**Лабораторна робота № 4. Внутрішнє подання інтегрованих структур даних**

***Мета:*** отримати та закріпити знання про внутрішнє подання інтегрованих структур даних у мовах програмування.

**1 Вимоги**

**1.1 Розробник**

* Макаренко Владислав Олександрович
* Студент 1-го курсу
* Групи КІТ-120а

**1.2 Загальне завдання**

* Написати програму, яка виводить на екран внутрішнє подання структури з варіантною частиною та з бітовими полями, а також масива структур.

**2 Описи програм**

**Код програми**

#include <stdio.h>  
#include <stdbool.h>  
  
#include <time.h>  
  
  
void BYTE(unsigned char *A*) // виведення вмісту байта  
{  
 for(int i = sizeof(*A*) \* 8 - 1; i >= 0; --i){  
 printf("%d", (*A* >> i) & 1);  
 }  
 printf("%c", ' ');  
}  
  
#pragma pack (push,1)  
struct Train\_transport {  
 bool stops\_at\_all\_stations;  
 bool Traffic\_on\_the\_schedule;  
 union {  
 struct {  
 short number\_of\_people;  
 }passenger\_train;  
 struct {  
 short number\_of\_cars;  
 double amount\_of\_cargo;  
 }freight\_train;  
 } un1;  
};  
#pragma pack(pop)  
  
  
#pragma pack (push,1)  
struct Initial  
{  
 int a:4;  
 int b:4;  
 int c;  
};  
#pragma pack(pop)  
  
  
int main()  
{  
 //srand(time(NULL));  
 struct Train\_transport obj1 = {true, true};  
 obj1.un1.freight\_train.amount\_of\_cargo = 121.245;  
 obj1.un1.freight\_train.number\_of\_cars = 44;  
 printf("Size of structure train\_transport structure: %ld\n", sizeof(struct Train\_transport));  
 unsigned char \*p = (unsigned char\*) &obj1;  
 int byte = 0;  
 for( ; byte < sizeof(struct Train\_transport); byte++, p++ ) {  
 if(byte % 8 == 0)  
 printf("\n");  
 BYTE(\*p);  
  
 }  
 printf("\n");  
 clock\_t start = clock();  
 printf("\nNumber of cars: %d\n", obj1.un1.freight\_train.number\_of\_cars);  
 clock\_t end = clock();  
 double seconds = (double)(end - start) / CLOCKS\_PER\_SEC;  
 printf("\nAccess time is %.10lf second(s)\n", seconds);  
  
 struct Initial obj2 = { 4, 4, 44 };  
 unsigned char\* pt = (unsigned char\*)&obj2;  
 int byte1 = 0;  
 printf("\nSize of Initial structure: %ld\n", sizeof(struct Initial));  
 for (; byte1 < sizeof(struct Initial); byte1++, pt++)  
 {  
 if (byte1 % 8 == 0) printf("\n");  
 BYTE(\*pt);  
 }  
 printf("\n");  
 printf("\n");  
  
 struct Train\_transport array[2] = {{false, true}, {false, false} };  
 array[0].un1.freight\_train.amount\_of\_cargo = 25.34;  
 array[0].un1.freight\_train.number\_of\_cars = 44;  
 array[1].un1.passenger\_train.number\_of\_people = 111;  
  
 p = (unsigned char\*)&array;  
 byte = 0;  
 printf("Size of struct array: %ld\n", sizeof(array));  
 for (; byte < sizeof(array); byte++, p++)  
 {  
 if (byte % 8 == 0) printf("\n");  
 BYTE(\*p);  
 }  
 printf("\n");  
 return 0;  
}

**Результати виконання програми з вирівнюванням**

За алгоритмом коду дізнаємося значення внутрішнього подання структури з вирівнюванням (див. рис. 1).

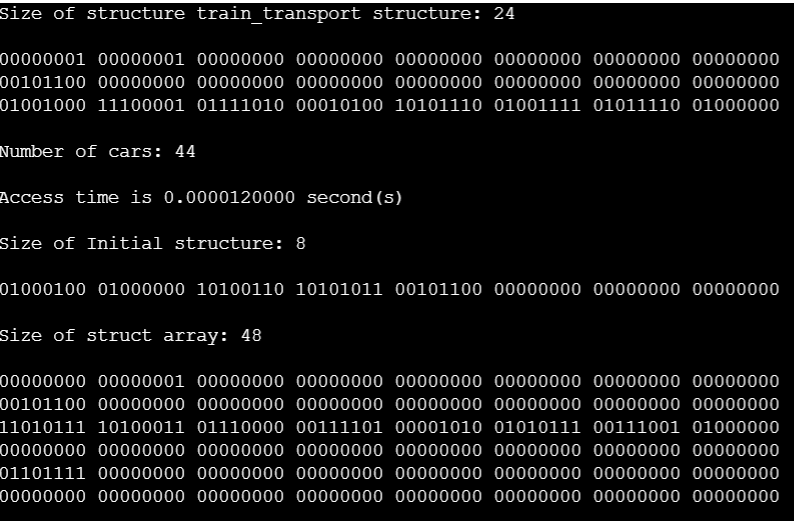


Рисунок 1 – Результати роботи першого кроку

**Результати виконання програми без вирівнювання**

За алгоритмом коду дізнаємося значення внутрішнього подання структури без вирівнювання (див. рис. 2).

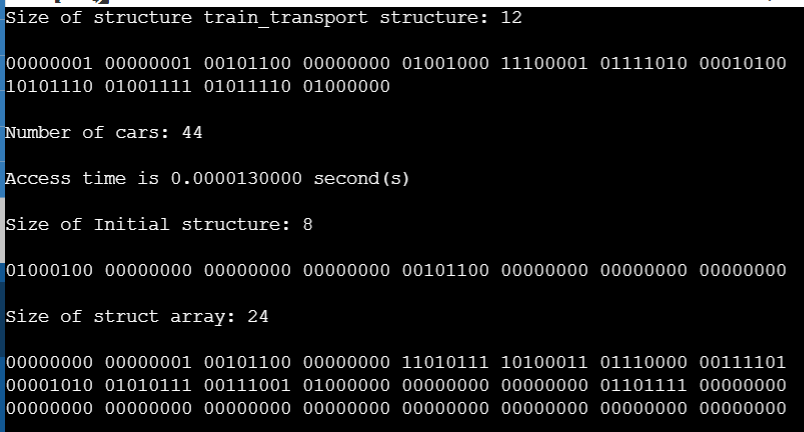


Рисунок 2 – Результати роботи другого кроку

**3 Порівняння часу програм**

Порівняємо час доступу до структур з вирівнюванням(зліва) та без нього(справа) за допомогою функції *clock()* (див. рис.3).

|  |  |  |
| --- | --- | --- |
|  | **<** |  |

Рисунок 3 – Порівняння часу

Як бачимо з (рис.3) з вирівнюванням час доступу до структури менший.

**Висновок:** на цій лабораторній роботі ми отримали та закріпили знання про внутрішнє подання інтегрованих структур даних у мовах програмування.