

Systemy operacyjne

System plików

[2] System plików

Zarządzanie plikami:

- zarządzanie plikami zadaniem systemu operacyjnego,
- dane wejściowe procesów pobierane z plików,
- dane wyjściowe zachowywane w plikach,
- nieulotność plików.

Pojęcia dotyczące plików:

- **pole**, podstawowa jednostka danych, zawiera pojedynczą wartość charakteryzującą się rozmiarem i typem danych,
- **rekord**, kolekcja powiązanych pól traktowana jako pewna całość,
- **plik**, kolekcja podobnych rekordów traktowana jako pewna całość, identyfikowana przez unikatową nazwę z dostępem ograniczonym zadanymi prawami dostępu.

[3] Zarządzanie plikami

System zarządzania plikami:

- udostępnia wyłączną metodę dostępu do plików,
- zwalnia z konieczności implementacji własnych metod dostępu.

Zadania systemu zarządzania plikami:

- gwarantowanie prawidłowości przechowywania danych w plikach,
- optymalizacja wydajności,
- oferowanie metod zarządzania danymi spełniających wymagania użytkowników,
- minimalizacja/ eliminacja potencjalnych możliwości uszkodzenia bądź utracenia danych,

- udostępnianie obsługi wejścia/ wyjścia dla wielu użytkowników,
- udostępnianie obsługi wejścia/wyjścia dla jak największej klasy urządzeń do przechowywania danych.

[4] Wymagania wobec systemu plików

- można tworzyć i modyfikować pliki,
- można mieć kontrolowany dostęp do plików innych użytkowników,
- można kontrolować dostęp do własnych plików,
- można zmieniać strukturę własnych plików na potrzeby aplikacji,
- można przemieszczać dane między plikami,
- można przechowywać i odtwarzać dane w przypadku uszkodzenia,
- można odwoływać się do plików poprzez nazwy symboliczne.

[5] Funkcje systemu plików

- identyfikacja i lokalizacja wybranych plików,
- wykorzystanie katalogów w celu opisania lokalizacji i atrybutów poszczególnych plików,
- zarządzanie dostępem użytkowników do plików,
- blokowanie na czas dostępu do plików,
- zarządzanie alokacją wolnych bloków dla plików,
- zarządzanie przestrzenią wolnych bloków.

Kryteria organizacji plików:

- szybki dostęp,
- łatwość aktualizacji,
- efektywność przechowywania,
- łatwość zarządzania,
- niezawodność.

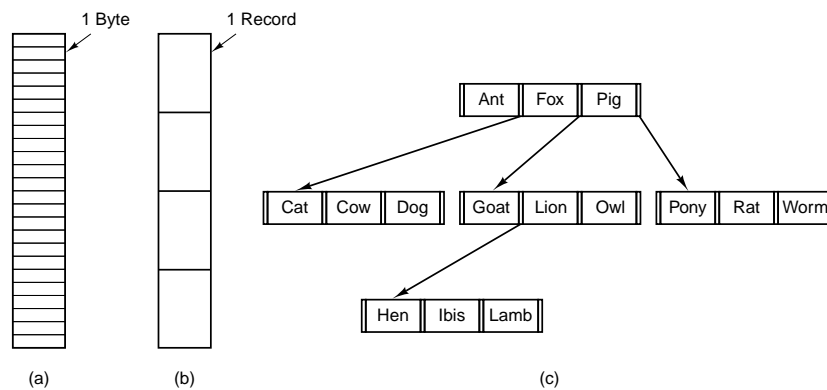
[6] Rodzaje organizacji plików (I)

- sterta,
- plik sekwencyjny,
- plik indeksowany,
- plik z kodowaniem mieszającym.

Istotne aspekty wpływające na przyjmowaną organizację wewnętrzną:

- możliwość rozszerzania,
- wydajność wstawiania,
- wydajność wyszukiwania,
- metody wyszukiwania,
- koszty potencjalnych reorganizacji,
- wymagana redundancja danych.

[7] Rodzaje organizacji plików (II)



Trzy rodzaje plików:

- a. sekwencja bajtów,
- b. sekwencja rekordów,

c. plik o strukturze drzewiastej.

[8] Przykładowe atrybuty plików

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

[9] Operacje na plikach (I)

```

/* File copy program. Error checking and reporting is minimal. */

#include <sys/types.h>           /* include necessary header files */
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

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

#define BUF_SIZE 4096           /* use a buffer size of 4096 bytes */
#define OUTPUT_MODE 0700       /* protection bits for output file */

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

    if (argc != 3) exit(1);      /* syntax error if argc is not 3 */

    /* Open the input file and create the output file */

```

[10] Operacje na plikach (II)

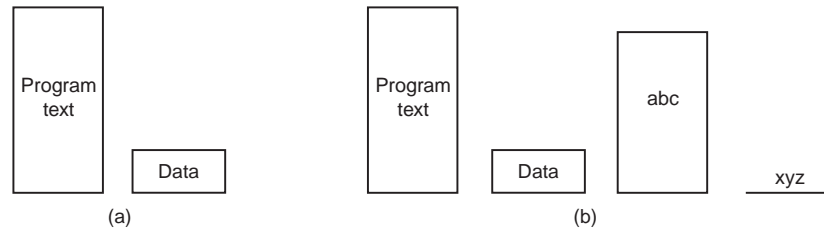
```

    in_fd = open(argv[1], O_RDONLY); /* open the source file */
    if (in_fd < 0) exit(2);          /* if it cannot be opened, exit */
    out_fd = creat(argv[2], OUTPUT_MODE); /* create the destination file */
    if (out_fd < 0) exit(3);        /* if it cannot be created, exit */

    /* Copy loop */
    while (TRUE) {
        rd_count = read(in_fd, buffer, BUF_SIZE); /* read a block of data */
        if (rd_count <= 0) break;                /* if end of file or error, exit loop */
        wt_count = write(out_fd, buffer, rd_count); /* write data */
        if (wt_count <= 0) exit(4);              /* wt_count <= 0 is an error */
    }

    /* Close the files */
    close(in_fd);
    close(out_fd);
    if (rd_count == 0) /* no error on last read */
        exit(0);
    else
        exit(5);      /* error on last read */
}

```

[11] Pliki odwzorowywane w pamięci

Postać segmentów procesu:

- a. przed odwzorowaniem pliku w jego przestrzeń adresową.
- b. po odwzorowaniu istniejącego pliku *abc* w segment i utworzeniu nowego segmentu dla pliku *xyz*.

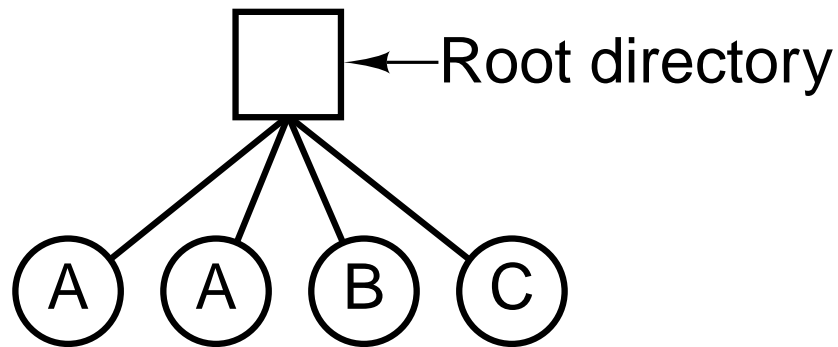
[12] Organizacja plików w katalogi

- katalogi zawierają informacje o plikach: atrybuty, lokalizacje, id właściciela,
- katalog sam w sobie może być plikiem (np. w systemie Unix),
- głównym przeznaczeniem katalogów jest udostępnianie odwzorowania między nazwami plików a samymi plikami.

Struktury organizacji katalogów:

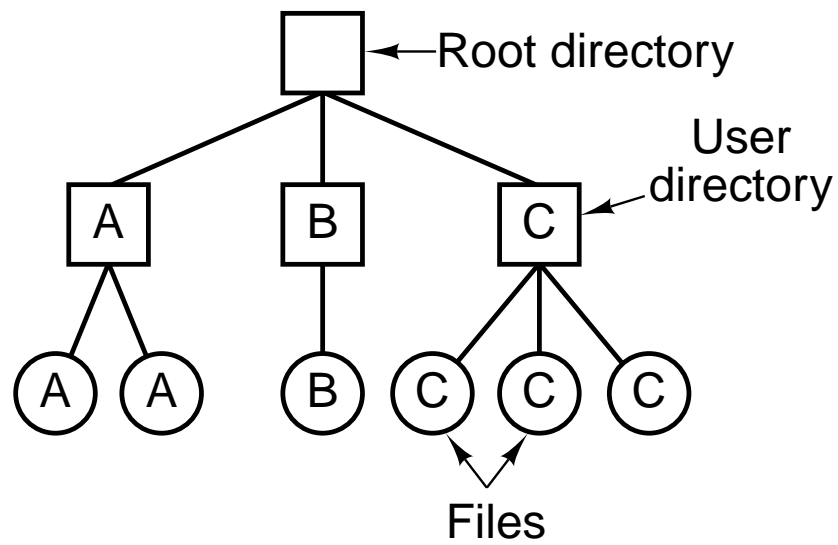
- struktura prosta,
- struktura dwupoziomowa, katalog główny + po katalogu dla każdego użytkownika,
- struktura hierarchiczna,
 - pliki identyfikowane przez ścieżkę,
 - możliwych wiele nazw dla tego samego pliku,
 - pojęcie katalogu bieżącego i odwołania relatywne wobec katalogu bieżącego.

[13] Systemy z jednopoziomowym katalogowaniem

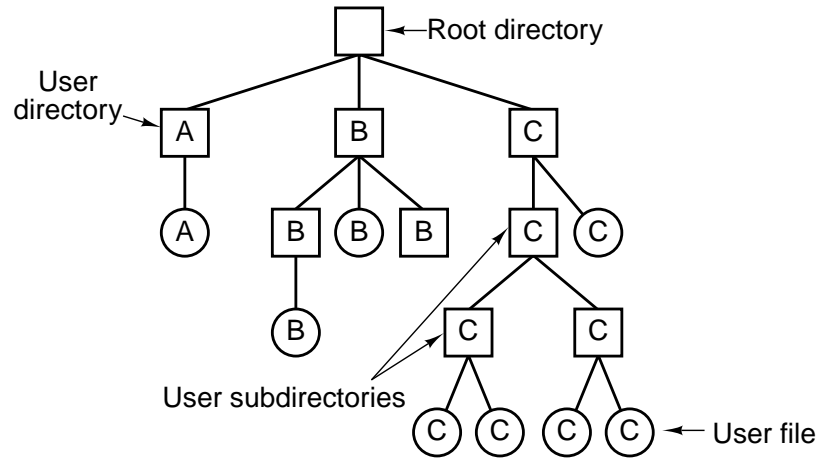


System z jednopoziomym katalogowaniem zawierający cztery pliki, we władaniu trzech różnych osób.

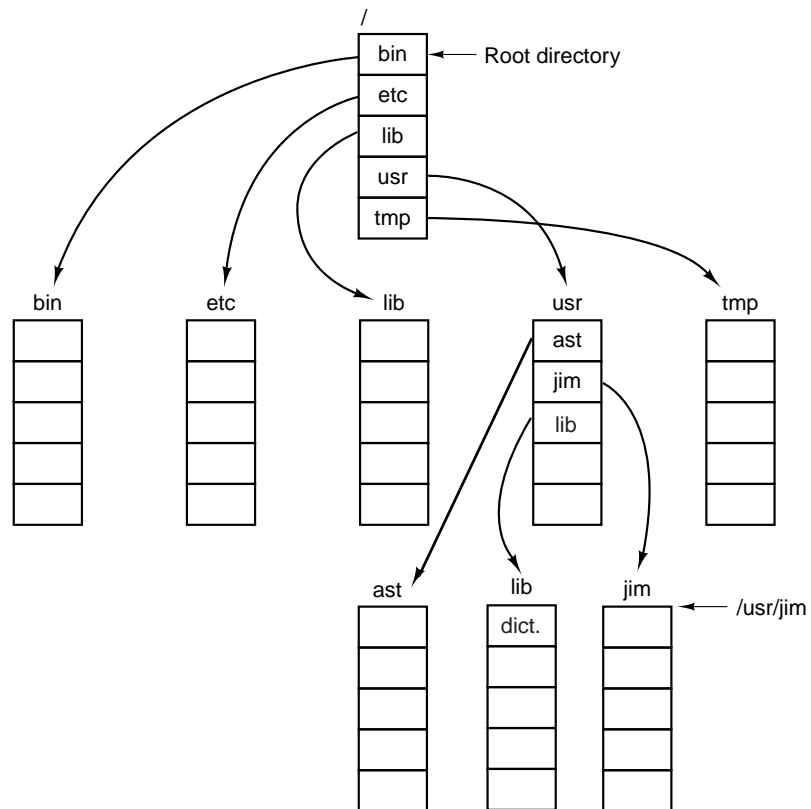
[14] Systemy z dwupoziomym katalogowaniem



[15] Systemy z hierarchią katalogów



[16] Drzewo katalogów w systemie Unix

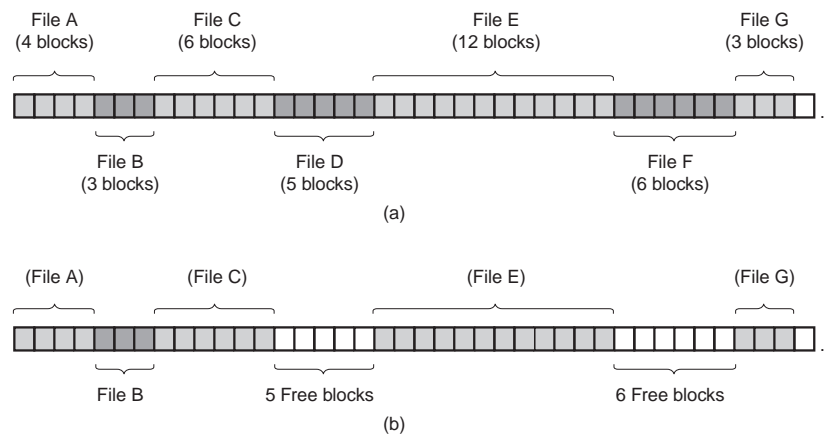


[17] Zarządzanie pamięcią pomocniczą

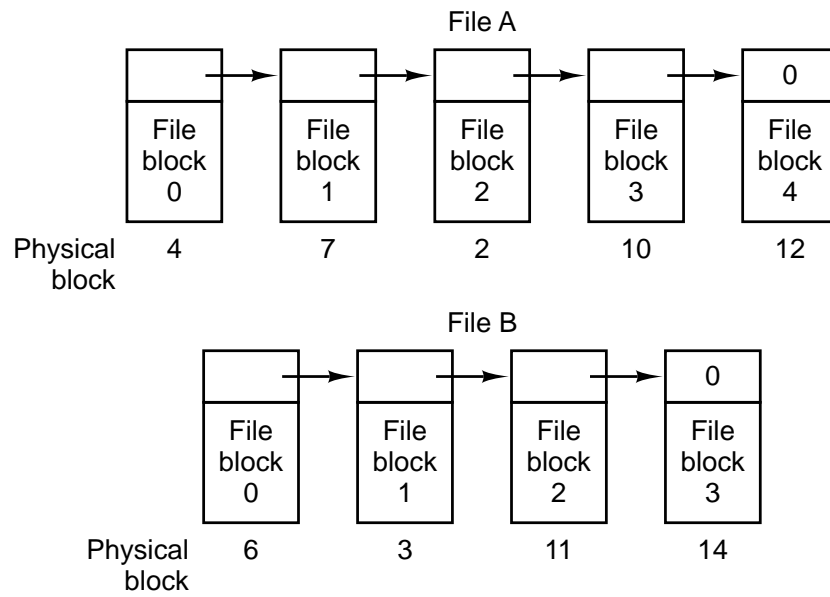
Metody alokacji plików:

- alokacja ciągła,
 - pozycja FAT = nazwa, blok startowy, rozmiar,
- alokacja łańcuchowa,
 - pozycja FAT = nazwa, blok startowy, rozmiar,
 - w każdym bloku pole z numerem kolejnego bloku.
- alokacja indeksowa,
 - pozycja FAT = nazwa, nr bloku z indeksami,
 - blok z indeksami zawiera indeksy bloków z danymi.
 - możliwe rozszerzenie o wprowadzenie obszarów z lokalną ciągłością, pozycja bloku z indeksami = nr bloku danych, liczba bloków,

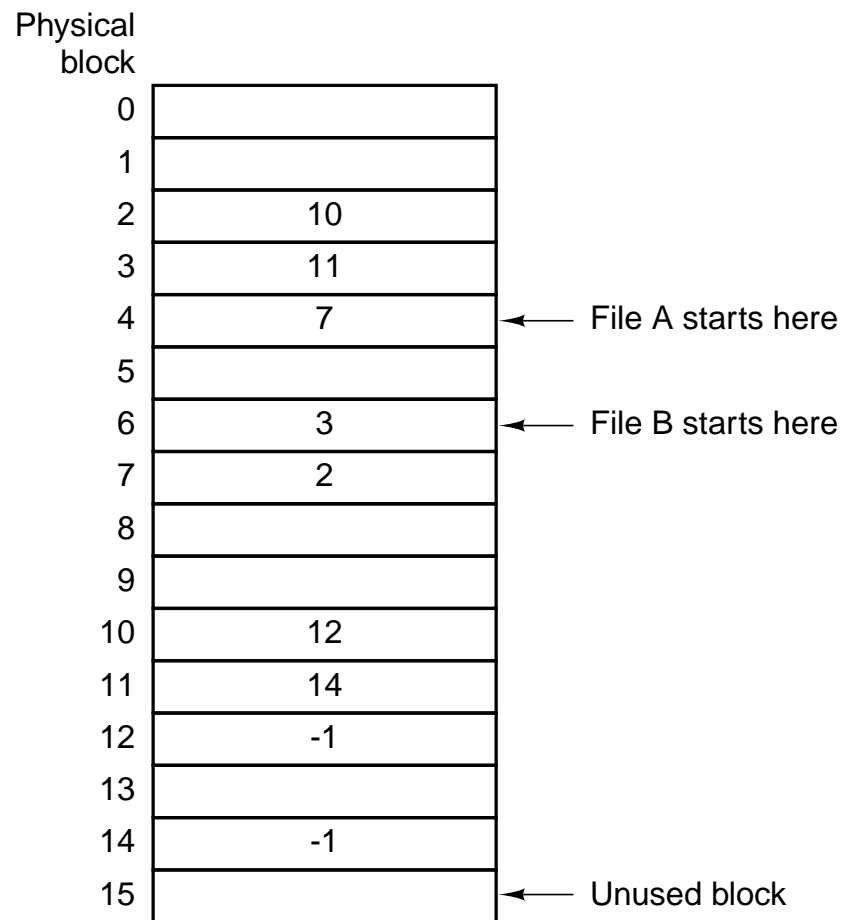
[18] Alokacja ciągła plików



[19] Alokacja z listą połączeń (I)

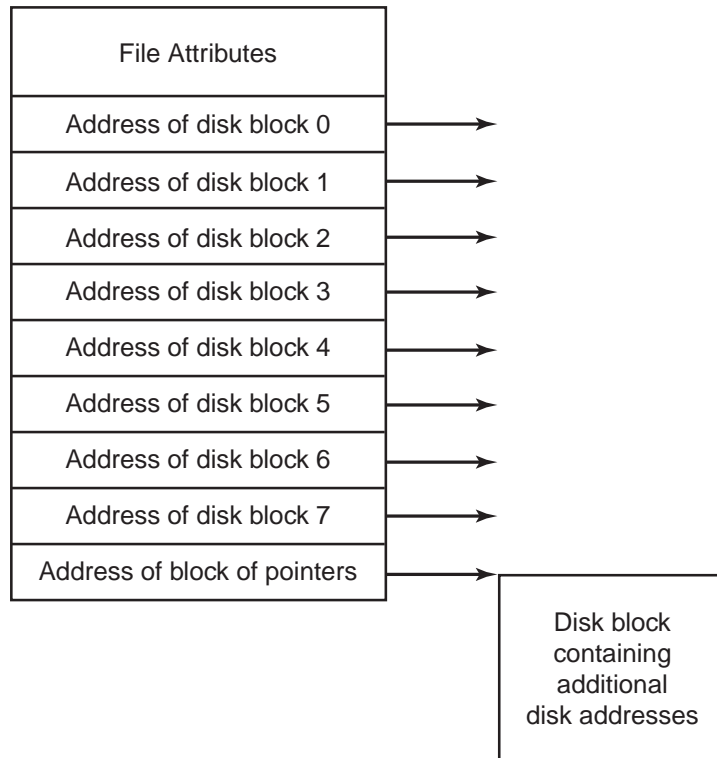


[20] **Alokacja z listą połączeń (II)**

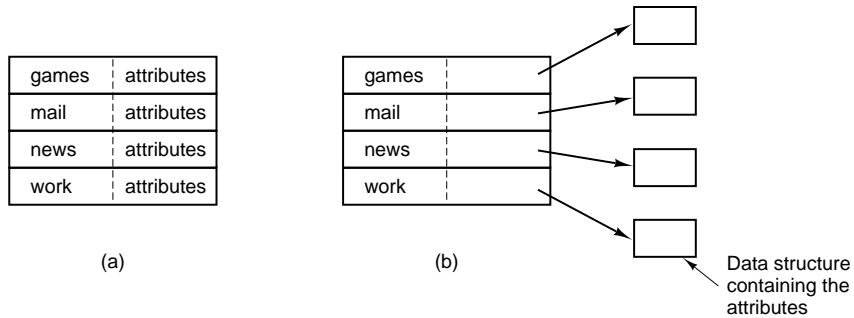


Alokacja z listą połączeń wykorzystująca tablicę alokacji plików w pamięci głównej.

[21] Przykładowy i-węzeł



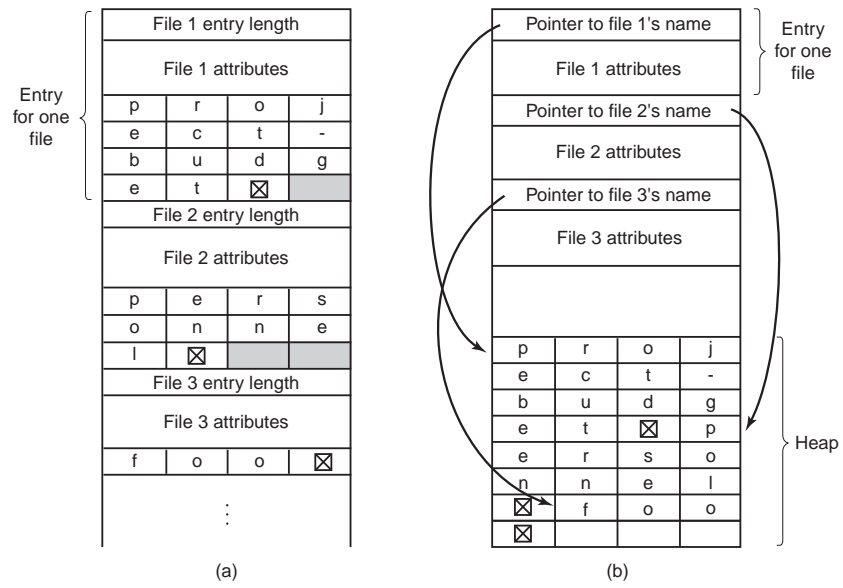
[22] Implementacja katalogów (I)



Przechowywanie atrybutów pozycji katalogowych:

- a. prosty katalog zawierający elementy o stałym rozmiarze z adresami dyskowymi oraz atrybutami w samej pozycji katalogowej,
- b. katalog, w którym każdy element wskazuje na i-węzeł.

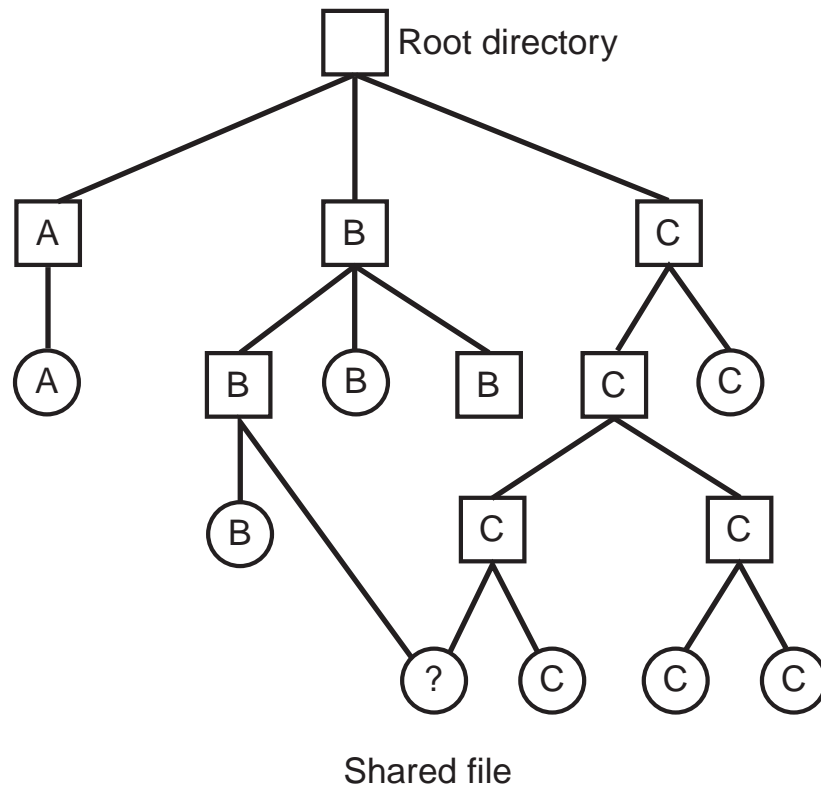
[23] Obsługa długich nazw plików



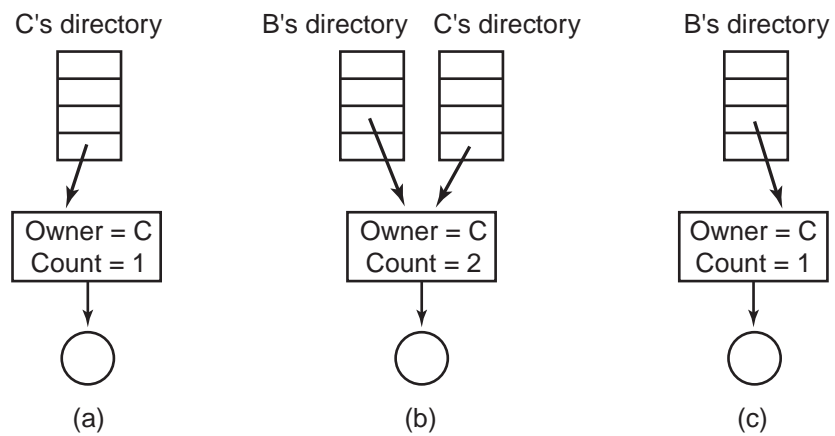
Metody obsługi długich nazw plików w katalogu:

- a. obsługa wewnętrzna (ang. *in-line*),
- b. obsługa poprzez dodatkową stertę.

[24] **Pliki współdzielone**



[25] Ukierunkowane grafy acykliczne

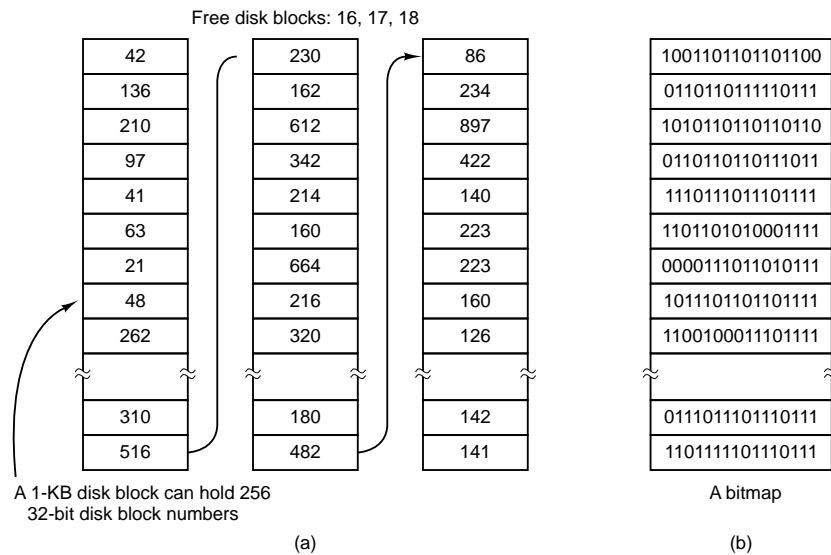


Odsyłacze do pliku:

- a. przed stworzeniem odsyłacza,
- b. po stworzeniu odsyłacza,

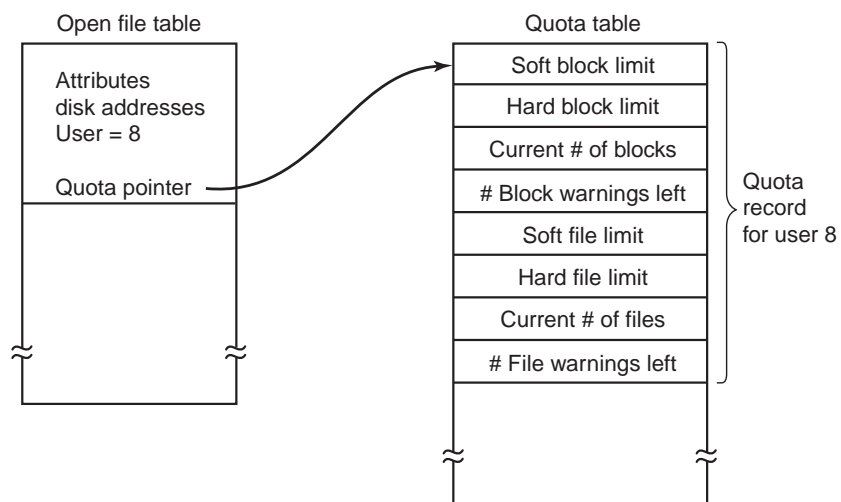
c. po usunięciu oryginalnego odsyłacza.

[26] Zarządzanie listą wolnych bloków



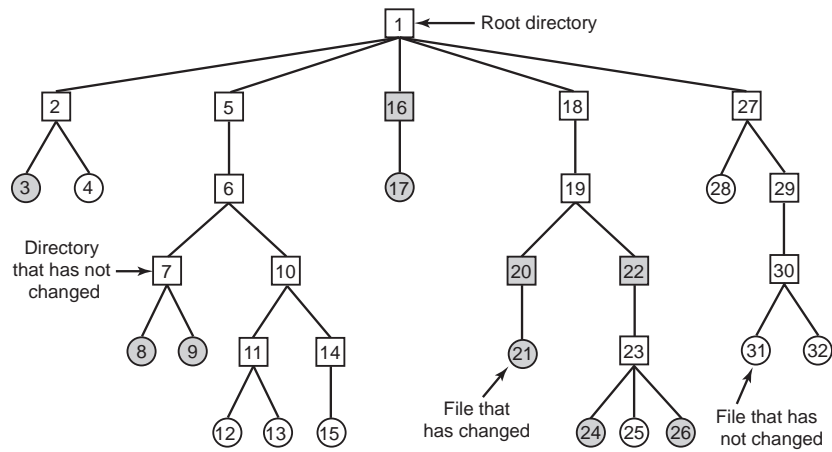
- a. zarządzanie wolnymi blokami w postaci listy,
- b. zarządzanie wolnymi blokami z wykorzystaniem mapy bitowej.

[27] Quota dyskowa



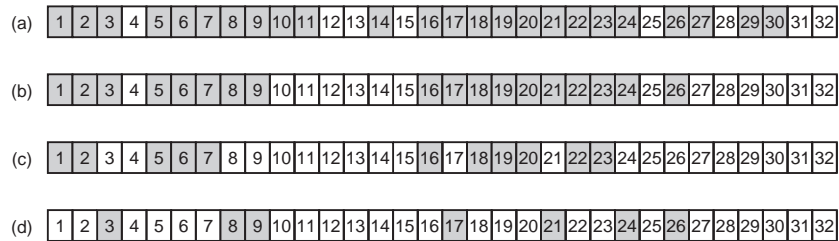
Quota kontrolowane dla poszczególnych użytkowników w tablicach quota.

[28] Archiwizacja plików

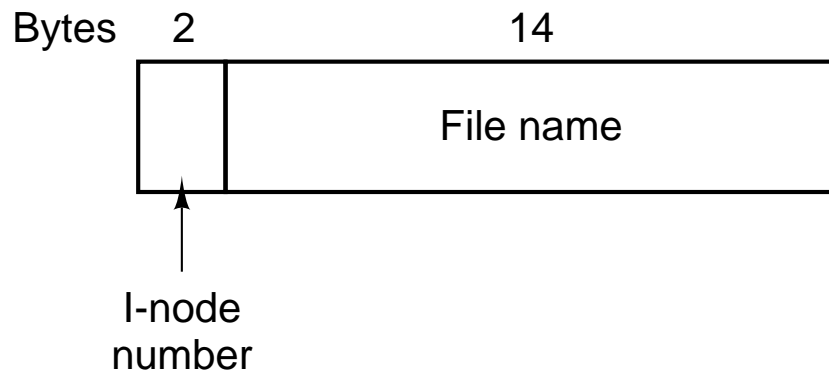


- każdy katalog i plik etykietowany odpowiadającym numerem i-węzła,
- obiekty zaciemnione modyfikowane od czasu ostatniej archiwizacji.

[29] Wykorzystanie map bitowych do archiwizacji

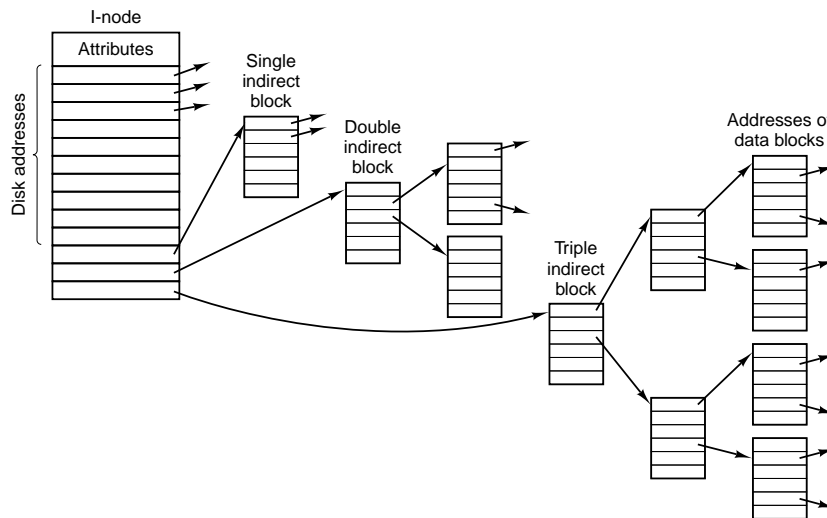


[30] System plików w Unix V7 (I)



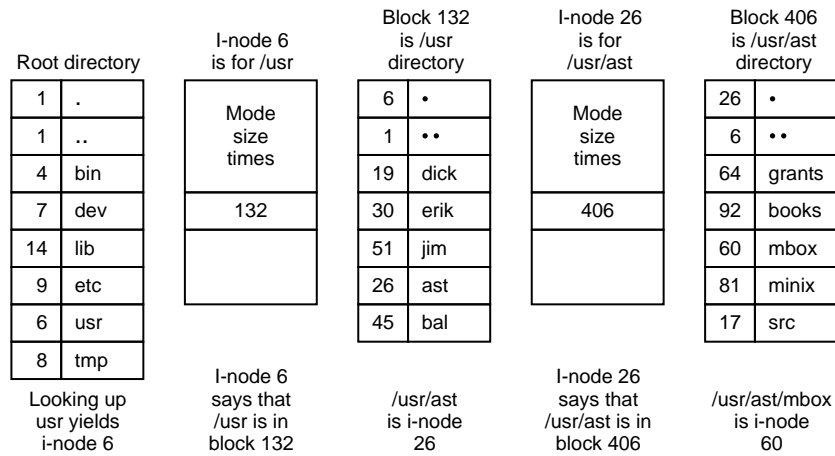
Pozycja katalogowa w systemie Unix V7

[31] System plików w Unix V7 (II)



I-węzeł w systemie Unix V7.

[32] System plików w Unix V7 (III)



Przykład rozwiązywania nazwy /usr/ast/mbox.

[33] System plików w Unix V7 (IV)

W systemie Unix system plików jest zapisywany na dysku na ogół w postaci czterech sekcji:

- blok 0 (**boot block**), wykorzystywany do ładowania przy starcie systemu,
- blok 1, (**superblock**), zawiera informacje o strukturze systemu plików,
- bloki 2 – m, (**i-nodes**), listy **i-węzłów**,
- bloki m+1 – n, bloki na dane.

I-węzły a pliki specjalne:

- w przypadku plików specjalnych w i-węzłach w miejsce pierwszego wskaźnika na blok danych wpisywany jest numer programu obsługi urządzenia,
- numer programu obsługi urządzenia składa się z dwóch części, numerów major number i minor number,
- do utworzenia pliku specjalnego stosuje się komendę **mknod**.

[34] System plików w Unix V7 (V)

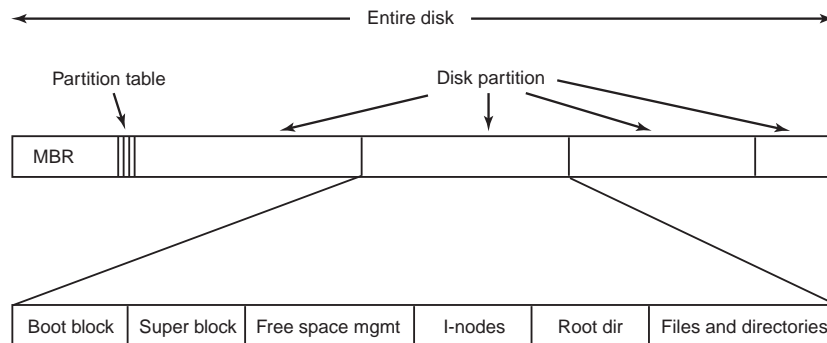
Informacje zawarte w bloku superblock:

- rozmiar w blokach listy i-węzłów,
- rozmiar w blokach systemu plików,
- liczba wolnych bloków w systemie plików,
- indeks następnego wolnego bloku na liście wolnych bloków,
- liczba wolnych i-węzłów w systemie plików,
- indeks następnego wolnego i-węzła na liście wolnych i-węzłów,
- pole blokad dla listy wolnych bloków i listy wolnych i-węzłów,
- znacznik modyfikacji bloku superblock,
- czas modyfikacji i nazwa systemu plików.

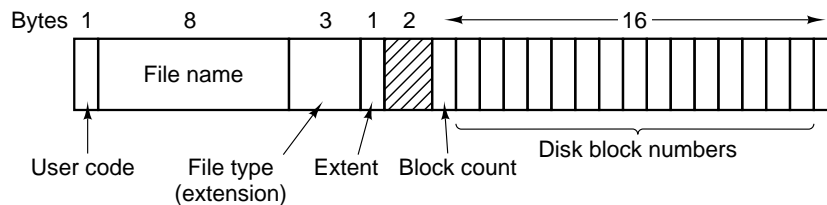
W nowoczesnych systemach często dodatkowe własności, takie jak:

- wiele kopii bloku superblock,
- dzienniki transakcyjne (ang. *journaling*),
- możliwość wykonywania migawek (ang. *snapshot*).

[35] Przykładowa struktura systemu plików

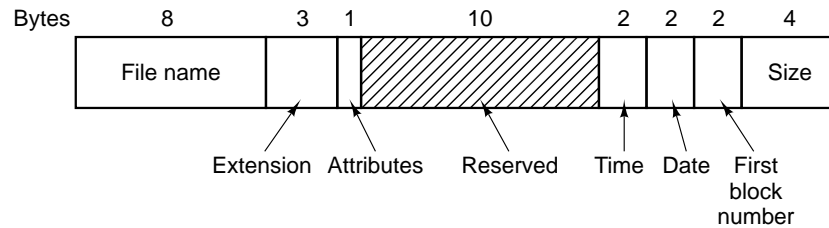


[36] System plików CP/M



Pozycja katalogowa w systemie CP/M.

[37] System plików w MS-DOS (I)



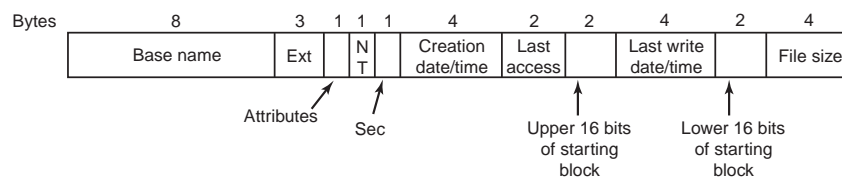
Pozycja katalogowa w systemie CP/M.

[38] System plików w MS-DOS (II)

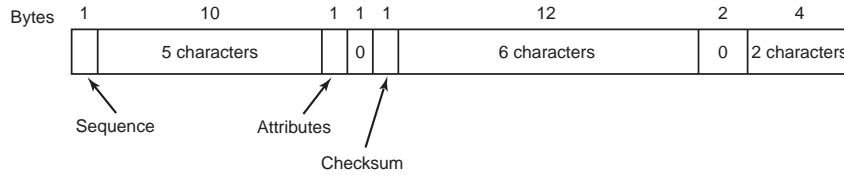
Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

Maksymalny rozmiar partycji dla różnych rozmiarów bloku.

[39] System plików w Windows 98 (I)



Rozszerzona postać pozycji katalogowej MS-DOS w systemie Windows 98.



Element stanowiący część opisu długiej nazwy w systemie Windows 98.

[40] System plików w Windows 98 (II)

68	d	o	g	A	0	C	K		0								
3	o	v	e	A	0	C	K	t	h	e	l	a	0	z	y		
2	w	n	f	o	A	0	C	K	x	j	u	m	p	0	s		
1	T	h	e	q	A	0	C	K	u	i	c	k	b	0	r	o	
Bytes	T	H	E	Q	U	I	~	1	A	N	S	Creation	Last	Upp	Last	Low	Size
												time	acc		write		

Przykład postaci przechowywania długiej nazwy w systemie Windows 98.