

Informatyka 2 (EZ1E3012)

Politechnika Białostocka - Wydział Elektryczny
Elektrotechnika, semestr III, studia niestacjonarne I stopnia
Rok akademicki 2022/2023

Pracownia nr 8 (17/18.12.2022)

dr inż. Jarosław Forenc

Przetwarzanie plików

- Operacje związane z przetwarzaniem pliku zazwyczaj składają się z trzech części

1. Otwarcie pliku (strumienia):

- funkcje: `fopen()`

2. Operacje na pliku (strumieniu), np. czytanie, pisanie:

- funkcje dla plików tekstowych: `fprintf()`, `fscanf()`, `fgetc()`,
`fputc()`, `fgets()`, `fputs()`...

- funkcje dla plików binarnych: `fread()`, `fwrite()`, ...

3. Zamknięcie pliku (strumienia):

- funkcja: `fclose()`

Otwarcie i zamknięcie pliku

```
#include <stdio.h>

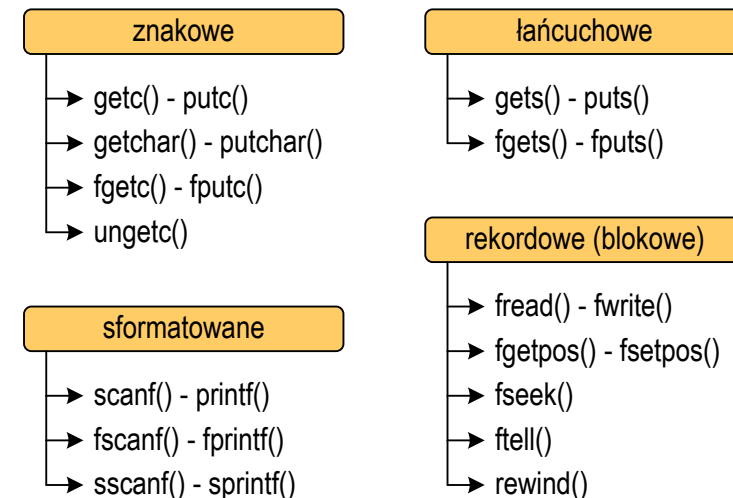
int main(void)
{
    FILE *fp;

    fp = fopen("plik.txt", "w");
    if (fp == NULL)
    {
        printf("Bład otwarcia pliku.\n");
        return (-1);
    }

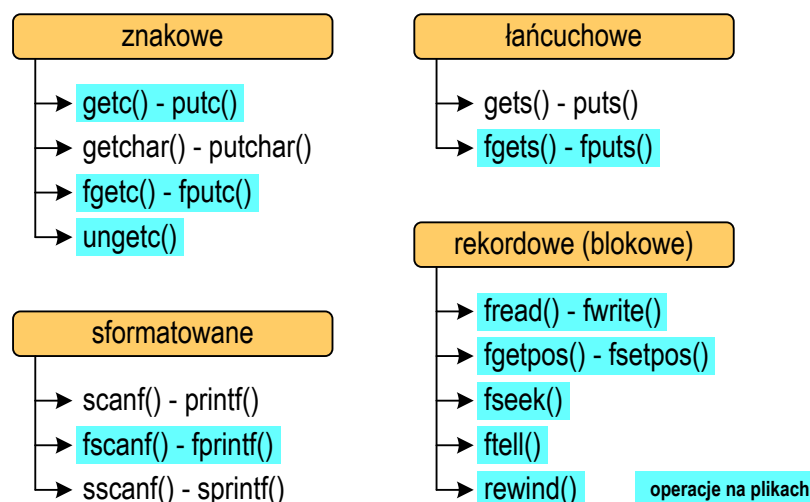
    /* przetwarzanie pliku */

    fclose(fp);
    return 0;
}
```

Typy standardowych operacji wejścia-wyjścia



Typy standardowych operacji wejścia-wyjścia (pliki)



Wykrycie końca pliku tekstowego

Funkcje	Metoda
getc(), fgetc()	zwracana wartość: EOF
fgets()	zwracana wartość: NULL
fscanf()	wywołanie funkcji: feof()

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

- zwraca wartość różną od zera, jeśli podczas ostatniej operacji odczytu pliku wskazywanego przez `fp` został wykryty jego koniec; w przeciwnym razie zwraca wartość `0` (zero)

Plik tekstowy i binarny

- Przykład zawartości pliku tekstowego (**Notatnik**):

Plik (ang. file) – uporządkowany zbiór danych o skończonej długości, posiadający szereg atrybutów i stanowiący dla użytkownika systemu operacyjnego całość. Nazwa pliku nie jest częścią tego pliku, lecz jest przechowywana w systemie plików.

- dane w pliku tekstowym zapisane są w postaci kodów ASCII

- Przykład zawartości pliku binarnego (**Notatnik**):

- dane w pliku binarnym zapisane są w takiej samej postaci jak w pamięci komputera

Operacje na plikach binarnych

```
size_t fwrite(const void *p, size_t s, size_t n, FILE *fp);
```

- zapisuje `n` elementów o rozmiarze `s` bajtów każdy, do pliku określanego przez `fp`, biorąc dane z obszaru pamięci wskazywanego przez `p`
- zwraca liczbę faktycznie zapisanych elementów

```
size_t fread(void *p, size_t s, size_t n, FILE *fp);
```

- pobiera `n` elementów o rozmiarze `s` bajtów każdy, z pliku określanego przez `fp` i umieszcza odczytane dane w obszarze pamięci wskazywanym przez `p`
- zwraca liczbę faktycznie odczytanych elementów

Zapisanie danych do pliku binarnego

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int x = 10, tab[5] = {1,2,3,4,5};
    float y = 1.2345f;

    fp = fopen("dane.dat", "wb");
    fwrite(&x, sizeof(int), 1, fp);
    fwrite(tab, sizeof(int), 5, fp);
    fwrite(tab, sizeof(tab), 1, fp);
    fwrite(&y, sizeof(float), 1, fp);
    fclose(fp);

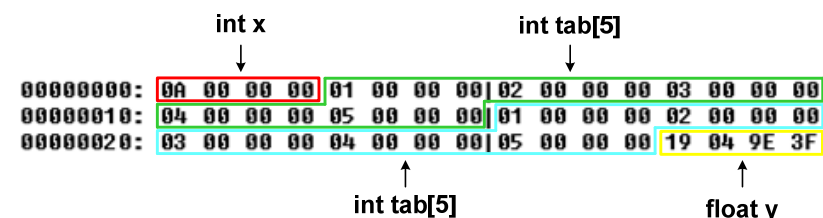
    return 0;
}
```

Zapisanie danych do pliku binarnego

- Czterokrotne wywołanie funkcji `fwrite()`

```
fwrite(&x, sizeof(int), 1, fp); // int x = 10;
fwrite(tab, sizeof(int), 5, fp); // int tab[5] = {1,2,3,4,5};
fwrite(tab, sizeof(tab), 1, fp); // int tab[5] = {1,2,3,4,5};
fwrite(&y, sizeof(float), 1, fp); // float y = 1.2345;
```

spowoduje zapisanie do pliku 48 bajtów:



Odczytanie liczb z pliku binarnego

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    int x, ile = 0;

    fp = fopen("liczby.dat", "rb");
    fread(&x, sizeof(int), 1, fp);
    while (!feof(fp))
    {
        ile++; printf("%d\n", x);
        fread(&x, sizeof(int), 1, fp);
    }
    fclose(fp);
    printf("Odczytano: %d liczb\n", ile);
    return 0;
}
```

```
37
31
83
27
6
62
31
50
Odczytano: 8 liczb
```

Schemat przetwarzania pliku

- Typowy schemat odczytywania danych z pliku

```
fread(&x, sizeof(int), 1, fp); ← odczytanie danych z pliku
while (!feof(fp)) ← sprawdzenie czy koniec pliku
{
    ile++; printf("%d\n", x); ← przetworzenie danych
    fread(&x, sizeof(int), 1, fp); ← odczytanie danych z pliku
}
```

Przykład: odczytanie liczb z pliku binarnego

- Po otwarciu pliku wskaźnik pozycji pliku pokazuje na jego początek

↓
25 00 00 00 1F 00 00 00|53 00 00 00 1B 00 00 00 | %XXXXXXXXSXXXXXXXX
06 00 00 00 3E 00 00 00|1F 00 00 00 32 00 00 00 | #####>XXXXXXXX2###

- Po odczytaniu jednej liczby: `fread(&x,sizeof(int),1,plik);`
wskaźnik jest automatycznie przesuwany o `sizeof(int)` bajtów

↓
25 00 00 00 1F 00 00 00|53 00 00 00 1B 00 00 00 | %XXXXXXXXSXXXXXXXX
06 00 00 00 3E 00 00 00|1F 00 00 00 32 00 00 00 | #####>XXXXXXXX2###

- Po odczytaniu kolejnej liczby: `fread(&x,sizeof(int),1,plik);`
wskaźnik jest ponownie przesuwany o `sizeof(int)` bajtów

↓
25 00 00 00 1F 00 00 00|53 00 00 00 1B 00 00 00 | %XXXXXXXXSXXXXXXXX
06 00 00 00 3E 00 00 00|1F 00 00 00 32 00 00 00 | #####>XXXXXXXX2###