

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Программирование»
Тема: Структуры и обзор stdlib

Студент гр. 8383
Преподаватель

Сырцова Е.А.
Чайка К.В.

Санкт-Петербург
2018

Цель работы: Научиться использовать структуры и изучить основные функции стандартной библиотеки языка программирования C.

Основные теоретические положения:

time.h В заголовочном файле time.h можно найти объявления типов и функций для работы с датой и временем. В том числе:

- Функция, позволяющая получить текущее календарное время
- Функция, позволяющая получить время в тактах процессора с начала выполнения программы
- Функция для вычисления разности в секундах между двумя временными штампами
- Функции для вывода значения даты и времени на экран А также структура tm, содержащая компоненты календарного времени и функция для преобразования значения времени в секундах в объект такого типа.

assert.h В стандартной библиотеке объявлен макрос препроцессора assert().

- С его помощью можно выполнять проверку некоторых условий в процессе выполнения программы. Если условие ложно, то в процессе выполнения будет выведена некоторая информация о том в каком месте это произошло (имя файла с исходным кодом, имя функции, номер строки) и само условие.

- Стоит заметить, что для отключения проверок достаточно всего лишь добавить макрос #define NDEBUG перед включением заголовочного файла assert.h

Использование assert может сильно упростить отладку программ, гарантируя, что все необходимые условия в процессе выполнения программы соблюдены.

string.h Стандартная библиотека Си содержит различные функции для работы с памятью и строками, объявленные в заголовочном файле string.h

Среди них есть функции для:

- копирования памяти
- сравнения памяти
- сравнения строк
- разбиения строки на токены
- конкатенации строк
- поиска символов и подстроки в строке
- определения длины строки и некоторые другие

qsort и bsearch В `stdlib.h` есть функции для сортировки и поиска в массиве любого типа. Пример функции `qsort`:

```
void qsort (void* base, size_t num, size_t size, int (*compar)(const void*,const void*));
```

Функция принимает указатель на начальный элемент массива, количество элементов и размер одного элемента, а также указатель на функцию для сравнения двух элементов. Так как тип элементов может быть любым, то и указатель на первый элемент массива имеет тип `void`. Это позволяет, зная адрес первого элемента и размер каждого элемента вычислить адрес любого элемента массива в памяти и обратиться к нему. Остается только сравнить 2 элемента имея 2 указателя на них. Это выполняет функция `compar`, указатель на которую передается функции `qsort` в качестве одного из параметров. Функция `compar` принимает 2 указателя типа `void`, но в своей реализации может привести их к конкретному типу (так как её реализация остается за программистом, он точно знает элементы какого типа он сортирует) и сравнивает их. Результат сравнения определяется знаком возвращаемого функцией `qsort` числа.

Вывод:

В процессе выполнения работы были изучены структуры и некоторые функции стандартной библиотеки языка C.

Приложение:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 1000
int sort (const void * a, const void * b){
    return ( abs(*(int*)b) - abs(*(int*)a) );
}
int main (){
    int arr[N];
    int* res1;
    clock_t start1;
    clock_t end1;

    for(int i = 0; i < N; i++){
        scanf("%d", &arr[i]);
    }

    start1 = clock();
    qsort (&arr, N, sizeof (int), sort);
    end1 = clock();

    for (int i = 0; i < N; i++) {
        printf("%d ", arr[i]);
    }

    printf("\n%f", 1.0 * (end1-start1)/ CLOCKS_PER_SEC);

    return 0;
}
```