloco de 512 bytes. Da representação faz parte uma diretoria com 5 ficheiros (assuma que o espaço ocupado pela diretoria não é relevante).



- a) Qual é a designação do método de alocação ilustrado pela representação fornecida ?
- b) Qual o espaço consumido pelo ficheiro `tr`, medido em bytes ?
- c) Qual é o bloco do disco que contém o byte 1033 do ficheiro `list` ?
- d) Assumindo que uma sequência de blocos livres é representada por um par `<bloco_inicial, numero_de_blocos>`, apresente a lista de sequências de blocos livres do disco da figura, ordenada por ordem crescente do `bloco_inicial`.
- e) Considere a lista de sequências da alínea d), e suponha que é necessário criar um novo ficheiro `so` de 513 bytes de dados.
- e1) Quais os blocos a atribuir ao ficheiro `so` considerando cada uma das políticas *first-fit*, *best-fit* e *worst-fit* ?
- e2) Apresente a nova lista de blocos livres resultante em cada uma das políticas referidas em e1.
- e3) Qual é a medida da fragmentação interna (em bytes) do último bloco de dados do ficheiro `so` ?

- f) Forneça uma representação de tipo *FAT* que reflita o estado do disco tal como representado na figura acima.
- g) Forneça uma representação de tipo *vetor de bits* que reflita o estado (livre/ocupado) dos blocos do disco da figura acima.

SOLUÇÃO

Dados do Problema:

“disco com 32 blocos de armazenamento”:
“cada bloco de 512 bytes”:

#BLOCOS(disco)=32 blocos
#BYTES(bloco)=512 bytes

Nota: #X(Y)=Z significa que Y tem uma quantidade Z da grandeza X

a) alocação contígua

- b) $\#BYTES(tr) = \#BLOCOS(tr) \times \#BYTES(bloco) = 3 \times 512 = 1536$ bytes
($\#BLOCOS(tr)$ é dado pelo valor da coluna `length` na linha do ficheiro `tr` na tabela da diretoria)
- c) $BL(1033) = 1033 \text{ div } \#BYTES(bloco) = 1033 \text{ div } 512 = 2$ (i.e., o byte 1033 está no bloco lógico 2)
(BL significa Bloco Lógico, donde $BL(b)$ é o bloco lógico que contém o byte b; div é o quociente da divisão inteira)

$BF(1033, list) = BF_{\text{inicial}}(list) + BL(1033) = 28 + 2 = 30$ (i.e., o byte 1033 está no bloco físico 30)
(BF significa Bloco Físico, donde $BF(b,f)$ é o bloco do disco que contém o byte b do ficheiro f)
($BF_{\text{inicial}}(list)$ é dado pelo valor da coluna `start` na linha do ficheiro `list` na tabela da diretoria)

- d) $LBL(\text{disco}) = <2,4> <8,6> <17,2> <25,3>$ ($LBL(\text{disco})$ significa Lista de Blocos Livres do disco)

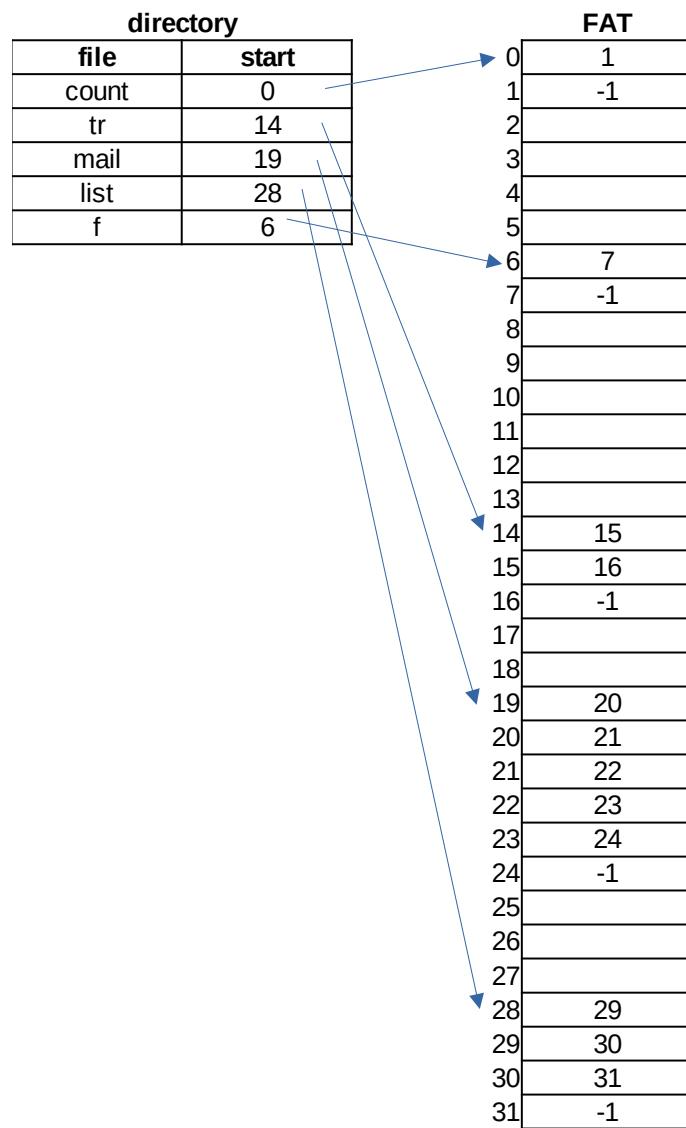
- e1) $\#BLOCOS(so) = \text{ceil} [\#BYTES(so) / \#BYTES(bloco)] = \text{ceil} (513 / 512) = \text{ceil} (1,0019...) = 2$ blocos
($\text{ceil}(x)$ representa a função que arredonda x para o inteiro y mais próximo tal que $y \geq x$)

first-fit: a primeira sequência de $LBL(\text{disco})$ que fornece pelo menos 2 blocos é $<2,4>$; os blocos seriam **2 e 3**
best-fit: a sequência de $LBL(\text{disco})$ com dimensão mais próxima de 2 blocos é $<17,2>$; os blocos seriam **17 e 18**
worst-fit: a sequência de $LBL(\text{disco})$ com dimensão menos próxima de 2 blocos é $<8,6>$; os blocos seriam **8 e 9**

Nota: por regra, os blocos de uma sequência livre consomem-se a partir do seu início

- e2) $LBL(\text{disco})$ após first-fit = $<4,2> <8,6> <17,2> <25,3>$
 $LBL(\text{disco})$ após best-fit = $<2,4> <8,6> <25,3>$
 $LBL(\text{disco})$ após worst-fit = $<2,4> <10,4> <17,2> <25,3>$
- e3) $\text{FRAG}(so) = \#BLOCOS(so) \times \#BYTES(bloco) - \#BYTES(so) = 2 \times 512 - 513 = 511$ bytes

f)



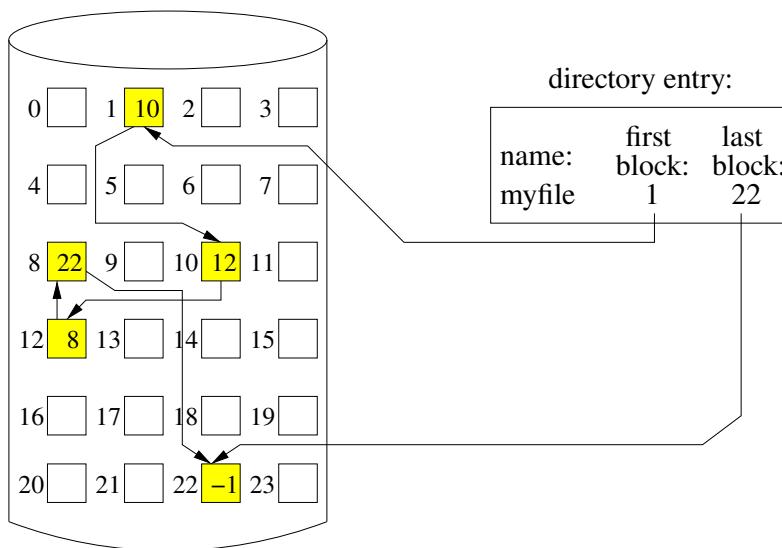
Nota: o valor -1 numa célula x da FAT significa que o bloco físico x é o último bloco do ficheiro em causa

g) 1100001100000011100111110001111

(considerando 0 = livre e 1 = ocupado)

EXERCÍCIO 6.2

Considere a seguinte representação de um disco, com 24 blocos de armazenamento, medindo cada bloco 4KBytes, e medindo cada índice de bloco 4 bytes. Da representação faz parte uma entrada numa certa diretoria, relativa a um ficheiro **myfile** (no contexto deste cenário, assume-se que o espaço ocupado pela diretoria em si é de zero bytes).



g) Considerando que uma sequência de blocos livres é representada por um par de inteiros da forma `<bloco_inicial, numero_de_blocos>`, apresente as sequências de blocos livres do disco da figura, por ordem decrescente do bloco_inicial (auxílio: a primeira sequência é `<23, 1>`, a segunda é `<13, 9>`, e assim sucessivamente).

h) Considerando a lista de sequências da alínea **g**), suponha que é necessário criar um novo ficheiro **xpto** de 5000 bytes de dados. Assumindo que os blocos desse ficheiro devem ser contíguos e uma política de alocação *best-fit*:

h1) Quais os blocos a atribuir ao ficheiro **xpto** ?

h2) Qual é a medida da fragmentação interna (em bytes) do último bloco de dados do ficheiro **xpto** ?

h3) Qual a nova sequência de blocos livres que resulta da criação do ficheiro **xpto** ?

SOLUÇÃO

Dados do Problema:

“disco, com 24 blocos de armazenamento”:

“cada bloco de 4KBytes”:

“medindo cada índice de bloco 4 bytes.”:

Nota: #X(Y)=Z significa que Y tem uma quantidade Z da grandeza X

lista dos blocos físicos do ficheiro myfile:

(a posição de cada bloco físico (BF) nesta lista corresponde ao respetivo bloco lógico (BL); donde, o BL 0 corresponde ao BF 1, o BL 1 corresponde ao BF 10, o BL 2 corresponde ao BF 12, etc.; notar que as posições são numeradas a partir de zero (0))

a) alocação ligada

c) $\#BYTES_{total}(myfile) = \#BLOCOS(myfile) \times \#BYTES(bloco) = 5 \times 4096 = 20480$ bytes

($\#BLOCOS(myfile)$ é o número de blocos da lista ligada que se inicia no bloco 1 termina no bloco 22)

d) $\#BYTES_{dados}(myfile) = \#BLOCOS(myfile) \times \#BYTES_{dados}(bloco) = 5 \times 4092 = 20460$ bytes

(considerando que $\#BYTES_{dados}(bloco) = \#BYTES(bloco) - \#BYTES(indice) = 4096 - 4 = 4092$ bytes)

e) $BL(4095) = 4095 \text{ div } \#BYTES_{dados}(bloco) = 4095 \text{ div } 4092 = 1$ (i.e., o byte 4095 está no bloco lógico 1)

(BL significa Bloco Lógico, donde $BL(b)$ é o bloco lógico que contém o byte b; div é o quociente da divisão inteira)

$BF(4095, myfile) = BF$ na posição 1 da $LBF(myfile) = 10$ (i.e., o byte 4095 está no bloco físico 10)

(BF significa Bloco Físico, donde $BF(b,f)$ é o bloco do disco que contém o byte b do ficheiro f)

f) 01000000101010000000010

(considerando 0 = livre e 1 = ocupado)

g) $LBL(disco) = <23,1> <13,9> <11,1> <9,1> <2,6> <0,1>$ ($LBL(disco)$ significa Lista de Blocos Livres do disco)

a) Qual é a designação do método de alocação ilustrado pela representação fornecida ?

b) Forneça uma representação equivalente à fornecida, mas em formato FAT.

c) Qual o espaço total consumido pelo ficheiro **myfile**, medido em bytes ?

d) Qual o espaço consumido pelos dados do ficheiro **myfile**, medido em bytes ?

e) Qual é o bloco do disco que contém o byte 4095 dos dados do ficheiro **myfile** ?

f) Apresente um vetor de bits que traduza o estado (livre/ocupado) dos blocos do disco.

h) #BLOCOS(xpto) = ceil [#BYTES_dados(xpto) / #BYTES_dados(bloco)] = ceil (5000 / **4092**) = ceil (1,22...) = 2 blocos
 (ceil(x) representa a função que arredonda x para o inteiro y mais próximo tal que y \geq x)

h1) **best-fit**: a sequência de LBL(disco) com dimensão mais próxima de 2 blocos é <2,6>; os blocos seriam **2 e 3**

Nota: por regra, os blocos de uma sequência livre consomem-se a partir do seu início

h3) LBL(disco) após best-fit = <23,1> <13,9> <11,1> <9,1> <4,4> <0,1>

h2) FRAG(xpto) = #BLOCOS(xpto) x #BYTES_dados(bloco) - #BYTES_dados(xpto) = 2 x **4092** - 5000 = **3184** bytes

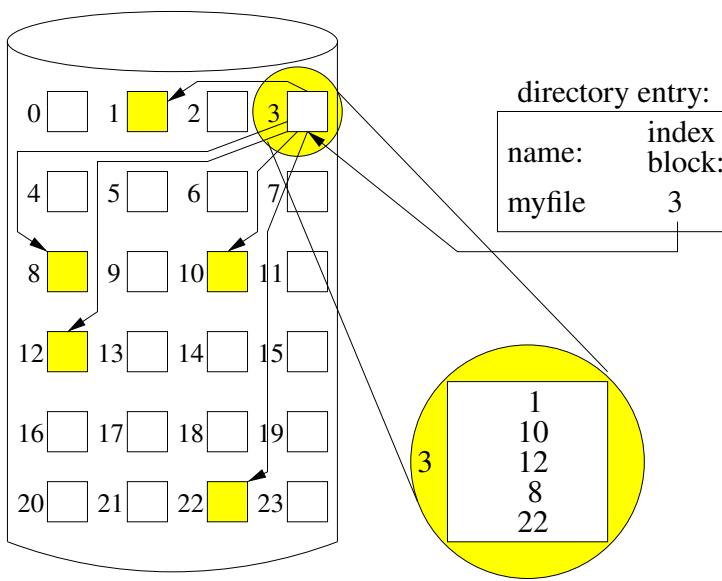
b)

directory		FAT
file	start	
myfile	1	10
		2
		3
		4
		5
		6
		7
		22
		9
		12
		11
		8
		13
		14
		15
		16
		17
		18
		19
		20
		21
		-1
		23

Nota: o valor -1 numa célula x da FAT significa que o bloco físico x é o último bloco do ficheiro em causa

EXERCÍCIO 6.3

Considere a seguinte representação de um disco com 24 blocos de armazenamento, medindo cada bloco 512 bytes. Da representação faz parte uma entrada numa certa diretoria, relativa a um ficheiro **myfile**, bem com o respetivo bloco de indexação. No contexto deste cenário, assume-se que o espaço ocupado pela diretoria não é relevante.



- a) Qual é a designação do método de alocação ilustrado pela representação fornecida ?
- b) Qual o espaço consumido pelos **dados** do ficheiro **myfile**, medido em bytes ?
- c) Qual o espaço **total** consumido pelo ficheiro **myfile**, medido em bytes ?
- d) Qual é o bloco do disco que contém o byte 1033 dos dados do ficheiro **myfile** ?
- e) Assumindo que uma sequência de blocos livres é representada por um par de números inteiros **<numero_de_blocos, bloco_inicial>**, apresente a lista de sequências de blocos livres do disco da figura, ordenada por ordem decrescente do **numero_de_blocos** (em caso de empate, desempate usando ordem decrescente do **bloco_inicial**).

f) Considerando a lista de sequências da alínea e), suponha que é necessário criar um novo ficheiro **myfile2** de 513 bytes de dados. Assumindo que os blocos desse ficheiro devem ser contíguos e uma política de alocação *worst-fit*:

- f1) Quais os blocos a atribuir ao ficheiro **myfile2** ?
- f2) Forneça uma representação do bloco de indexação do ficheiro **myfile2**.
- f3) Qual é a medida da fragmentação interna (em bytes) do último bloco de dados do ficheiro **myfile2** ?
- f4) Qual a nova lista de sequências de blocos livres após a criação do ficheiro **myfile2** ?
- f5) Forneça uma representação de tipo FAT considerando apenas os blocos de dados do ficheiro **myfile2**.

SOLUÇÃO

Dados do Problema:

“disco, com 24 blocos de armazenamento”;
“medindo cada bloco 512 bytes”.

Nota: #X(Y)=Z significa que Y tem uma quantidade Z da grandeza X

lista dos blocos físicos com dados do ficheiro myfile:
(a posição de cada bloco físico (BF) nesta lista corresponde ao respetivo bloco lógico (BL); donde, o BL 0 corresponde ao BF 1, o BL 1 corresponde ao BF 10, o BL 2 corresponde ao BF 12, etc.; notar que as posições são numeradas a partir de zero (0))
(ao contrário da alocação ligada, a LBF não está dispersa no disco, mas concentrada no bloco de indexação, sendo um array)

a) alocação indexada

b) #BYTES_dados(myfile) = #BLOCOS_dados(myfile) x #BYTES(bloco) = 5 x 512 = **2560** bytes
(#BLOCOS_dados(myfile) corresponde à quantidade de índices de blocos físicos válidos guardados na LBF(myfile))

c) #BYTES_total(myfile) = #BLOCOS(myfile) x #BYTES(bloco)
= [#BLOCOS_index(myfile) + #BLOCOS_dados(myfile)] x #BYTES(bloco)
= (1 + 5) x 512 = 6 x 512 = **3072** bytes

(#BLOCOS(myfile) tem de considerar o número de blocos gastos em indexação, que neste caso é apenas 1)

d) BL(1033) = 1033 div #BYTES(bloco) = 1033 div 512 = **2** (i.e., o byte 1033 está no bloco lógico 2)
(BL significa Bloco Lógico, donde BL(b) é o bloco lógico que contém o byte b; div é o quociente da divisão inteira)

BF(1033,myfile) = BF na posição 2 da LBF_dados(myfile) = **12** (i.e., o byte 1033 está no bloco físico 12)
(BF significa Bloco Físico, donde BF(b,f) é o bloco do disco que contém o byte b do ficheiro f)

e) LBL(disco)= <9,13> <4,4> <1,23> <1,11> <1,9> <1,2> <1,0> (LBL(disco) significa Lista de Blocos Livres do disco)

f) #BLOCOS_dados(myfile2) = ceil [#BYTES_dados(myfile2) / #BYTES(bloco)]
 $= \text{ceil} (513 / 512) = \text{ceil} (1,0019...) = 2$ blocos
 (ceil(x) representa a função que arredonda x para o inteiro y mais próximo tal que $y \geq x$)

#BLOCOS(myfile2) = #BLOCOS_index myfile2 + #BLOCOS_dados myfile2 = 1 + 2 = 3 blocos
 (#BLOCOS myfile2) tem de considerar o número de blocos gastos em indexação, que neste caso é apenas 1)

f1) worst-fit: a sequência de LBL(disco) com dimensão menos próxima de 3 blocos é <9,13>; os blocos seriam 13, 14 e 15

Nota: por regra, os blocos de uma sequência livre consomem-se a partir do seu início

f2)

index block (13)	
0	14
1	15
2	-1
...	-1

Notas:

- o valor -1 numa posição x do bloco de indexação significa que não há bloco físico para o bloco lógico x;
 - como não é fornecida informação sobre a dimensão das referências para blocos físicos, não é possível saber quantas referências dessas cabem num bloco de indexação e por isso não é possível indicar o último índice do bloco de indexação

f3) FRAG(myfile2) = #BLOCOS_dados myfile2 x #BYTES(bloco) - #BYTES_dados myfile2 = 2 x 512 - 513 = 511 bytes

f4) LBL(disco) após worst-fit = <6,16> <4,4> <1,23> <1,11> <1,9> <1,2> <1,0>

f5)

The diagram illustrates the mapping between a file entry in the directory and its corresponding entry in the FAT table.

directory	
file	start
myfile2	14

A blue arrow points from the "start" address (14) in the directory entry to the 14th entry in the FAT table.

FAT	
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	15
15	-1
16	
17	
18	
19	
20	
21	
22	
23	

Notas:

- o valor -1 numa célula x da FAT significa que o bloco físico x é o último bloco do ficheiro em causa
 - assumiu-se que havendo uma FAT, o bloco de indexação (13) deixa de ser necessário