

Laboratorium informatyki

Programowanie w języku C

Ćwiczenie 13

1. Wprowadzenie

Program może pobierać lub wysyłać dane nie tylko na standardowe porty (klawiatura i ekran). Jednym z ważniejszych sposobów komunikacji aplikacji jest wykorzystanie plików tekstowych do przechowywania informacji. Aplikacja może pobierać dane wejściowe z pliku, a wyniki działania w nim zapisywać.

2. Dostęp do pliku

W języku C aby skorzystać z pliku zewnętrznego należy określić sposób dostępu do pliku. C umożliwia dostęp w trybie normalnym lub aktualizacji:

tryb pracy	symbol	opis
Odczyt	r	Plik musi fizycznie istnieć na dysku, możliwe jest odczytanie danych z pliku
	r+	Możliwy jest zapis i odczyt danych, plik musi istnieć
Zapis	w	Zapisywane są dane do pliku, jeżeli plik nie istnieje, zostanie utworzony, jeżeli plik zawiera dane, zostaną one skasowane.
	w+	Możliwy jest zapis i odczyt danych.
Dopisywanie	a	Zapisywane są dane do pliku, jeżeli plik nie istnieje, zostanie utworzony, jeżeli plik zawiera dane, nowe dane zostaną dopisane do pliku.
	a+	Możliwy jest zapis i odczyt danych.

Przed podłączenie pliku do programu, konieczne jest utworzenie wskaźnika do zmiennej plikowej wiążącej plik tekstowy na dysku z kodem programu:

```
FILE * nazwa_zmiennej_plikowej;
```

Podpięcie pliku realizowane jest funkcją `fopen`:

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

Dostęp do pliku jest realizowany w sekwencji działań:

a. *Otwarcie dostępu do pliku*

b. *Zapis i odczyt danych*

c. *Zamknięcie dostępu do pliku*

Dostęp do pliku zamykany jest funkcją: `fclose(nazwa_zmiennej_plikowej)`;

Przy otwieraniu dostępu do pliku warto proces ten realizować za pomocą konstrukcji warunkowej upewniając się czy dostęp został przyznany poprawnie.

```
if((nazwa_zmiennej_plikowej = fopen("plik.txt", "w")) != NULL) {}
```

3. Zapis-odczyt danych

Zapis i odczyt danych może być realizowany znak po znaku, lub poprzez bufor tekstowy.

Zapis znaku do pliku realizuje:

```
putc('znak', nazwa_zmiennej_plikowej);
```

Odczyt znaku i przypisanie do zmiennej typu char:

```
zmienna=getc(nazwa_zmiennej_plikowej);
```

Wykorzystując bufor tekstowy (tablica typu char: `char bufor[]="Tekst bufora";`) można zapisywać i odczytywać dłuższe ciągi znaków.

Odczyt danych:

```
fscanf(nazwa_zmiennej_plikowej,"%s",bufor);  
fgets(bufor,limit,nazwa_zmiennej_plikowej);
```

`limit` określa ilość znaków jaka może zostać maksymalnie przekazana do bufora

Zapis danych:

```
fprintf(nazwa_zmiennej_plikowej,"Tekst sformatowany",zmienne);  
fputs(bufor,nazwa_zmiennej_plikowej);
```

Odczytując dane z pliku istotne jest upewnienie się czy odczyt jest możliwy, czy nie doszliśmy już do końca pliku. W języku C można to zrealizować na dwa sposoby:

- Sprawdzanie czy nie odczytano znaku końca pliku tożsamego z flagą EOF.
- Funkcją detekcji końca pliku `feof(plik)`;

```
while((c = getc(plik)) != EOF)  
{}
```

```
while(!feof(plik))  
{}
```

```
c plik.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int main()
5 {
6     printf("Test zapisu i odczytu danych z pliku tekstowego.");
7     printf("Podaj nazwe pliku: ");
8     char plik[30];
9     gets(plik);
10    strcat(plik, ".txt");
11    FILE *plikTXT;
12    plikTXT = fopen(plik, "w");
13    printf("Podaj dane::>");
14    char c;
15    while((c=getchar())!='\n')
16        putc(c,plikTXT);
17    putc('\n',plikTXT);
18    fclose(plikTXT);
19    plikTXT = fopen(plik, "r");
20    printf("Z pliku '%s' odczytano tekst: ",plik);
21    while((c = getc(plikTXT))!=EOF)
22        printf("%c",c);
23    fclose(plikTXT);
24 }
```

```
Testowy.txt
1 Dane zapisane do pliku tekstowego.
2
```

```
C:\Windows\system32\cmd.exe
Test zapisu i odczytu danych z pliku tekstowego.
Podaj nazwe pliku: Testowy
Podaj dane::>Dane zapisane do pliku tekstowego.
Z pliku 'Testowy.txt' odczytano tekst: Dane zapisane do pliku tekstowego.
Press any key to continue . . .
```

4. Zadania

Napisać program umożliwiający gromadzenie i przeglądanie danych kontaktowych (imię, nazwisko, adres, telefon, email, itp.). Program przechowuje wprowadzone dane w pliku tekstowym.