
Politechnika Świętokrzyska

Wydział Elektrotechniki, Automatyki i Informatyki

Katedra Elektrotechniki Przemysłowej i Automatyki

Zakład Urządzeń i Systemów Automatyki

Programowanie komputerów 2

Operacje na plikach w języku C

Dostęp wysokopoziomowy

Instrukcja laboratoryjna

(wersja robocza)

Paweł Strączyński

2016

W języku C dostęp do plików możliwy jest na dwa sposoby:

- wysokopoziomowy – wykorzystujący tzw. strumienie,
- niskopoziomowy – z wykorzystaniem tzw. deskryptorów plików.

Tematem zajęć jest dostęp do plików z wykorzystaniem strumieni (sposób wysokopoziomowy).

Wysokopoziomowy dostęp do plików

Wysokopoziomowy dostęp do pliku możliwy jest z wykorzystaniem wskaźnika na zmienną plikową . Plikowy typ danych **FILE** zdefiniowany został w pliku nagłówkowym <stdio.h>.

Listing 1. Deklaracja uchwytu do pliku tekstowego

```
FILE *plik;
```

Dostęp do pliku składa się z trzech etapów:

- otwarcie strumienia (pliku) – funkcja *fopen()*
- operacje na pliku np. odczyt, zapis (*fprintf()*, *fputs()*, *fgets()*, *fgetc()* itd.)
- zamknięcie strumienia (pliku) – funkcja *fclose()*

1. Otwarcie pliku

Deklarację funkcji *fopen()* przedstawiono na listingu 2. Funkcja pobiera dwa argumenty:

- nazwę pliku lub bezwzględną ścieżkę do pliku,
- tryb otwarcia – atrybuty przedstawiono w tabeli 1.

Funkcja zwraca wskaźnik do pliku lub w przypadku gdy plik nie istnieje wartość **NULL**.

Listing 2. Deklaracja funkcji *fopen()*

```
FILE *fopen(const char *filename, const char *mode );
```

Tabela 1. Tryby otwarcia pliku

Tryb	Znaczenie
r	tryb tylko do odczytu
r+	tryb odczytu i aktualizacji (nadpisania zawartości)
w	tryb zapisu, jeśli plik istnieje kasuje jego zawartość
w+	tryb zapisu, z możliwością odczytu
a	tryb dopisywania, jeśli plik nie istnieje to jest tworzony
a+	tryb dopisywania i odczytu

Domyślnie plik otwierany jest w trybie tekstowym. Dodanie litery b do atrybutu powoduje otwarcie w trybie binarnym.

2. Odczyt pliku

Do odczytu zawartości pliku z wykorzystaniem mechanizmu wysokopoziomowego w języku C służy szereg funkcji. Funkcje te to:

- funkcja **fgetc()** - funkcja pobiera jeden znak z pliku. Jako argument pobiera wskaźnik do pliku, zwraca liczbę całkowitą (*integer*) z kodem ASCII odczytanego znaku.

Listing 3. Deklaracja funkcji fgetc()

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

- funkcja **fgets()** - funkcja pobiera znaki z pliku tekstowego i zapisuje do tablicy podanej jako parametr. Poza tablicą, oraz wskaźnikiem do pliku funkcja pobiera także ilość znaków do odczytu. Funkcja pobiera dane z pliku tekstowego do napotkania znaku nowej linii **\n** lub odczytania (n-1) znaków. Funkcja zwraca wartość **NULL** jeżeli napotka znak końca pliku lub gdy wystąpi błąd. W przeciwnym razie zwracany jest wskaźnik do bufora.

Listing 4. Deklaracja funkcji fgets()

```
char *fgets(char *buff, int n, FILE *plik);
```

- funkcja **fscanf()** - funkcje odczytuje dane z pliku zgodnie z podanym formatem analogicznie jak dla funkcji **scanf()**.

Listing 5. Deklaracja funkcji fscanf()

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

3. Zapis do pliku

Podobnie jak w przypadku odczytu, zapis do pliku również możliwy jest z wykorzystaniem kilku funkcji. Funkcje te to:

- funkcja **fputc()** - funkcja wpisuje znak dany argumentem funkcji do pliku tekstowego. Drugim argumentem funkcji jest wskaźnik na zmienną plikową. W przypadku powodzenia operacji funkcja zwraca zapisany znak. W przeciwnym wypadku zwracana **EOF**.

Listing 6. Deklaracja funkcji fputc()

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

- funkcja **fputs()** - funkcja ta zapisuje dane z bufora (tablicy znaków) do pliku. Bufor i wskaźnik do plik przekazywane są do funkcji jako argumenty. Funkcja zwraca w przypadku powodzenia ostatni wpisany plik. Jeśli operacja nie powiedzie się funkcja zwraca **EOF**.

Listing 7. Deklaracja funkcji fputs()

```
int fputs (const char *buff, FILE *plik);
```

- funkcja **fprintf()** - funkcja zapisuje dane do pliku zgodnie z zadany formatem – analogicznie do funkcji **printf()**.

Listing 8. Deklaracja funkcji fprintf()

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

4. Zamknięcie pliku

Do zamknięcia pliku tekstowego służy funkcja **fclose()**. Funkcja przyjmuje jeden parametr będący wskaźnikiem na zmienną plikową. Podczas operacji na plikach należy pamiętać aby przed zakończeniem działania programu dokonywać zamknięcia pliku. W przypadku powodzenia wykonania funkcja zwraca wartość 0. W przeciwnym wypadku zwracana jest wartość **EOF**.

Listing 9. Deklaracja funkcji fclose()

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

*UWAGA: Poza przedstawionymi wyżej funkcjami istnieje szereg innych. Warto zapoznać się między innymi z funkcjami: **fread()**, **fwrite()**, **fseek()**, **ftell()**, **feof()** oraz **ferror()**.*

Zadania do wykonania

Zadanie 1

Napisać program dokonujący losowania liczb z przedziału <0,99> i zapisujący wynik operacji do pliku. Wykorzystać metodę wysokopoziomowego dostępu do pliku.

Program powinien:

- przyjmować dwa argumenty: nazwa pliku oraz ilość losowanych liczb,
- w przypadku podania niewłaściwej ilości parametrów wyświetlać stosowny komunikat.

Dane zapisywane powinny być według podanego niżej wzoru.

Listing 2. Przykładowa zawartość pliku tekstowego

```
Liczba 1: 096  
Liczba 2: 045  
Liczba 3: 074  
Liczba 4: 088  
Liczba 5: 025  
Liczba 6: 000
```

Wskazówka:

Do losowania liczb wykorzystać funkcję **rand()**.

Zadanie 2

Zmodyfikować program z zadania 1 tak aby każdorazowe wywołanie programu dopisywało do pliku nowe wartości z zachowując ciągłość numerowania liczb jak na listingu 1.

Zadanie 3

Napisać program który odczyta zawartość pliku utworzonego w wyniku działania programu z zadania 2 i wyznaczy sumę wylosowanych liczb. Wynik powinien być drukowany na standardowe wyjście. Wykorzystać wysokopoziomowy dostęp do pliku.

Wskazówka:

Dokonując parsowania danych zwrócić uwagę na to iż linie są tej samej długości.

Zadanie 4

Zmodyfikować program z zadania 3 tak aby wyświetlał na standardowym wyjściu największą i najmniejszą wylosowaną liczbę.

Zadanie 5

Napisać program odczytując dane książek z pliku i wpisujący je do tablicy struktur. Zawartość pliku przedstawiono poniżej.

Listing 2. Zawartość pliku zawierającego dane książek

```
Kernighan B.W., Ritchie D.M.  
Język C  
WNT  
Warszawa  
1998  
  
Jones R., Stewart I.
```

Sztuka programowania w języku C

WNT

Warszawa

1992

Stroustrup B.

Jezyk C++

WNT

Warszawa 1994

Eckel B.

Thinking in C++

Helion

2002

Zdefiniować nowy typ struktury posiadający pola:

- autor,
- tytuł,
- miejsce_wydania,
- rok_wydania.

Zaimportowane dane wyświetlić na standardowym wyjściu. Do wykonywania operacji na pliku wykorzystać mechanizm wysokopoziomowy.