

Cap. 5 – Gerenciamento de Memória Secundária

Parte 2

Prof. Marcelo Moreno

moreno@ice.ufjf.br

Armazenamento da informação

- Processos desejam armazenar a informação usada, gerada ou mantida
 - Deve ser possível armazenar uma quantidade grande de informação
 - A informação deve sobreviver ao término do processo que a usa
 - Múltiplos processos têm de ser capazes de acessar a informação concorrentemente
 - Abstração da unidade de informação deve ser bem entendida pelo usuário
- Arquivos
 - Unidade de informação armazenada normalmente em dispositivo de memória secundária e de forma persistente



Organização de Arquivos

- Sistema de Arquivos
 - Estruturação feita pelo sistema operacional para organizar arquivos
 - Estabelece regras para:
 - Estruturação de arquivo
 - Nomeação de arquivo
 - Acesso a arquivo
 - Proteção de arquivo
- Todo sistema operacional moderno possui abstração de sistema de arquivos
 - Mesmo que isso seja transparente ao usuário



Nomeação de Arquivo

- Acessar arquivos pelo nome é importante peça na transparência da gerência de arquivos ao usuário
 - Quando um processo cria um arquivo, deve dar um nome a ele
- Regras de nomes variam muito
 - Todos os sistemas de arquivo suportam pelo menos 8 caracteres para o nome de arquivo
 - Todos normalmente suportam nomes com letras, algarismos e pontos
 - Alguns sistemas permitem nomes com 255 caracteres, inclusive caracteres especiais
 - Alguns distinguem letras maiúsculas de minúsculas
 - Maioria suporta nomes com duas partes, separadas por ponto



Nomeação de Arquivos

Extensão	Significado
file.bak	Arquivo de cópia de segurança
file.c	Programa fonte em C
file.gif	Imagem no formato de intercâmbio gráfico da Compuserve (graphical interchange format)
file.hlp	Arquivo de auxílio
file.html	Documento da World Wide Web em Linguagem de Marcação de Hipertexto (<i>hypertext markup language</i> — HTML)
file.jpg	Imagem codificada com o padrão JPEG
file.mp3	Música codificada no formato de áudio MPEG — camada 3
file.mpg	Filme codificado com o padrão MPEG
file.o	Arquivo-objeto (saída do compilador, ainda não ligado)
file.pdf	Arquivo no formato portátil de documentos (por table document format — PDF)
file.ps	Arquivo no formato PostScript
file.tex	Entrada para o programa de formatação TEX
file.txt	Arquivo de textos
file.zip	Arquivo comprimido

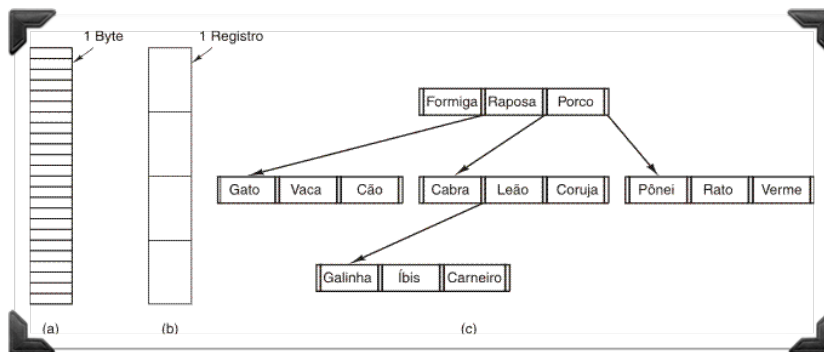


Estrutura de Arquivos

- Sistemas operacionais normalmente tratam arquivos como uma sequência de bytes
 - Trata-se de uma sequência de unidades mínimas de informação
 - Oferece flexibilidade para que outras organizações internas sejam feitas por programas
 - “Não ajudam, mas também não atrapalham”
- Principais estruturações de arquivo
 - Sequência de bytes
 - Sequência de registros
 - Árvore



Estrutura de Arquivos



Tipos de Arquivos

- Tipos básicos são:
 - Arquivos regulares
 - Diretórios
- Alguns sistemas possuem arquivos especiais
 - Arquivos especiais de caractere
 - Arquivos especiais de bloco
- Arquivos regulares podem ser arquivos textuais padrão ASCII
 - Marcação para fim de linha (CR, LF, CR/LF)



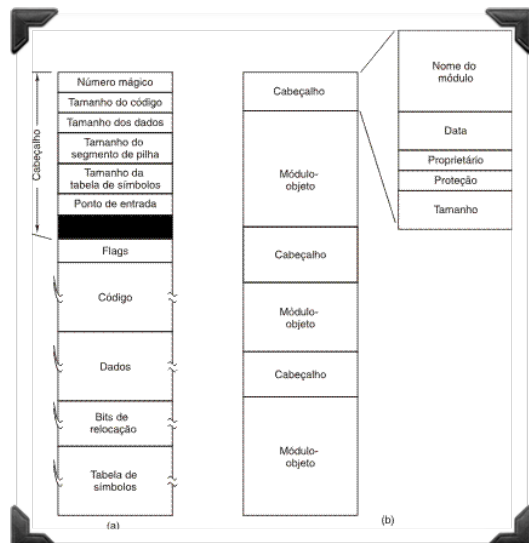
Tabela ASCII

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL_	63	3F	?	95	5F	_	127	7F	DEL

Tipos de Arquivos

- **Arquivos regulares podem ser binários**
 - Estruturação ditada pelo programa que o criou e usa
 - O S.O. também estabelece arquivos binários específicos
 - **Arquivos executáveis Unix**
 - Cabeçalho, Texto, Dados, bits de relocação e tabela de símbolos
 - **Arquivos de repositório Unix (Archive)**
 - Coleção de módulos não ligados

Arquivos Unix: Executável e Archive



Acesso a Arquivos

- **Acesso sequencial**
 - lê todos os bytes/registros desde o início
 - não pode saltar ou ler fora de sequência
 - conveniente quando o meio era a fita magnética
- **Acesso aleatório**
 - bytes/registros lidos em qualquer ordem
 - essencial para sistemas de bases de dados
 - ler pode ser ...
 - mover marcador de arquivo (seek), e então ler ou ...
 - ler e então mover marcador de arquivo

Atributos de Arquivos

Atributo	Significado
Proteção	Quem pode ter acesso ao arquivo e de que maneira
Senha	Senha necessária para ter acesso ao arquivo
Criador	ID da pessoa que criou o arquivo
Proprietário	Atual proprietário
Flag de apenas para leitura	0 para leitura/escrita; 1 se apenas para leitura
Flag de oculto	0 para normal; 1 para não exibir nas listagens
Flag de sistema	0 para arquivos normais; 1 para arquivos do sistema
Flag de repositório (archive)	0 se foi feita cópia de segurança; 1 se precisar fazer cópia de segurança
Flag ASCII/binário	0 para arquivo ASCII; 1 para arquivo binário
Flag de acesso aleatório	0 se apenas para acesso seqüencial; 1 para acesso aleatório
Flag de temporário	0 para normal; 1 para remover o arquivo na saída do processo
Flag de impedimento	0 para desimpedido; diferente de zero para impedido
Tamanho do registro	Número de bytes em um registro
Posição da chave	Deslocamento da chave dentro de cada registro
Tamanho da chave	Número de bytes no campo-chave
Momento da criação	Data e horário em que o arquivo foi criado
Momento do último acesso	Data e horário do último acesso ao arquivo
Momento da última mudança	Data e horário da última mudança ocorrida no arquivo
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número de bytes que o arquivo pode vir a ter

Operações com arquivos

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek
- Get attributes
- Set Attributes
- Rename

Exemplo

```

/* Programa que copia arquivos. Verificação e relato de erros é mínimo */
#include <sys/types.h>          /* inclui os arquivos de cabeçalho necessários */
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[]); /* protótipo ANSI */

#define BUF_SIZE 4096           /* usa um tamanho de buffer de 4096 bytes */
#define OUTPUT_MODE 0700        /* bits de proteção para o arquivo de saída */

int main(int argc, char *argv[])
{
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if (argc != 3) exit(1); /* erro de sintaxe se argc não for 3 */

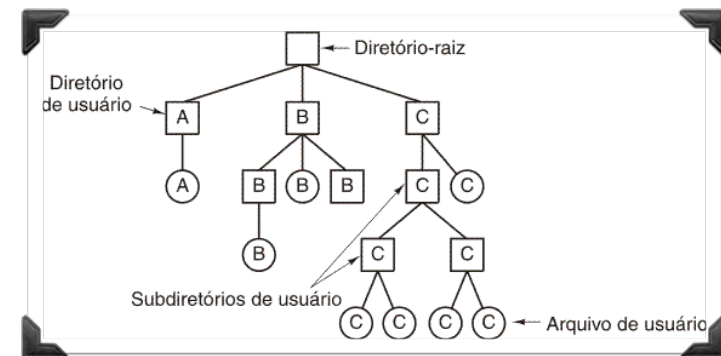
    /* Abre o arquivo de entrada e cria o arquivo de saída */
    in_fd = open(argv[1], O_RDONLY); /* abre o arquivo de origem */
    if (in_fd < 0) exit(2); /* se não puder ser aberto, saída */
    out_fd = creat(argv[2], OUTPUT_MODE); /* cria o arquivo de destino */
    if (out_fd < 0) exit(3); /* se não puder ser criado, saída */

    /* Laço de cópia */
    while (TRUE) {
        rd_count = read(in_fd, buffer, BUF_SIZE); /* lê um bloco de dados */
        if (rd_count == 0) break; /* se fim de arquivo ou erro, sai do laço */
        wt_count = write(out_fd, buffer, rd_count); /* escreve dados */
        if (wt_count <= 0) exit(4); /* wt_count <= 0 é um erro */
    }

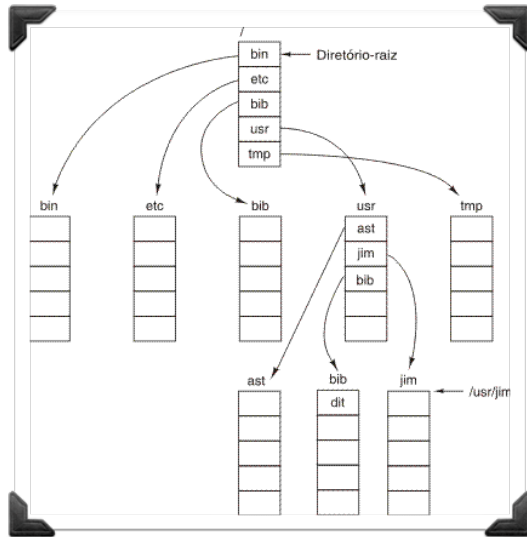
    /* Fecha os arquivos */
    close(in_fd);
    close(out_fd);
    if (rd_count == 0) /* nenhum erro na última leitura */
        exit(0);
    else
        exit(5); /* erro na última leitura */
}

```

Hierarquia de diretórios



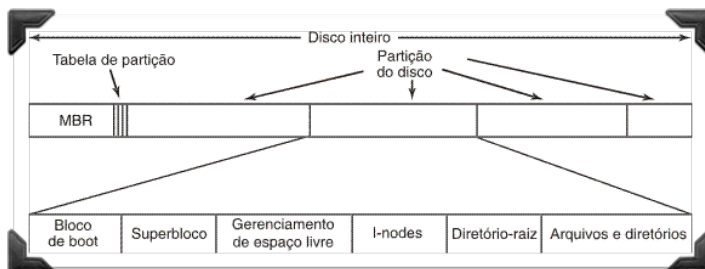
Nomes de Caminhos



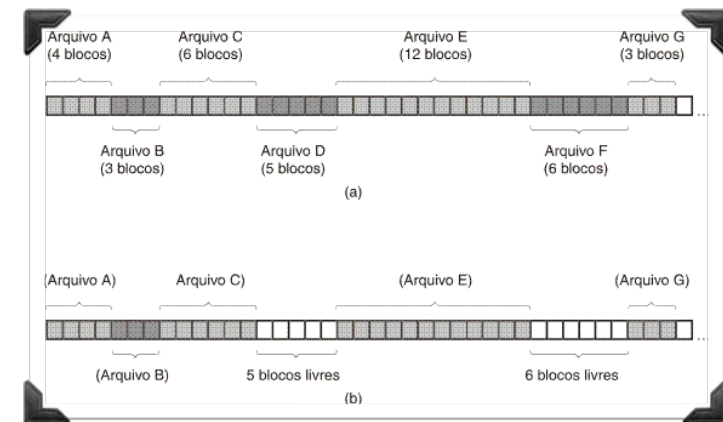
Operações com Diretórios

- Create
- Delete
- Opendir
- Closedir
- Readdir
- Rename
- Link
- Unlink

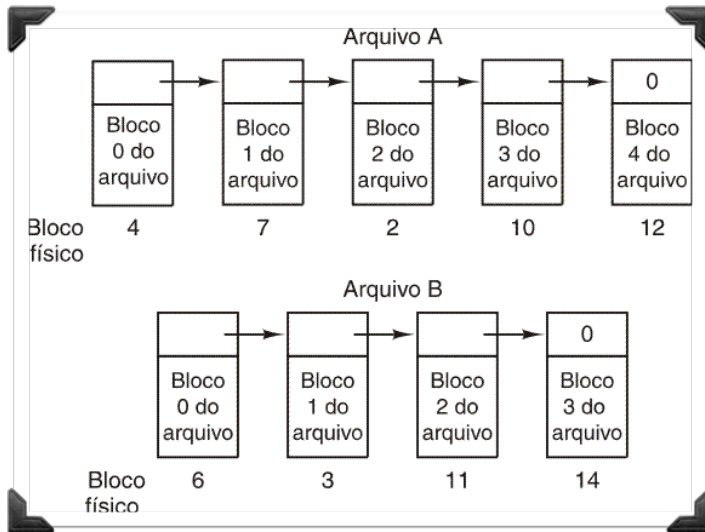
Implementação de Sistemas de Arquivos



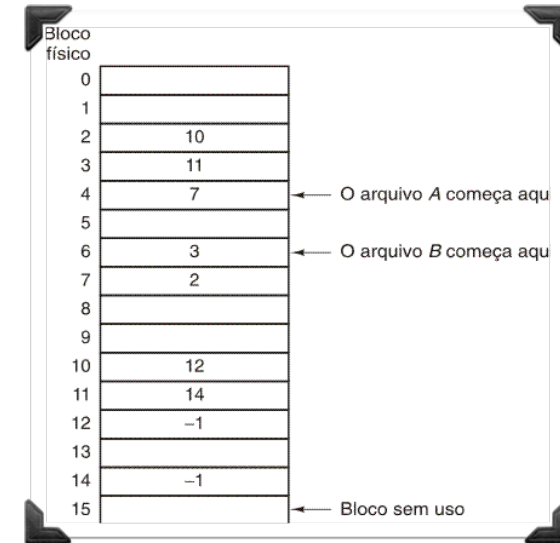
Implementação de Arquivos



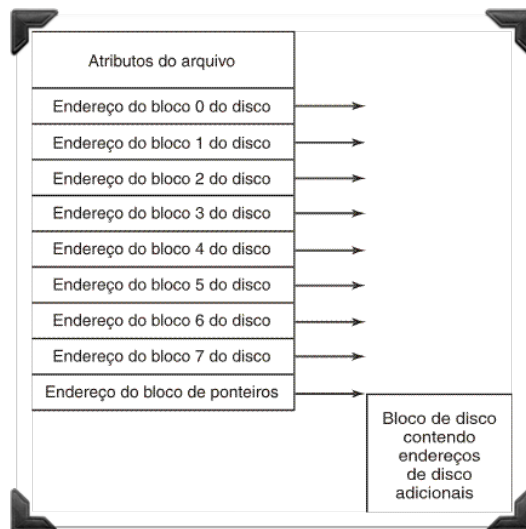
Implementação de Arquivos



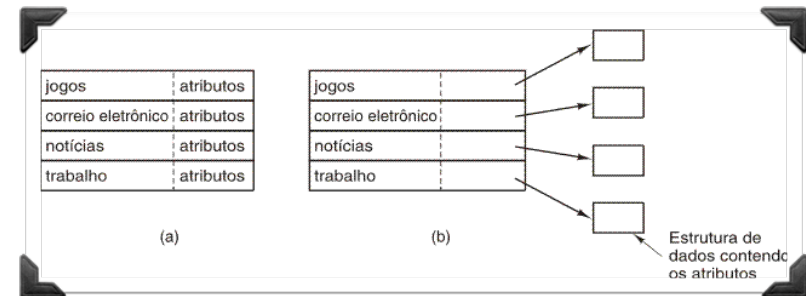
Implementação de Arquivos



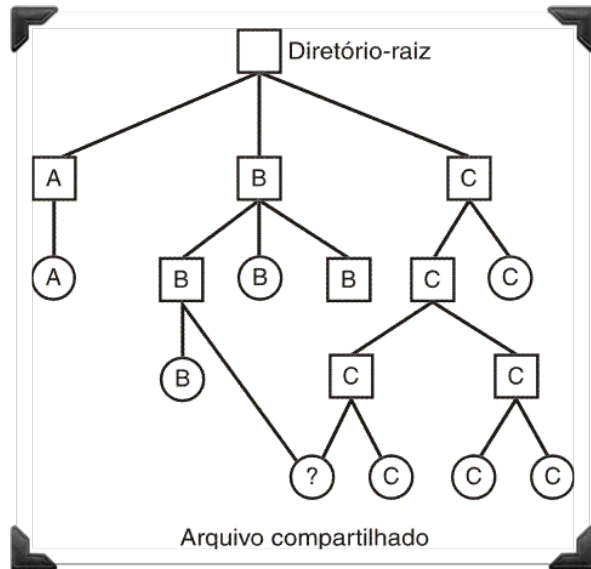
Implementação de Arquivos



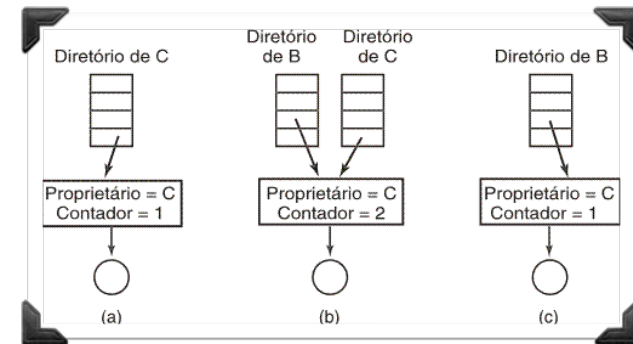
Implementação de Diretórios



Elos para Arquivos



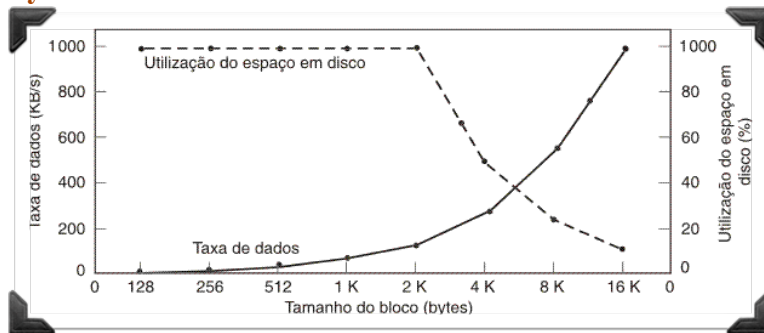
Elos para Arquivos



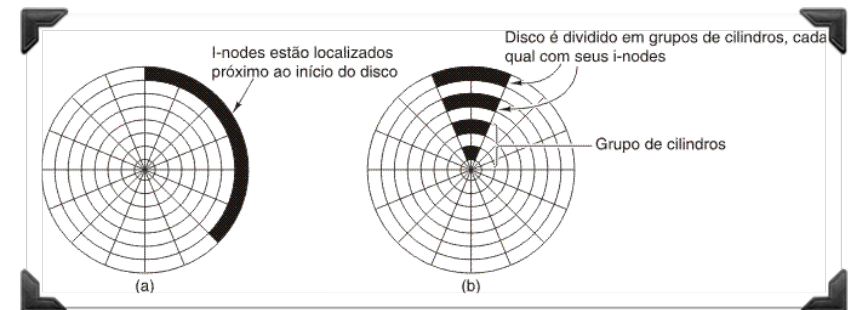
Gerenciamento de Espaço em Disco

Tamanho do bloco (cluster)

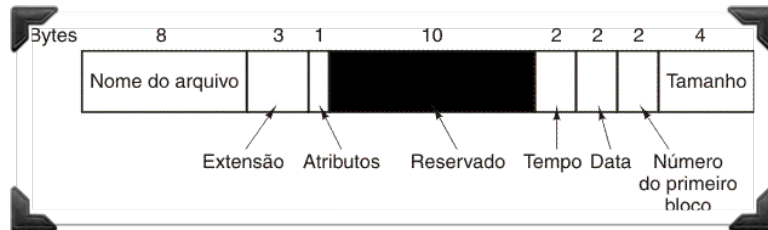
- Considere um disco com 131072 bytes/trilha; tempo de rotação de 8,33ms; tempo médio de posicionamento de 10ms. Qual o tempo médio de leitura de um bloco qualquer de n bytes?



Desempenho



MS-DOS

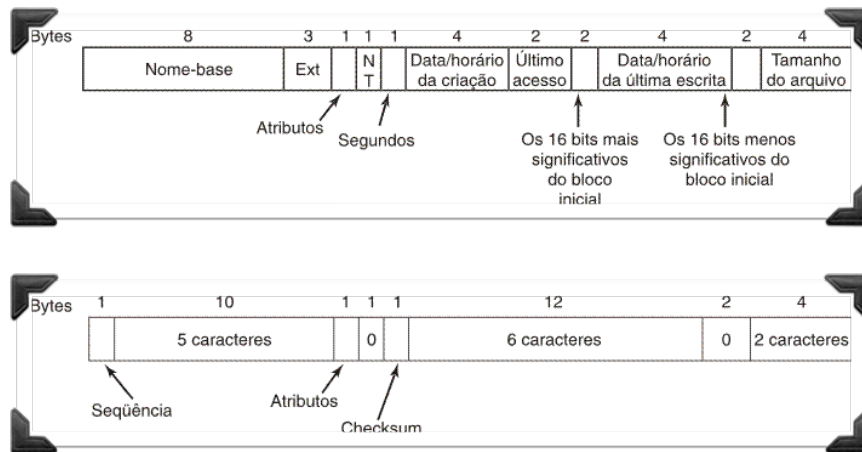


MS-DOS

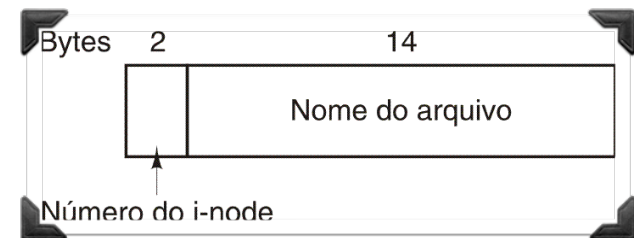
Tamanho do bloco	FAT-12	FAT-16	FAT-32
0,5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB



Windows 98



UNIX v7



UNIX v7

Diretório-raiz	Procurar usr resulta no i-node 6	Bloco 132 é o diretório /usr	I-node 26 é para /usr/ast	Bloco 406 é o diretório /usr/ast
1 .	Modo tamanho datas/horários	6 *	Modo tamanho momentos	26 *
1 ..		1 **		6 **
4 bin		19 dick		64 grants
7 dev		30 erik		92 books
14 lib		51 jim		60 mbox
9 etc	132	26 ast	406	81 minix
6 usr		45 bal		17 src
8 tmp				
I-node 6 é para /usr 6	I-node 6 que /usr está no bloco 1322	/usr/ast está no i-node 26	I-node 26 /usr/ast está no bloco 406	/usr/ast/mbox está no i-node 60