

Przechowywanie danych czyli pliki w języku C++

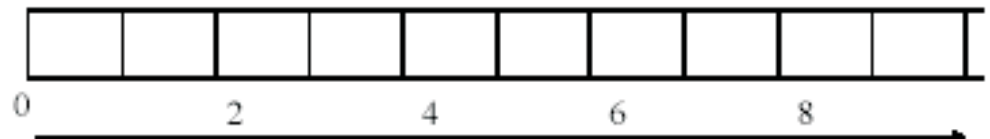
mgr inż. Piotr Kaczmarek

`Piotr.Kaczmarek@put.poznan.pl`

Politechnika Poznańska, Instytut Automatyki i Inżynierii Informatycznej

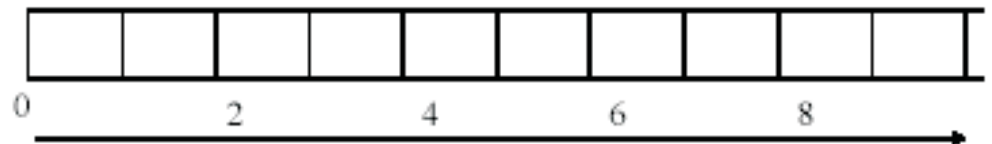
Plik a kaseeta magnetofonowa

- ⑥ Plik służy do przechowywania danych na dysku lub innym nośniku.
- ⑥ Ma określony początek i koniec
- ⑥ Dane ułożone są jedna po drugiej
- ⑥ Odczyt pliku realizowany jest tak jak odczyt kasety magnetofonowej (odtworzenie pewnego fragmentu powoduje przewinięcie kasety) a tym samym ponowne odtworzenie spowoduje przeczytanie/odtworzenie nowego fragmentu,



Plik a kasetą magnetofonową cd...

- ⑥ W pliku, w odróżnieniu od tablicy zapisując dane nie określa się ich położenia (zostaną one po prostu dograne). W operacjach zapisu/odczytu określa się wyłącznie ilość danych do zapisu/ odczytu
- ⑥ Aby odczytać pewien fragment pliku należy najpierw ustawić marker aktualnego położenia na początek fragmentu który chcemy przeczytać/zapisać a następnie odczytać (read) lub zapisać (write) dane które nas interesują.



Pliki tekstowe vs binarne

- ⑥ Brak różnicy dla tekstu
- ⑥ istnieją istotne różnice w formie przechowywania liczb zależne od sposobu kodowania
- ⑥ W plikach binarnych położenie kolejnej danej określane jest na podstawie rozmiaru poprzedniej
- ⑥ W plikach tekstowych dane rozdzielone są znakami przecinkami, średnikami, spacjami, tabulatorami itp.

Reprezentacja liczb w pliku tekstowym

6 Przechować liczby: 12345,128,12346

△ 1 = 49 = 0x31 (w ASCII)

△ 2 = 50 = 0x32 (w ASCII)

△ ...

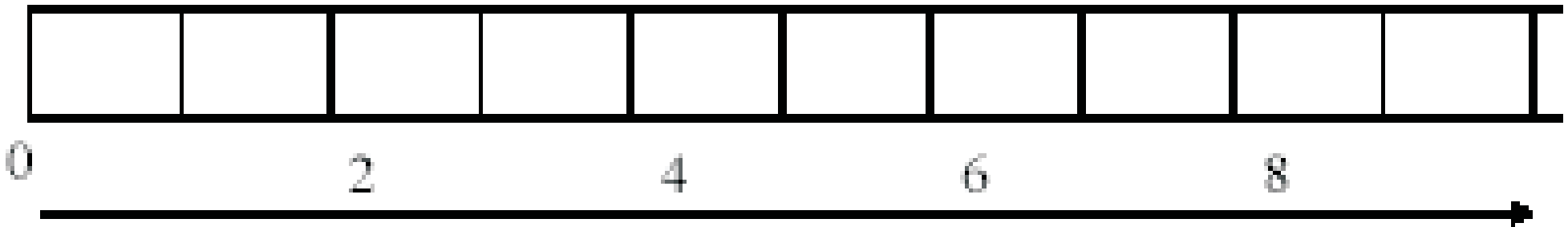
△ , = 44 = 0x2C (w ASCII)

△ ...



Reprezentacja liczb w pliku binarnym

- ⑥ Przechować liczby: 12345, 128, 12346
 - △ $12345 = 0x3039$
 - △ $128 = 0x0080$
 - △ $12346 = 0x303A$



Dostęp do pliku

- ⑥ Trzy kroki które muszą zostać wykonane przy dostępie do pliku:
 1. Otworzyć plik
 2. Odczyt lub/i zapis
 3. Zamknąć plik
- ⑥ Przy tradycyjnym sposobie obsługi plików krok 1 i 3 są takie same niezależnie od tego czy mamy do czynienia z plikiem tekstowym czy binarnym

Otwarcie pliku - `fopen`

1. Zadeklarować wskaźnik do pliku *handler*
2. Wywołać funkcję `fopen()` Funkcja przekazuje wskaźnik do otwartego strumienia pliku.

Przykład:

```
FILE *strumien;  
strumien = fopen("Plik.txt", "wt");
```


Otwieranie pliku - argumenty (1)

```
fopen(arg1, arg2
```

- ⑥ arg1- Nazwa pliku zawierająca nazwę i ew. ścieżkę dostępu: "c:\\ mojkatalog\\ Plik.txt". Nazwa ta może zostać podana bezpośrednio lub przez zmienną przechowującą tablicę znaków np
tt char[.]

Otwieranie pliku - argumenty (2)

```
fopen(arg1, arg2)
```

⑥ arg2 - tryb dostępu:

- △ `wt` Otwarcie pliku ASCII do zapisu (plik jest tworzony od nowa),
- △ `rt` Otwarcie pliku ASCII do odczytu,
- △ `at` Otwarcie pliku ASCII do dołączenia (zapisu na końcu pliku)
- △ `wb`, `rb`, `ab` Jak wyżej tylko dla plików binarnych

Otwieranie pliku - argumenty (3)

Ponadto

- ⑥ $r+t$, $r+b$ otwarcie pliku do zapisu i odczytu (strumień jest ustawiany na początku pliku)
- ⑥ $w+t$, $w+b$ plik jest tworzony od nowa (jeśli istnieje to czyszczony i otwierany do odczytu i zapisu)
- ⑥ $a+t$, $a+b$ Tak samo jak dla $r+$ z tym, że strumień ustawiany jest na końcu

Dostęp do pliku binarnego

Zapis/odczyt realizowany jest za pomocą funkcji `fwrite`, `fread`. Funkcje dokonują zapisu/odczytu wyspecyfikowanej ilości bajtów począwszy od aktualnej pozycji w strumieniu.

Funkcje `fread` i `fwrite` mają taką samą listę argumentów:

```
fwrite(void * buffer, long size, long count, FILE * stream );
```

- ⑥ `buffer` - adres początku pamięci spod której kopiowane/zapisywane będą dane
- ⑥ `size` - rozmiar pojedynczego bloku pamięci (elementu)
- ⑥ `count` - ilość elementów do skopiowania
- ⑥ `stream` - wskaźnik do strumienia pliku

Dostęp do pliku binarnego

Przykład:

```
FILE *strumien ;
strumien = fopen("Plik_bin.bin", "w+b");
int zmienna1=5;
long zmienna2=8;
long zmienna3=0xCCCCCC;
int tab[]={10,16,8,1};
char napis[]="zwykly_tekst";
fwrite(&zmienna1, sizeof(int), 1, strumien);
fwrite(&zmienna2, sizeof(long), 1, strumien);
fwrite(&zmienna3, sizeof(long), 1, strumien);
fwrite(&tab, sizeof(int), 4, strumien); // zapis tablicy
fwrite(napis, sizeof(char), strlen(napis), strumien);
fclose(strumien);
```

Przemieszczanie w pliku

6 Aktualne położenie w pliku:

```
FILE *strumien;  
strumien = fopen("Plik_bin.bin", "w+b");  
int pos = ftell(strumien);
```

6 przesunięcie położenia w strumieniu do pozycji względem początku, aktualnej pozycji lub końca strumienia:

```
fseek(strumien, offset, seek_dir);  
np:      fseek(strumien, 100, SEEK_SET);
```

Przemieszczanie w pliku

- ⑥ gdzie seek dir może wynosić:
 - △ SEEK_SET - położenie względem początku pliku
 - △ SEEK_CURR - położenie względem aktualnego położenia
 - △ SEEK_END - położenie względem końca pliku

Obsługa plików tekstowych - zapis

Do obsługi plików stosuje się funkcje analogiczne jak printf i scanf, z tym, że zawierają one wskaźnik do strumienia, z którego dane będą odczytywane czy też zapisywane.

```
FILE *strumien;  
strumien = fopen("Plik_txt.txt", "w+t");  
int z1=5;  
long z2=8;  
long z3=0xCCCCCC;  
char napis[] = "zwykly_tekst";  
printf("%d, %ld, %ld, %s", z1, z2, z3, napis);  
    \\wyświetla dane na ekranie  
fprintf(strumien, "%d, %ld, %ld, %s", z1, z2, z3, napis);  
    \\zapisuje dane do pliku tak samo jak na ekranie  
fclose(strumien);
```


Obsługa plików tekstowych - zapis

Funkcje `printf(.)` i `fprintf(.)` umożliwiają ustawienie minimalnej szerokości pola oraz precyzji (ilości miejsc po przecinku) w wyświetlaniu liczb zmiennoprzecinkowych:

```
float liczba=1.5673892;
printf ("%f*\n", liczba);
//wyświetla liczbę z domyślnymi ustawieniami
printf ("%f10.3*\n", liczba);
printf ("%f10.4*\n", liczba);
//wyświetla liczbę na polu 10znakowym
z dokładnością do 3 miejsc po przecinku
```

wynik:

```
*1.56738*
*      1.567*
*      1.5673*
```

Odczyt danych z pliku tekstowego (1)

Założmy że mamy następujący plik:

```
1,1.1  
2,14.3  
6,12.8
```

zawiera on w każdej linii dwie liczby: całkowitą oraz zmiennoprzecinkową. Do wyświetlenia pojedynczej linii na ekranie służy więc wzorzec:

```
"%d, %f"
```

Ten sam wzorzec stosuje się do odczytu przy wykorzystaniu funkcji `scanf (.)`

Odczyt danych z pliku tekstowego(2)

```
FILE *strumien;  
int liczba1;  
float liczba2;  
strumien = fopen("Dane_txt.txt", "r+t");  
fscanf(strumien, "%d,%f", &liczba1, &liczba2);  
    \\odczyt pierwszej linii  
fscanf(strumien, "%d,%f", &liczba1, &liczba2);  
    \\odczyt drugiej linii  
...  
fclose(strumien);
```

Wykorzystanie strumieni (1)

1. Tworzony jest strumień
2. Strumień dowiązywany jest do pliku
3. zapis i odczyt z pliku odbywa się za pomocą operatorów «,»
4. strumień jest zamykany

Wykorzystanie strumieni (2)

Dostęp do pliku tekstowego jest również możliwy analogicznie jak dla cin i cout.

Dla czytania i zapisu z pliku wykorzystuje się strumień typu `fstream`

```
fstream strumien_pliku("nazwa_pliku", tryb);
```

Tryb określa czy plik będzie otwierany do zapisu i odczytu:

- ⑥ `ios::in` - otwarcie do odczytu
- ⑥ `ios::out` - otwarcie do zapisu
- ⑥ `ios::app` - otwarcie do zapisu na końcu istniejącego pliku
- ⑥ `ios::in|ios::out` - otwarcie do zapisu i odczytu

Wykorzystanie strumieni (2)

```
fstream strumien("plik_str.txt", ios::out);
int z1=5;
long z2=8;
long z3=0xCCCCCC;
char napis[]="zwykly_tekst";
    // zapis
strumien<<z1<<" , "<<z2<<" , "<<z3<<" , "<<napis;
strumien.close(); //zamkniecie strumienia
```

Wykorzystanie strumieni - manipulatory (1)

Manipulatory umożliwiają modyfikowanie sposobu wyświetlania danych:

- ⑥ Zmiana formatu wyswietlania:
 - △ hex - wyświetlanie liczby w formie szesnastkowej
 - △ oct - wyświetlanie liczby w formie ósemkowej
 - △ dec - wyświetlanie liczby w formie dziesiętnej

```
int z1=193;  
strumien <<z1<<" , "<<hex<<z1<<" , "<<oct<<z1 ;
```

Wykorzystanie strumieni - manipulatory (2)

6 Precyzja

```
float a=10.56789;
strumien<<setprecision(3)<<a;
//ustawia nową precyzję wyświetlania
```

6 Szerokość pola

```
int z1=67;
strumien<<' (<<setw(10)<<a<<' ) '<<endl;
//umieszcza a w polu 10 znakowym
```

6 Wypełnienie pustych miejsc w polu

```
strumien<<' (<<setw(10)<< setfill( '#' )<<a<<' ) '<<endl;
//umieszcza 'a' w polu 10 znakowym, puste miejsca wypełn  
znakiem '#'
```


Pozycja w strumieniu

Aby uzyskać dostęp do określonej danej znajdującej się wewnątrz pliku, bez konieczności czytania wszystkich elementów poprzedzających, należy ustawić położenie w strumieniu na miejsce w którym się ona znajduje:

- ⑥ Uzyskanie informacji o aktualnym położeniu:

```
int pos=strumien.tellp();
```

- ⑥ przesunięcie położenia w strumieniu do pozycji względem początku strumienia:

```
strumien.seekp(100);
```

```
//przesuwa pozycję w strumieniu o 100 bajtów od początku
```

Pozycja w strumieniu

- ⑥ przesunięcie położenia w strumieniu do pozycji względem początku, aktualnej pozycji lub końca strumienia:

```
strumien.seekp(100, seek_dir);  
    //przesuwa pozycję w strumieniu o 100 bajtów od początku  
    końca lub pozycji aktualnej
```

- ⑥ gdzie seek dir może wynosić:
 - △ ios::beg - położenie względem początku pliku
 - △ ios::cur - położenie względem aktualnego położenia
 - △ ios::end - położenie względem końca pliku