

🏠 (<http://cppstudio.com>)

/ Стандартные заголовочные файлы из Си в C++ (<http://cppstudio.com/cat/309/>)

/ Заголовочный файл cstdlib (stdlib.h) (<http://cppstudio.com/cat/309/324/>) / Функция bsearch

Функция bsearch

★★★★★ Оценка: **4,00** (голосов: 1)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции bsearch:

```
1 void * bsearch( const void * searchkey, const void * baseptr, size_t number, size_t
```

Название	Язык
stdlib.h	C
cstdlib	C++

Описание

Функция `bsearch` выполняет двоичный поиск в массиве.

Алгоритм поиска запрашивает искомое значение, которое передается через параметр `searchkey`, в массиве, на который ссылается указатель `baseptr`. Количество элементов массива передается через параметр `number`, каждый элемент массива имеет размер `size` байт. В конце, функция возвращает указатель `void *` на первое вхождение искомого значения.

Для выполнения поиска, функция сравнивает элементы с искомым элементом, путем вызова функции `comparator` через параметр `*comparator`, указанного в качестве последнего аргумента.

Потому как эта функция выполняет алгоритм бинарного поиска, все значения в массиве `baseptr` должны быть отсортированы в порядке возрастания, с теми же критериями, которые использует функция `funccompar`.

Параметры:

- **searchkey**
Указатель на искомый объект типа `void*`.
- **baseptr**
Указатель на первый элемент массива, в котором будет выполняться поиск, типа `void*`.
- **number**
Количество элементов массива `base`.

- **size**

Размер каждого элемента массива `base` в байтах.

- **funccompa**

Функция для сравнения двух элементов массива. Функция должна иметь следующий прототип:

```
1 int funccompar( const void * searchkey, const void * cmpelem );
```

Функция должна принимать два параметра: первый указывает на искомый объект, а второй – на один из элементов массива, типа `void *`. Функция должна привести передаваемые параметры к нужным типам данных и выполнить сравнение.

Возвращаемое значение этой функции неоднозначно. Если сравниваемый элемент с искомым значением меньше, равен или больше, функция должна вернуть, отрицательное значение, ноль или положительное значение, соответственно.

Для основных типов данных, которые поддерживают регулярные выражения, функция сравнения может выглядеть так:

```
1 int funccompar (const void * key, const void * cmpelem)
2 {
3     if ( *(type*)key >  *(type*)cmpelem ) return 1;
4     if ( *(type*)key == *(type*)cmpelem ) return 0;
5     if ( *(type*)key <  *(type*)cmpelem ) return -1;
6 }
```

Возвращаемое значение

Указатель на элемент массива, который равен значению в `key`.

Если искомый элемент не найден, возвращается нулевой указатель.

Пример: исходный код программы

```
1 //пример использования функции bsearch
2 #include <iostream>
3 #include <cstdlib>
4
5 int funcncmp(const void * x1, const void * x2) // фун
6 {
7     return ( *(int*)x1 - *(int*)x2 ); //
8 }
9
10 int values[] = { 10, 20, 25, 40, 90, 100 }; //
11
12 int main()
13 {
14
15     int key = 40; //
16     int * ptrItem = (int*) bsearch(&key, values, 6, sizeof (int), funcncmp);
17
18     if (ptrItem != NULL)
19         std::cout << "Число " << *ptrItem << " находится в массиве";
20     else
21         std::cout << "Числа " << key << " в массиве нет";
22     return 0;
23 }
```

Пример работы программы

В примере есть массив отсортированных целых чисел. Существует также функция `funcstr`, которая сравнивает два значения передаваемых через параметры `x1` и `x2`, и возвращает результат вычитания значений. Если результат `0` – значения равны, положительный результат – сравниваемое значение меньше, отрицательный результат – сравниваемое значение больше.

В `main` вызывается функция `bsearch` для поиска в массиве `values` значение `40`. Как только искомое значение найдено, программа выводит его на экран, смотреть пример работы программы ниже.

CppStudio.com

Число 40 находится в массиве

Для Си-строк функция `strcmp` может быть использована в качестве последнего аргумента функции `bsearch`.

Пример: исходный код программы

```
1 //пример использования функции bsearch, для поиска строки в массиве строк
2 #include <iostream>
3 #include <cstdlib>
4 #include <cstring>
5
6 char arraystr[][20] = {"галактика", "млечный", "путь", "вселенная", "космос"};
7
8 int main()
9 {
10     char key[20] = "путь";
11
12     // сортируем строки в массиве
13     qsort(arraystr, 5, 20, (int (*)(const void*, const void*)) strcmp);
14
15     std::cout << "Отсортированный массив строк:\n";
16     for (int ix = 0; ix < 5; ix++)
17         std::cout << arraystr[ix] << " ";
18
19     // поиск строки
20     char * ptrItem = (char*) bsearch(key, arraystr, 5, 20, (int (*)(const void*, const void*)) strcmp);
21
22     if (ptrItem != NULL)
23         std::cout << "Строка " << ptrItem << " есть в массиве";
24     else
25         std::cout << "Строки " << key << " в массиве нет";
26     return 0;
27 }
```

Пример работы программы


CppStudio.com


Отсортированный массив строк:


вселенная галактика космос млечный путь

Строка «путь» есть в массиве

 Обсудить на форуме (/topics/)

 Автор: admin (/forums/users/admin/)

 Дата: 04.09.2012

 Поделиться:

Похожие статьи:

1. Класс Массив, описывающий одномерный массив (<http://cppstudio.com/post/4975/>)
2. Обработка массива данных предметной области «штатное расписание» (<http://cppstudio.com/post/1500/>)
3. Массивы в C++ (<http://cppstudio.com/post/389/>)
4. Шаблоны функций в C++ (<http://cppstudio.com/post/5165/>)
5. Класс Матрица, описывающий двумерный массив (<http://cppstudio.com/post/4946/>)

Оставить комментарий

Вы должны войти ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F887%2F)

[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F887%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F887%2F)), чтобы оставить комментарий.

Translation

 (/post/887/)Русский (/post/887/)
 (/uk/post/887/)Українська (/uk/post/887/)
 (/en/post/887/)English (/en/post/887/)
 (/de/post/887/)Deutsch (/de/post/887/)
 (/be/post/887/)Беларуская (/be/post/887/)
 (/kk/post/887/)Қазақ тілі (/kk/post/887/)
 (/uz/post/887/)O'zbek tili (/uz/post/887/)
 (/tr/post/887/)Türkçe (/tr/post/887/)

Новое

Популярное

Sorry. No data so far.

➤ Особенности Qt: слоты и сигналы, описание QObject и QApplication, виды окон и т.д.
(<http://cppstudio.com/post/11167/>)

➤ Первая программа на Qt:
(<http://cppstudio.com/post/11127/>)

➤ Введение – графическая библиотека Qt
(<http://cppstudio.com/post/11097/>)

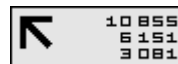
➤ Наследование классов
(<http://cppstudio.com/post/10103/>)

➤ Перегрузка операторов в C++
(часть 2)
(<http://cppstudio.com/post/10058/>)

© 2022 CppStudio – Программирование для начинающих на C++



(<https://plus.google.com/u/0/106109650739084338784>)



(<http://www.liveinternet.ru/click>)



(<http://orphus.ru>)