Strumienie i pliki.

Ostatnia aktualizacja: 29 września 2026

- **Plik** - ciąg bajtów o skończonej długości
- Nawa pliku nie stanowi jego zawartości, jest elementem systemu plików
- Lokalizacja pliku jest określona przez **ścieżkę dostępu**
- Pliki są opatrzone **atrybutami**: uprawnienia, własności pliku, np. plik ukryty, itp.
- Plik tekstowy a plik binarny - w obu wypadkach to jest ciąg bajtów

http://pl.wikipedia.org/wiki/Plik

Format pliku: określa rodzaj i sposób zapisu danych w pliku.

- Rozszerzenia plików (DOS, Winows)
  `txt html c csv bmp mp3 jpg`
- Metadane zawarte w pliku, sygnatura formatu zakodowana najczęściej na początku pliku: nagłówek pliku, liczba magiczna
  `FF D8 FF` dla formatu `jpg`
  `D0 CF 11 E0` dokumenty MS Office
- Metadane zewnętrzne, np. umieszczone w systemie plików.
  Typy **MIME**, Multipurpose Internet Mail Extensions
  `Content-Type: text/plain`
  `Content-Type: audio/mpeg:`

http://en.wikipedia.org/wiki/List_of_file_signatures
http://en.wikipedia.org/wiki/File_format

Organizacja plików w komputerze

System plików: określa sposób przechowywania plików na nośniku fizycznym i organizację dostępu do nich z systemu operacyjnego

- fizyczny zapis danych, np. blokowa struktura danych na dysku (sektory, klastry)
- logiczna struktura widoczna dla użytkownika udostępniana przez system operacyjny
- przykłady systemów plików:
  - DOS/Windows: FAT, FAT32, NTFS
  - Linux: ext2, ext3, ext4
  - inne: HFS (Mac OS), NFS (sieć), ISO9660 (CD-ROM)
- hierarchia plików: katalogi (foldery), podkatalogi i pliki.
  Ścieżka dostępu od korzenia do pliku
  `C:\Documents and Settings\user\moje dokumenty\plik.txt`
  `/home/user/doc/plik.txt`

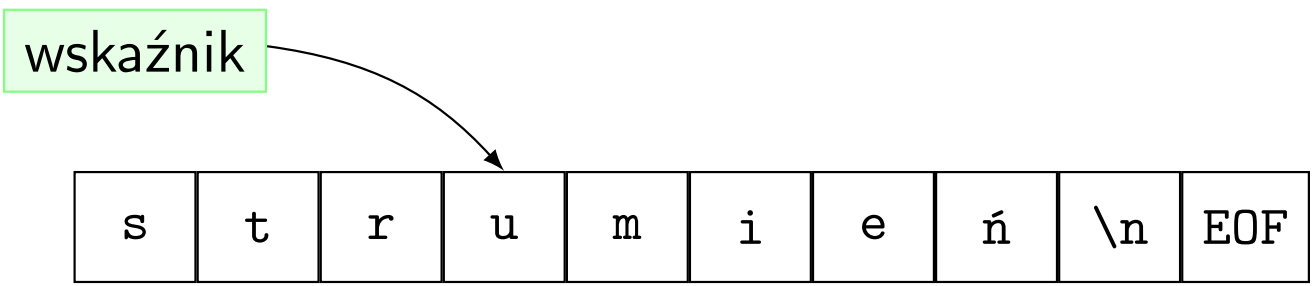
http://en.wikipedia.org/wiki/List_of_file_systems

Atrybuty plików

- nazwa pliku
- rozmiar
- data utworzenia, data modyfikacji, dostępu
- uprawnienia: właściciel, grupy i ich prawa (odczyt/zapis)
- DOS/Windows: ukryty, tylko do odczytu, archiwalny, systemowy, zaszyfrowany (NTFS), skompresowany (NTFS)
- rodzaje plików: plik zwykły, katalog, dowiązanie, FIFO, blokowe, ...
- położenie pliku, wskaźnik do miejsca na nośniku (często tablica wskaźników FAT), Linux i-węzły

- **Deskryptor pliku**: niskopoziomowa reprezentacja (liczba całkowita) wykorzystywana przez funkcje systemowe (np. biblioteka `unistd.h` w systemach Unix/Linux)
- **Strumień:** ogólny sposób komunikacji między procesami i urządzeniami (obsługa wejścia/wyjścia)
- Wykorzystanie strumieni: dostęp do plików (odczyt/zapis), komunikacja między procesami (np. potoki), komunikacja sieciowa, komunikacja z urządzeniami, łańcuchy znakowe jako strumienie.
- Plik nagłówkowy `stdio.h` zawiera deklaracje funkcji obsługi wejścia i wyjścia (***St**andar**d** **i**nput **o**utput*)
- Strumień do/z pliku dostępny za pomocą zmiennej typu `FILE*` (wskaźnik do pliku, „uchwyt" do pliku)

- Typ `FILE*` z biblioteki `stdio.h`
- Struktura zawierająca informacje o pliku: nazwa, deskryptor pliku, rodzaj dostępu, znacznik pozycji, adres bufora, ...
- Strumień udostępnia dane jako sekwencję bajtów zakończoną wartością końca pliku `EOF`
- Sekwencyjny odczyt i zapis, automatyczne buforowanie
- Wskaźnik bieżącej pozycji - miejsce odczytu/zapisu
- Obsługa błędów i końca pliku



1. Deklaracja zmiennej typu `FILE*`

```
FILE* plik;
```

2. Uzyskanie dostępu do strumienia `fopen()`

```
plik = fopen("plik.txt", "w");
```

3. Odczyt, zapis lub zmiana pozycji w strumieniu, np.:

```
fprintf(plik, "Hello world!\n");
fscanf(plik2, "%d", &x);
```

4. Zamknięcie strumienia `fclose()`

```
fclose(plik);
```

Nie zapomnij o obsłudze błędów.
Każda operacja na pliku może zakończyć się niepowodzeniem.

```
FILE *fopen(char *ścieżka, char *tryb);
```

- ścieżka do pliku:
  - względem bieżącego katalogu `"dane.xml"`, `"../plik.txt"`
  - bezwzględna: `"C:\\plik.txt"`, `"/home/user/plik"`.

  Ścieżki bezwzględne w kodzie to nie jest najlepszy pomysł - brak przenośności!
- tryb otwarcia:
  - `"w"` zapis (*write*), plik powstaje od początku
  - `"r"` odczyt (*read*) pliku istniejącego
  - `"a"` dopisanie (*append*) na końcu pliku istniejącego
- tryb binarny: `"rb"`, `"wb"`, `"ab"`

```
FILE *odczyt, *zapis;
odczyt = fopen("plik1.txt", "r");
zapis = fopen("plik2.txt", "w");
```

Błędy przy otwieraniu strumienia

- W przypadku niepowodzenia `fopen()` zwraca wartość `NULL`
- Wówczas nie ma możliwości wykonania jakiejkolwiek operacji na strumieniu
- Zawsze sprawdzaj czy udało się uzyskać dostęp do strumienia

```
FILE *plik;
plik = fopen("plik.txt","r");
if ( plik != NULL ) { /*...*/ }
```

- Możliwe przyczyny niepowodzenia: zła nazwa pliku, zła ścieżka, brak nośnika, uszkodzenie nośnika, brak uprawnień do odczytu lub zapisu, za duża liczba otworzonych strumieni, itp...

Formatowany zapis i odczyt

```
int fprintf(FILE *plik, char *format, ...);
int fscanf(FILE *plik, char *format, ...);
```

Zapis i odczyt pojedynczego bajtu (znaku)

```
int fgetc(FILE *plik);
int fputc(int c, FILE *plik);
```

Odczyt łańcucha

```
char *fgets(char *tablica, int rozmiar, FILE *strumien);
```

- Wartość zwracana może informować o błędach lub końcu pliku
- Odczyt i zapis strumieni binarnych: `fread`, `fwrite`
- Więcej informacji w dokumentacji biblioteki `stdio.h`

Typowy schemat: zapis do pliku

```c
#include <stdio.h>

int main()
{
    FILE *plik = NULL;
    float pi = 3.1415;

    plik = fopen("plik.txt", "w");
    if (plik != NULL)
    {
        /* Tutaj operacje na pliku */
        fprintf(plik, "Witaj swiecie!\nPI=%f\n", pi);
        fclose(plik);
    }
    else
    {
        /* Obsluga bledu otwarcia pliku */
        printf("Blad otwarcia pliku %s\n", "plik.txt");
    }
    return 0;
}
```

☞ zapis.c

Koniec strumienia

```
int feof(FILE* plik);
```

Wartość niezerowa gdy wystąpił koniec pliku.

Funkcje odczytu: `fscanf()`, `fgetc()`, `fgets()` zwracają w takiej sytuacji wartość `EOF` (*end of file*)

Inne błędy operacji wejścia-wyjścia

```
int ferror(FILE *stream);
```

Zwraca wartość niezerową, gdy w strumieniu wystąpił błąd

Typowy schemat: odczyt znaków z pliku

```c
#include<stdio.h>

int main()
{
    FILE *plik = NULL;
    int znak;

    plik = fopen( "plik.txt", "r");
    if (plik == NULL)
    {
        perror("Wystapil blad");
        return 1;
    }

    while (feof(plik) == 0)
    {
        znak = fgetc(plik);
        if (znak != EOF) printf("%c\n", znak);
    }
    fclose(plik);

    return 0;
}
```

☞ odczyt.c

Standardowe wejście i wyjście programu

```
cat < plik1.txt > plik2.txt
```

< przekierowanie strumienia wejściowego

> przekierowanie strumienia wyjściowego

Potoki

```
ls | sort | head | wc
```

Strumienie to podstawowy sposób komunikacji miedzy procesami.

Strumienie `FILE*` dostępne w `stdio.h`

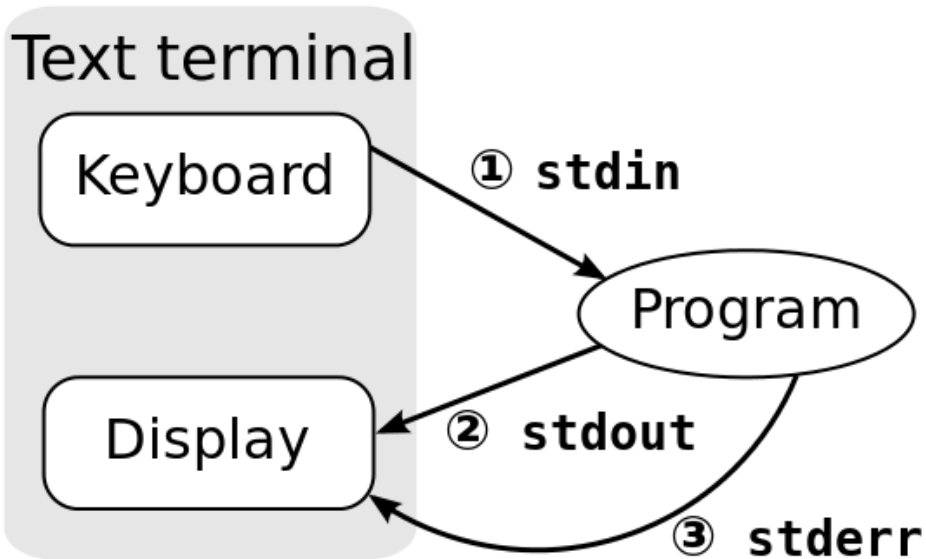
0 `stdin`
standardowe wejście, domyślnie klawiatura
`getchar()`, `scanf()`, `...`

1 `stdout`
standardowe wyjście, domyślnie ekran
`putchar()`, `printf()`, `...`

2 `stderr`
standardowe wejście diagnostyczne, domyślnie ekran
`perror()`, `...`

```
z=getchar();
putchar(z);
printf("x=%d\n",x);
scanf("%f", &y);
gets(tab);
```

```
z=fgetc(stdin);
fputc(z, stdout);
fprintf(stdout, "x=%d\n",x);
fscanf(stdin, "%f", &y);
fgets(tab, n , stdin);
```



Źródło: http://upload.wikimedia.org/

- DOS/Windows: konwersja plików tekstowych
  Nowy wiersz: `\r\n` $\Longleftrightarrow$ `\n`
- Niepoprawny format danych dla `scanf()`, `fscanf()`
  Wartość zwracana to ilość wczytanych elementów.

```
if(scanf("%d", &x) > 0 )  {  /* OK */ }
```

- Ostrożnie z mieszaniem odczytu formatowanego z nieformatowanym

```
scanf("%d", &x);
getchar();      /* wczyta co zostawil scanf() */
```

- Jednoczesny zapis i odczyt może wymagać dodatkowych zabiegów (np. czyszczenie bufora)
- W Unix/Linux wielkość liter w ścieżkach ma znaczenie
  `plik.txt` `PLIK.TXT` `Plik.txt` `Plik.TXT`
- W razie niepewności zajrzyj do dokumentacji

Literatura dodatkowa

- Wikipedia: ☞Plik, ☞List of file signatures, ☞File format. ☞System plików, ☞List of file systems, ☞Comparison of file systems, ☞Standardowe strumienie
- The GNU C Library: ☞Input/Output Overview, ☞Input/Output on Streams