

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: Обзор стандартной библиотеки

Студент гр. 8383

Федоров И.А

Преподаватель

Круглик А.Д

Санкт-Петербург

2018

Цель работы.

Ознакомиться со стандартной библиотекой Си и закрепить полученные знания на практике, выполняя задачу лабораторной работы.

Основные теоретические положения.

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.

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. Для отключения проверок достаточно всего лишь добавить макрос `#define NDEBUG` перед включением заголовочного файла `assert.h`

В языке Си возможно использование функций с произвольным числом и типом аргументов (вспомним функцию `printf`, например). Для удобства работы с такими аргументами есть средства, находящиеся в заголовочном файле `stdarg.h`

Заголовочный файл определяет специальный тип `va_list` и набор функций для работы с этим типом `va_start`, `va_arg`, `va_end`, с помощью которых можно по очереди перебирать аргументы функции.

В состав стандартной библиотеки входят функции для работы с символами, объявленные в заголовочном файле `ctype.h`

Эти функции позволяют переводить символы из верхнего регистра в нижний, определять является ли этот символ буквой, цифрой или каким-то служебным.

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

В заголовочном файле `stdlib.h` собраны объявления различных функций

- Функции для работы с динамической памятью
- Функции для преобразования строки в число
- Генерации псевдослучайных чисел
- Функции для управления процессом выполнения программы
- Функции для вычисления абсолютного значения и деления целых чисел
- Функции для сортировки и поиска

Указатели на функции

Указатель на функцию позволяет нам работать с функцией как с обычной переменной, в том числе передавать функцию в качестве аргумента другой функции. [Подробнее](#) 5.11

Для написания функции нахождения минимума в массиве элементов неизвестного типа указатель на функцию нужен, чтобы сравнивать элементы. Похожая логика используется во многих языках программирования, а функцию сравнения двух элементов обычно называют компаратор (*англ compare - сравнивать*). Компаратор работает по следующему принципу: если элементы равны, результатом сравнения будет 0, если первый больше - результат 1 или любое положительное число, иначе -1 или любое отрицательное.

Перепишем нашу универсальную функцию нахождения минимума с учетом указателя на функцию-компаратор:

```
void* min(void*arr, int array_length, int size_of_element, int (*compar)(const void*,const void*)){\n    void* min = arr;\n    for(int i = 0; i < array_length; i++){ \n        if(compar(arr + i*size_of_element, min) < 0)\n            min = arr + i*size_of_element;\n    }
```

```
}  
    return min;  
  
}
```

Теперь мы можем вызывать функцию `min` для любого типа, главное определить функцию сравнения двух элементов.

Функции работы с датой и временем

Для работы с системной датой и временем используется библиотека `time.h`.

Типы, определенные в библиотеке `time.h`

Тип	Описание
<code>size_t</code>	Целочисленный тип, возвращаемый операцией <code>sizeof</code> .
<code>clock_t</code>	Арифметический тип, подходящий для представления времени.
<code>time_t</code>	Арифметический тип, подходящий для представления времени.
<code>struct tm</code>	Структурный тип, содержащий компоненты календарного времени.

Для определения текущего календарного времени используется функция

```
time_t time(NULL);
```

Данная функция возвращает время в секундах начиная с 1 января 1970 г.

qsort и bsearch

Как мы уже говорили, в `stdlib.h` есть функции для сортировки и поиска в массиве любого типа. Давайте рассмотрим как такое возможно на примере функции `qsort`:

```
void qsort (void* base, size_t num, size_t size,  
            int (*comp)(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 `comp`, указатель на которую передается функции `qsort` в качестве одного из параметров.

Функция `strcmp` 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.

Выводы.

Были изучены такие заголовочные файлы стандартной библиотеки, как `time.h`, `assert.h`, `stdarg.h`, `ctype.h`, `string.h`, `locale.h`, `wchar.h`, `stdlib.h`, а также функции бинарного поиска и быстрой сортировки.

Приложение

Код программы:

```
#include <time.h>
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_LEN 1000
#define CLOCKS_PER_SEC 1000

int comp(const void * a, const void * b){
    return ( abs(*(int*)b) - abs(*(int*)a));
}

int main()
{
    int array[ARRAY_LEN] = {0};
    int i = 0;
    time_t time;
    while(i < ARRAY_LEN)
    {
        scanf("%d", &array[i]);
        i++;
    }
    time = clock();
    qsort(array, ARRAY_LEN, sizeof(int), comp);
    time = clock() - time;
    for(int i = 0; i < ARRAY_LEN; i++)
    {
        printf("%d ", array[i]);
    }
    printf("\nExecution time of ""Quick Sort"": %f\n",
        ((float)time)/CLOCKS_PER_SEC);
    return 0;
}
```