

## LABORATORIUM 14. ZŁOŻONE TYPY DANYCH. PLIKI.

### Cel laboratorium:

Zaznajomienie z obsługą plików do przechowywania danych w zewnętrznej pamięci. Nabycie praktycznych umiejętności pracy z plikami tekstowymi i binarnymi.

### Zakres tematyczny zajęć:

- *pojęcie pliku,*
- *rodzaje plików,*
- *algorytm przetwarzania plików,*
- *funkcje otwarcia i zamknięcia pliku,*
- *zapis do pliku i odczyt z pliku.*

### Kompendium wiedzy:

**Plik** to wydzielony fragment pamięci (najczęściej dyskowej), posiadający nazwę (ciąg bajtów).

#### **Rodzaje plików:**

- **Tekstowe** – dane w postaci znakowej (konwersja liczb na znaki),
- **Binarne** – dane w postaci wewnętrznej reprezentacji (brak konwersji, dokładne dane).

#### **Obsługa plików:**

- Niskopoziomowa – obsługa poprzez funkcje:  
`read()`, `open()`, `write()`, `close()` – `<io.h>`
- Wysokopoziomowa – obsługa poprzez funkcje:  
`fopen()`, `fclose()`, `fread()`, `fprintf()`, ... – `<stdio.h>`

#### **Przetwarzanie plików metodą wysokopoziomą:**

1. Otwarcie pliku: `fopen()`.
2. Wykonanie operacji na pliku (zapis, odczyt, szukanie, obliczanie, ...).
  - Funkcje zapisu:  
`putc()`, `fputs()`, `fprintf()`, `fwrite()`
  - Funkcje odczytu:  
`getc()`, `fgets()`, `fscanf()`, `fread()`
  - Funkcje pomocnicze:  
`feof()`, `fseek()`, `rewind()`, `ftell()`, ...
3. Zamknięcie pliku: `fclose()`.

**Otwarcie pliku** – zwraca wskaźnik na strukturę typu `FILE` (identyfikator pliku) lub wskaźnik zerowy (`NULL`).

**Zamknięcie pliku** – zwraca 0, jeśli operacja się powiodła, lub `EOF`, jeśli nie.

```
fopen(nazwa_pliku, tryb_otwarcia); //otwarcie
fclose(wskaźnik_pliku); //zamknięcie
```



**Tryb otwarcia dla plików tekstowych:**

"r" – odczyt,  
"w" – zapis (nadpisywanie lub tworzenie),  
"a" – zapis na końcu (dopisywanie lub tworzenie),  
"r+" – odczyt i zapis,  
"w+" – odczyt i zapis (nadpisywanie lub tworzenie),  
"a+" – odczyt i zapis na końcu (dopisywanie lub tworzenie).

**Tryb otwarcia dla plików binarnych:**

"rb"; "wb"; "ab" ; "rb+"; "r+b"; "wb+" ; "w+b" ;  
"ab+"; "a+b"

```
FILE *fp;  
fp=fopen("test.txt", "w");  
fclose(fp);
```

**Zapis do pliku:**

```
putc(znak, wskaźnik_pliku);  
fputs(*tekst, wskaźnik_pliku);  
fprintf(wskaźnik_pliku, format, dane);  
fwrite(adres_w_pamięci, rozmiar_bloku, ilość_bloków,  
wskaźnik_pliku);
```

```
FILE *fp; char ch; char * slowo; int dane;  
putc(ch,fp); //zapis znaku  
fputs(slowo,fp); // zapis łańcucha znaków  
fprintf(fp,"%d",dane); //zapis z formatem
```

**Odczyt z pliku:**

```
getc(wskaźnik_pliku);  
fgets(*tekst, dlugosc, wskaźnik_pliku);  
fscanf(wskaźnik_pliku, format, dane);  
fread(adres_w_pamięci, rozmiar_bloku, ilość_bloków,  
wskaźnik_pliku);  
ilość_bloków, wskaźnik_pliku);
```

```
FILE *fp; char ch; char buf[256]; int dane;  
...  
ch= getc(fp); //odczyt znaku  
fgets(buf, 256,fp); // odczyt łańcucha znaków  
fscanf(fp,"%d",dane); //odczyt z formatem
```

**Funkcje pomocnicze:**

- Przesuwanie wskaźnika w pliku:

```
int fseek(wskaźnik_pliku, pozycja, tryb);
```

gdzie: tryb – sposób liczenia pozycji/przesunięcia (0/ 1/ 2),  
pozycja – pozycja wskaźnika w pliku uzależniona od trybu:



- o Jeśli tryb=SEEK\_SET (0) => pozycja liczona jest od początku.
- o Jeśli tryb=SEEK\_CUR (1) => pozycja jest liczona jako przesunięcie od aktualnej pozycji,
- o Jeśli tryb=SEEK\_END (2) => pozycja jest liczona jako przesunięcie od końca pliku (wskaźnik pliku jest przesuwany do pozycji <EOF> + pozycja).

```
FILE *fp;
```

```
...
```

```
fseek(fp, 10, 0); // dziesiąta pozycja, licząc od początku
```

```
fseek(fp, 10, 1); // dziesiąta pozycja, licząc od bieżącej
```

```
fseek(fp, -10, 2); // dziesiąta pozycja, licząc od końca pliku
```

- Ustawienie wskaźnika na początek pliku:

```
rewind(wskaźnik pliku);
```

```
...
```

```
rewind(fp);
```

```
...
```

- Pobranie aktualnej pozycji z pliku:

```
ftell(wskaźnik pliku);
```

```
...
```

```
fseek (fp, 0, SEEK_END);
```

```
long size=ftell (pFile);
```

```
...
```

- Informowanie o osiągnięciu/nie osiągnięciu pozycji końca pliku (prawda – koniec pliku, fałsz – brak końca pliku:

```
feof(wskaźnik pliku);
```

```
...
```

```
while(!feof(fp)) { //odczyt }
```

```
...
```

### Pytania kontrolne:

1. Co to jest plik?
2. Jakie są rodzaje plików?
3. Podaj algorytm przetwarzania plików metodą wysokopoziomową.
4. Podaj funkcje otwarcia i zamknięcia pliku.
5. Podaj funkcje zapisu danych do pliku.
6. Podaj funkcje odczytu danych z pliku.
7. Jak przesuwac wskaźnik w pliku?
8. Jak uzyskać informację gdzie znajduje się wskaźnik w pliku?

### Zadania do analizy

#### Zadanie 14.1. Imiona przyjaciół w pliku tekstowym

- Przeanalizuj przykład programu wykorzystującego pliki tekstowe.



Fundusze Europejskie  
Wiedza Edukacja Rozwój



Rzeczpospolita  
Polska

Unia Europejska  
Europejski Fundusz Społeczny



- Podaj tekst w komentarzach.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int zapisT1(char nazwa[20], FILE *fplik); //???
5  int odczytT1(char nazwa[20], FILE *fplik); //???
6  //=====
7  int main(int argc, char *argv[])
8  { char nazwa1[]="lista.txt"; //???
9    FILE *f; //???
10   int wynik;
11   wynik=zapisT1(nazwa1,f); //???
12   if(wynik==0)printf("Operacja zapisu ok\n"); //???
13   wynik=odczytT1(nazwa1,f); //???
14   if(wynik==0)printf("Operacja odczytu ok\n"); //???
15   system("PAUSE");
16   return EXIT_SUCCESS;
17 }
18 //=====
19 int zapisT1(char nazwa[20], FILE *fplik) //???
20 {char nazwisko[25]; int i=0;
21 if((fplik=fopen(nazwa, "a"))==NULL) //???
22     {printf("Blad otwarcia\n");
23     system("PAUSE"); abort();}
24 printf("Podaj nazwiska kończąc enterem\n");
25 while(gets(nazwisko)!=NULL && nazwisko[0]!='\0') //???
26 {fprintf(fplik, "%s\n", nazwisko);i++; //???
27 }
28 if(fclose(fplik)!=0){exit(2);} //???
29 printf("Do pliku zapisano %d nazwisk\n",i);
30 return 0;
31 }
32 //-----
33 int odczytT1(char nazwa[20], FILE *fplik) //???
34 {char nazwisko[25]; int i=0;
35 if((fplik=fopen(nazwa, "r"))==NULL) //???
36     {printf("blad otwarcia\n");
37     system("PAUSE"); abort();}
38 printf("\nZawartosc pliku %s\n",nazwa);
39 while(fscanf(fplik,"%s",nazwisko)==1) //???
40 {puts(nazwisko);i++; //???
41 }
42 if(fclose(fplik)!=0){exit(2);} //???
43 printf("\nZ pliku odczytano %d nazwisk\n",i);
44 return 0;
45 }
```

## Zadanie 14.2. Dane studenta w pliku binarnym

- Przeanalizuj przykład programu wykorzystującego pliki binarne.
- Podaj tekst w komentarzach.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  struct student {char nazwisko[25]; int ocena;};
4  int zapisB(char nazwa[20], FILE *fpplik); //???
5  int odczytB(char nazwa[20], FILE *fpplik); //???
6  //=====
7  int main(int argc, char *argv[])
8  { char nazwa2[]="studenci.txt";
9    FILE *f;
10   int wynik;
11   wynik=zapisB(nazwa2,f); //???
12   if(wynik==0)printf("operacja zapisu ok\n"); //???
13   wynik=odczytB(nazwa2,f); //???
14   if(wynik==0)printf("operacja odczytu ok\n"); //???
15   system("PAUSE");
16   return EXIT_SUCCESS;
17 }
18 //=====
19 int zapisB(char nazwa[20], FILE *fpplik) //???
20 {struct student st; int i; //???
21  int rozmiar=sizeof(struct student); //???
22  int licznik=1; //liczba zapisów
23  if ((fpplik=fopen(nazwa, "ab"))==NULL) //???
24      { printf ("błąd");exit(1); }
25  printf("Podaj liczbę zapisów ");
26  scanf("%d", &licznik);
27  for(i=1;i<=licznik;i++)
28      {printf("Podaj nazwisko %d: ",i);
29        scanf("%s", st.nazwisko); //???
30        printf("Podaj ocene %d: ",i);
31        scanf("%d", &st.ocena); //???
32        fwrite(&st, rozmiar, 1, fpplik); //???
33      }
34  if (fclose(fpplik) !=0) {printf ("Bład "); } //???
35  return 0;
36 }
37 //-----
38 int odczytB(char nazwa[20], FILE *fpplik) //???
39 { struct student st; //???
40  int rozmiar=sizeof(struct student); //???
41  int licznik=0;
42  if ((fpplik=fopen(nazwa, "rb"))==NULL) //???
43      { printf ("błąd");exit(1); }
44  printf("Zawartość pliku %s\n", nazwa);

```



```
45 while(fread(&st,rozmiar,1,fplik)==1) //???
46     {printf("student: %s ocena: %d\n",
47         st.nazwisko,st.ocena); //???
48         licznik++;}
49     printf("liczba pozycji: %d\n",licznik);
50     if (fclose(fplik) !=0) //???
51         {printf ("Blad przy zamykaniu pliku"); }
52 return 0;
53 }
```

### **Zadania do wykonania**

#### **Zadanie 14.3. Płaca pracownika w pliku tekstowym**

Oblicz płacę pracownika fizycznego i zapisz ją do pliku tekstowego. Zadeklaruj strukturę o polach: imię, nazwisko, liczba godzin, stawka, premia w % i do wypłaty (pole wyliczane). Napisz funkcje zapisu i odczytu z pliku. Wywołaj te funkcje kilkakrotnie. Odczytaj plik.

#### **Zadanie 14.4. Płaca pracownika w pliku binarnym**

Oblicz płacę pracownika fizycznego i zapisz ją do pliku binarnego. Zadeklaruj strukturę o polach: imię, nazwisko, liczba godzin, stawka, premia w % i do wypłaty (pole wyliczane). Napisz funkcje zapisu i odczytu z pliku. Wywołaj te funkcje kilkakrotnie. Odczytaj plik. Wyświetl dane pracowników, których kwota do wypłaty przekracza podaną wartość.

#### **Zadanie 14.5. Baza danych książek**

Załącz plik, będący prostą kartotekową bazą danych książek. Zadeklaruj strukturę opisującą pozycję bibliograficzną. Napisz funkcje zapisu i odczytu z pliku. Zapis pozycji (liczba zapisów nie jest określona) zakończ umownym znakiem (np. \* zamiast nazwiska autora). Napisz funkcję wyświetlającą tytuły książek podanego autora.

#### **Zadanie 14.6. Pomiary temperatur w pliku**

Załącz plik z pomiarami temperatur (liczby rzeczywiste). Napisz funkcje zapisu i odczytu z pliku. Wywołaj te funkcje. Odczytaj plik. Oblicz średnią arytmetyczną z pomiarów przechowywanych w pliku.

### **Zadania dodatkowe**

#### **Zadanie 14.7. Dostęp swobodny do elementu pliku**

Załącz plik z  $n$  wylosowanymi liczbami całkowitymi. Napisz funkcje zapisu i odczytu z pliku. Wywołaj te funkcje. Odczytaj plik. Wyświetl element pliku na podanej pozycji.

Wskazówka: wykorzystaj funkcję `fseek()`.



### Zadanie 14.8. Zawody sportowe

Załącz plik z wynikami zawodów sportowych. Nazwa pliku to nazwa konkurencji. Zawartość pliku to: imię, nazwisko i wynik zawodnika. Napisz funkcje zapisu i odczytu z pliku. Wywołaj te funkcje. Odczytaj plik. Wyświetl trzy najlepsze wyniki i dane zawodników, którzy je otrzymali.

### Zadanie 14.9. Eksport towarów

Załącz plik z danymi o eksportowanych za granicę towarach zawierający dane: nazwa towaru, kraj eksportu i wielkość eksportu w sztukach. Wyświetlić listę krajów, które eksportują podany towar i podać ogólną wielkość importu.

Napisz program, w którym wczytane są dane, wywołane odpowiednie funkcje, wyświetlone wyniki.

### Zadanie 14.10. Kopiowanie plików

Napisz program kopiujący co trzeci znak z jednego pliku do drugiego. Wyświetl zawartość obydwu plików.

### Zadanie 14.11. Pobieranie danych o plikach z argumentów wiersza poleceń

Napisz program, który do pliku o podanej nazwie wpisuje  $n$  kolejnych liczb całkowitych (od 1), a następnie wyświetla zawartość podanego pliku i średnią arytmetyczną liczb z pliku. Informację o nazwie pliku i informację ile liczb należy zapisać w pliku program pobiera z linii poleceń. W przypadku braku argumentów linii poleceń nazwa pliku to `dane.txt`, a  $n=10$ .

#### Wskazówka:

Funkcja `main` posiada dwa argumenty:

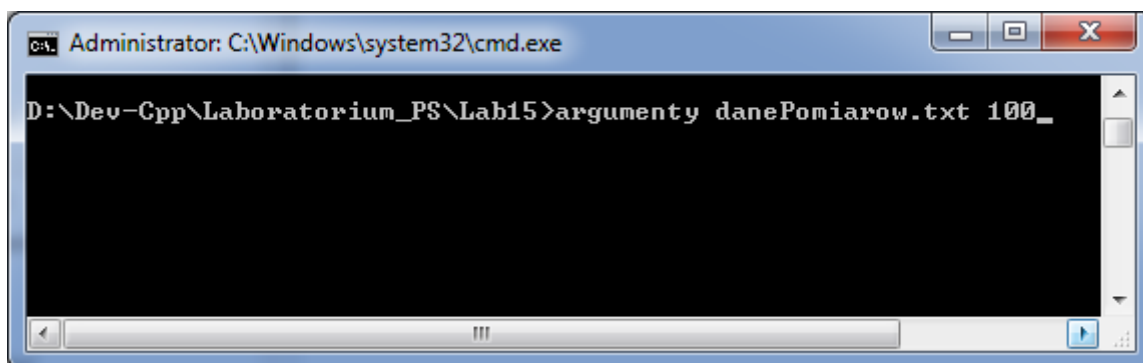
- argument `argc` (typu całkowitego) – przechowuje liczbę słów wpisanych w linii poleceń, uwzględniając nazwę programu,
- tablica łańcuchów `argv[]`, przechowuje słowa wpisane w linii poleceń:
  - `argv[0]` jest nazwą programu,
  - `argv[1]` jest pierwszym argumentem wywołanego programu (tylko wtedy, gdy w linii poleceń wpisujemy coś więcej niż tylko nazwę programu),
  - `argv[2]` – kolejnym argumentem, itd...
- wykorzystaj funkcję konwersji tekstu na liczbę.

Poniższy program wyświetla wszystkie argumenty podane w linii poleceń:

```
1  #include <stdio.h>
2
3  int main(int argc, char *argv[])
4  {
5      int i;
6      for(i=0;i<argc;i++)
7          printf("%s\n", argv[i]);
8      return 0;
9  }
```

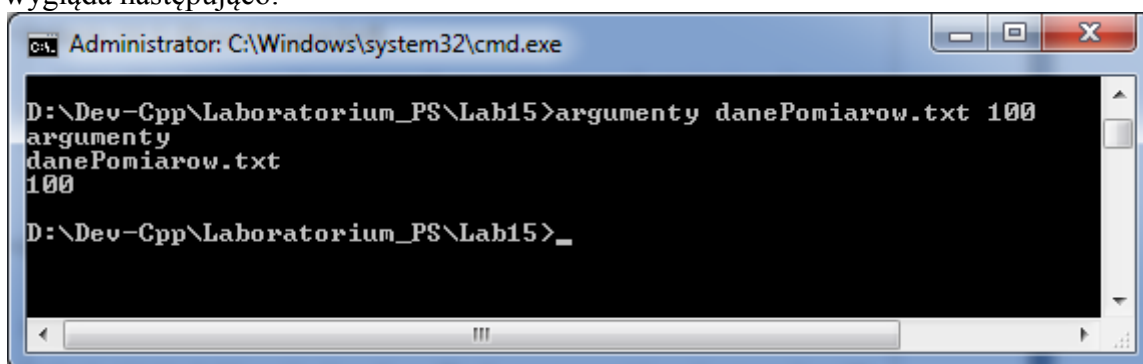


Przykładowe działanie programu, o nazwie `argumenty.exe`, uruchomionego z wiersza poleceń w oknie konsoli `cmd`:



```
Administrator: C:\Windows\system32\cmd.exe
D:\Dev-Cpp\Laboratorium_PS\Lab15>argumenty danePomiarow.txt 100_
```

wygląda następująco:



```
Administrator: C:\Windows\system32\cmd.exe
D:\Dev-Cpp\Laboratorium_PS\Lab15>argumenty danePomiarow.txt 100
argumenty
danePomiarow.txt
100
D:\Dev-Cpp\Laboratorium_PS\Lab15>_
```

Interpretacja wyniku:

`Argc =3`

`Argv[0]` – `argumenty.exe` – nazwa uruchamianego pliku

`Argv[1]` – `danePomiarow.txt`

`Argv[2]` – `100`