

Capítulo 6

Gerência de arquivos

1. Enumere os principais atributos de um arquivo.
2. Enumere as principais operações sobre arquivos.
3. Apresente e comente as principais formas de atribuição de tipos aos arquivos. Quais são as vantagens e desvantagens de cada uma?
4. Analise as seguintes afirmações relativas a formatos de arquivos:
 - I. Um *magic number* consiste de um atributo numérico separado que identifica o tipo de arquivo.
 - II. A forma mais comum de identificação de tipo de arquivo é o uso de extensões ao seu nome.
 - III. Arquivos de texto em sistemas DOS e UNIX diferem nos caracteres de controle usados para identificar o fim de arquivo.
 - IV. Para a maioria dos núcleos de sistema operacional, arquivos são quase sempre vistos como meras sequências de bytes.
 - V. ELF e PE são dois formatos típicos de arquivos de configuração.
 - VI. O padrão MIME é usado no Linux para identificação de tipos de arquivos pelo sistema operacional.

As alternativas corretas são:

- (a) II e IV
- (b) II e V
- (c) I e III
- (d) IV e V
- (e) III e VI

Justifique as afirmações julgadas erradas (Assim: *XIV está errada porque ...*):

5. O que é um ponteiro de arquivo? Para que ele serve?

6. Comente as principais formas de acesso a arquivos. Qual o uso mais apropriado para cada uma delas?
7. Quais as principais estruturas de diretórios empregadas em sistemas operacionais?
8. Do ponto de vista lógico, quais as principais diferenças entre a estrutura de diretórios Unix e Windows?
9. Explique os tipos de referências possíveis a arquivos em uma estrutura de diretórios.
10. Explique as formas de referência a arquivos direta, absoluta e relativa.
11. Analise as seguintes afirmações relativas ao uso de arquivos:
 - I. No acesso sequencial, o ponteiro de posição corrente do arquivo é reiniciado a cada operação.
 - II. O acesso direto pode ser implementado usando o acesso sequencial e operações de posicionamento do ponteiro do arquivo.
 - III. No acesso mapeado em memória, o conteúdo do arquivo é copiado para a memória RAM durante a sua abertura.
 - IV. O acesso indexado é raramente implementado pelo núcleo em sistemas operacionais *desktop*, sendo mais frequente em ambientes *mainframe*.
 - V. Travas de uso exclusivo e compartilhado implementam um modelo de sincronização de tipo *produtor/consumidor* no acesso ao arquivo.
 - VI. Segundo a semântica de compartilhamento UNIX, o conteúdo de um arquivo é considerado imutável durante um compartilhamento.

As alternativas corretas são:

- (a) II e IV
- (b) II e V
- (c) III e V
- (d) I e IV
- (e) III e VI

Justifique as afirmações julgadas erradas (Assim: *XIV está errada porque ...*):

12. Um conjunto de processos p_1 , p_2 , p_3 e p_4 abrem em leitura/escrita um arquivo compartilhado contendo um número inteiro, cujo valor inicial é 34. As operações realizadas pelos processos são indicadas na tabela a seguir no formato $[t, op]$, onde t é o instante da operação e op é a operação realizada:

p_1	p_2	p_3	p_4
[0, open]	[3, open]	[7, open]	[9, open]
[2, write 41]	[6, write 27]	[8, read X]	[11, read Y]
[6, close]	[8, close]	[9, write 4]	[12, close]
		[10, close]	

Considerando a semântica de sessão para o compartilhamento de arquivos, determine os valores de X e Y , **explicando seu raciocínio**. Cada operação de escrita no arquivo substitui o valor anterior.

13. Enumere principais problemas a resolver na implementação de um sistema de arquivos.
14. Apresente a arquitetura de gerência de arquivos presente em um sistema operacional típico, explicando seus principais elementos constituintes.
15. Explique o que é alocação contígua de arquivos, apresentando suas vantagens e desvantagens.
16. No contexto de alocação de arquivos, o que significa o termo *best-fit*?
17. Explique a alocação de arquivos em listas encadeadas, apresentando suas principais vantagens e desvantagens.
18. Explique a estrutura do sistema de arquivos conhecido como FAT, comentando sobre suas qualidades e deficiências.
19. Por que a alocação de arquivos em listas encadeadas é considerada pouco robusta? O que pode ser feito para melhorar essa característica?
20. Explique o esquema de alocação indexada de arquivos usando índices multi-níveis.
21. O que é fragmentação interna e fragmentação externa? Por que elas ocorrem?
22. Analise o impacto das fragmentações interna e externa nos sistemas de alocação contígua, indexada e por lista encadeadas.
23. Considere um sistema operacional hipotético que suporte simultaneamente as estratégias de alocação contígua, encadeada e indexada para armazenamento de arquivos em disco. Que critérios devem ser considerados para decidir a estratégia a usar para cada arquivo em particular?
24. Avalie as seguintes afirmações sobre as técnicas de alocação de arquivos:
 - I. A alocação contígua é muito utilizada em sistemas desktop, por sua flexibilidade.
 - II. A alocação FAT é uma alocação encadeada na qual os ponteiros de blocos foram transferidos para um vetor de ponteiros.

- III. Na alocação indexada os custos de acesso seqüencial e aleatório a blocos são similares.
- IV. Na alocação contígua, blocos defeituosos podem impedir o acesso aos demais blocos do arquivo.
- V. Na alocação contígua, o custo de acesso a blocos aleatórios é alto.
- VI. Apesar de complexa, a alocação indexada é muito usada em *desktops* e servidores.

As afirmações corretas são:

- (a) II, III e VI
- (b) I, III e IV
- (c) I, IV e V
- (d) II, IV e V
- (e) IV, V e VI

Justifique as afirmações julgadas erradas (Assim: *XIV está errada porque ...*):

25. Considerando um arquivo com 500 blocos em disco, calcule quantas leituras e quantas escritas em disco são necessárias para (a) inserir um novo bloco no início do arquivo ou (b) inserir um novo bloco no final do arquivo, usando as formas de alocação de blocos contígua, encadeada e indexada.

Alocação	Contígua		Encadeada		Indexada	
Operações	leituras	escritas	leituras	escritas	leituras	escritas
Inserir um novo bloco no início do arquivo						
Inserir um novo bloco no final do arquivo						

Observações:

- (a) Considere somente as operações de leitura e escrita nos blocos do próprio arquivo e no *i-node*; a tabela de diretório sempre está em memória;
- (b) Para a alocação contígua, assuma que não há espaço livre depois do arquivo, somente antes dele;
- (c) Para a alocação encadeada, assuma que a tabela de diretório contém apenas um ponteiro para o início do arquivo no disco. Os ponteiros dos blocos estão contidos nos próprios blocos;
- (d) Para a alocação indexada, considere *i-nodes* com somente um nível, contendo somente os ponteiros para os blocos de dados. O *i-node* está no disco.

26. Considere um disco rígido com capacidade total de 1 Mbyte, dividido em 1.024 blocos de 1.024 bytes cada. Os dez primeiros blocos do disco são reservados para a tabela de partições, o código de inicialização (*boot*) e o diretório raiz do sistema de arquivos. Calcule o tamanho máximo de arquivo (em bytes) que pode ser criado nesse disco para cada uma das formas de alocação a seguir, explicando seu raciocínio:

- Alocação contígua.
- Alocação encadeada, com ponteiros de 64 bits contidos nos próprios blocos.
- Alocação indexada, com *i-nodes* contendo somente ponteiros diretos de 64 bits; considere que o *i-node* não contém meta-dados, somente ponteiros, e que ele ocupa exatamente um bloco do disco.

27. Considerando a tabela FAT (*File Allocation Table*) a seguir, indique:

- o número de blocos ocupados pelo arquivo `relat.pdf`;
- o tamanho (em blocos) do maior arquivo que ainda pode ser criado nesse disco;
- quais arquivos estão íntegros e quais estão corrompidos por blocos defeituosos (*bad blocks*);
- quantos blocos do disco estão perdidos, ou seja, não são usados por arquivos nem estão marcados como livres ou defeituosos.

Na tabela, a letra R indica bloco reservado (*Reserved*), F indica bloco livre (*Free*), L indica o último bloco de um arquivo (*Last*) e B indica bloco defeituoso (*Bad*).

arquivo	início	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
		R	R	R	R	R	F	17	F	15	68	13	53	F	L	63	L	F	26	F	F
readme.txt	76	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
icone.gif	14	33	L	F	38	L	F	11	55	F	36	F	35	43	B	F	B	20	F	8	F
retrato.jpg	29	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
relat.pdf	6	21	32	F	50	B	L	F	F	40	F	L	45	F	58	F	B	F	F	72	F
format.exe	31	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
carta.doc	67	44	F	F	51	F	F	F	60	24	F	F	F	10	27	F	F	41	F	L	F
programa.c	73																				

28. O sistema de arquivos indexado do sistema *Minix* possui os seguintes campos em cada *i-node*:

- meta-dados (tipo, dono, grupo, permissões, datas e tamanho)
- 7 ponteiros diretos
- 1 ponteiro indireto
- 1 ponteiro duplamente indireto

A implementação básica desse sistema de arquivos considera blocos de 1.024 bytes e ponteiros de 32 bits. Desenhe o diagrama do sistema de arquivos e calcule o tamanho máximo de arquivo que ele suporta, indicando seu raciocínio.

29. O sistema de arquivos indexado *ext2fs*, usado no *Linux*, possui os seguintes campos em cada *i-node*:

- meta-dados (tipo, dono, grupo, permissões, datas e tamanho)
- 12 ponteiros diretos
- 1 ponteiro indireto
- 1 ponteiro duplamente indireto
- 1 ponteiro triplamente indireto

A implementação básica do *ext2fs* considera blocos de 1.024 bytes e ponteiros de 64 bits. Desenhe o diagrama do sistema de arquivos e determine o tamanho máximo de arquivo que ele suporta, indicando seu raciocínio.

30. Explique como é efetuada a gerência de espaço livre através de *bitmaps*.