

🏠 (<http://cppstudio.com>)

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

/ Заголовочный файл csignal (signal.h) (<http://cppstudio.com/cat/309/321/>) / Функция signal

Функция signal

★★★★★ Оценка: **4,00** (голосов: 1)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции signal:

```
1 void (*signal)(int sign, void (*funcptr)(int))(int);
```

Заголовочный файл

Название	Язык
signal.h	C
csignal	C++

Описание

Функция `signal` определяет способ обработки сигналов, номер сигнала передается через параметр `sign`.

Есть три способа, обработки сигнала программой (способ обработки указывается параметром `funcptr`):

- Обработка сигналов по умолчанию `SIG_DFL`: сигнал обрабатывается, по умолчанию, событием для данного сигнала.
- Игнорировать сигнал `SIG_IGN`: сигнал игнорируется и выполнение кода будет продолжаться, даже если это бессмысленно.
- Функция обработчик: специфическая функция, определена для обработки сигналов.

Конкретная реализация компилятора может использовать или `SIG_DFL`, или `SIG_IGN`, как способ обработки сигнала по умолчанию при запуске программы.

Параметры:

- **sign**

Номер сигнала, который функция будет обрабатывать. Далее в таблице указаны константы, определяющие стандартные номера сигналов:

Макрос	Сигнал
SIGABRT	(Сигнал остановки) Аварийное завершение, такой сигнал вызывает функцию <code>abort</code> .
SIGFPE	(Сигнал исключения с плавающей точкой) Возникновение ошибочных арифметических операций, таких, как деление на ноль или переполнение типов данных (данный сигнал актуален и для целочисленных операций).
SIGILL	(Сигнал недопустимой инструкции) Неправильная запись функции, то есть неправильно сформированная инструкция. Как правило, это связано с ошибками в коде или ошибочной попыткой обработать информацию.
SIGINT	(Сигнал прерывания) Интерактивный сигнал внимания. Как правило, генерируемый приложением пользователя.
SIGSEGV	(Сигнал о нарушении сегментации) Неверный доступ к памяти: когда программа пытается считать или записать данные в некорректно выделенную память.
SIGTERM	(Сигнал Завершения) Прекращение запроса, отправленного в программу.

В различных реализациях компиляторов могут быть предусмотрены дополнительные номера сигналов и их макро-константы, для использования этой функцией.

Обратите внимание, что не все среды программирования поддерживают возможность создания автоматических сигналов, даже тех, которые описаны выше.

• funcptr

Указатель на функцию. Это может быть либо функция, определенная программистом или один из следующих предопределенных способов обработки сигналов:

SIG_DFL	Сигнал обрабатывается по умолчанию.
SIG_IGN	Сигнал игнорируется.

Если функция, то её прототип должен быть таким:

```
1 void function( int param );
```

Возвращаемое значение

Тип данных возвращаемого значения точно такой же, как для параметра `func`.

Если запрос выполнен успешно, `signal` возвращает указатель на конкретную функцию обработчик, которая отвечает за обработку этого сигнала, если таковой имеется. Или `SIG_DFL` или `SIG_IGN`, если перед вызовом, сигнал обрабатывается по умолчанию или игнорируется.

Если функция не обработала успешно сигнал, она возвращает `SIG_ERR` и устанавливает `ERRNO` в положительное значение.

Пример: исходный код программы

```

1  // пример использования функции signal
2
3  #include <iostream>                // для оператора cout
4  #include <cstdio>                  // для функций tmpnam, remove
5  #include <cstdlib>                 // для функции exit
6  #include <csignal>
7
8  char tmpfilename [L_tmpnam];      // строка для хранения имени файла
9
10 void terminate (int param)
11 {
12     std::cout << "Завершение программы..." << std::endl;
13
14     remove (tmpfilename);          // удалить временный файл
15
16     exit(1);                       // завершить работу программы
17 }
18
19 int main()
20 {
21     void (*funcptr)(int);          // указатель на функцию
22
23     funcptr = signal (SIGTERM, terminate); // обработка сигнала
24     if (funcptr == SIG_IGN) signal(SIGTERM, SIG_IGN); // в случае, если funcptr был SIG_IGN
25
26     tmpnam(tmpfilename);           // сгенерировать имя файла
27
28     // остальной код программы
29     raise(SIGTERM); // отправить программе сигнал SIGTERM
30     return 0;
31 }

```

Пример работы программы

Эта программа определяет функцию `terminate`, которая удаляет ранее созданные временные файлы. Эта функция использует сигнал `SIGTERM`, если этот сигнал игнорируется, он сбрасывается в `SIG_IGN`, так что поведение остается неизменным.

Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 08.09.2012

Поделиться:

Похожие статьи:

1. Особенности Qt: слоты и сигналы, описание `QObject` и `QApplication`, виды окон и т.д. (<http://cppstudio.com/post/11167/>)
2. Функция `raise` (<http://cppstudio.com/post/1200/>)
3. Функция `atexit` (<http://cppstudio.com/post/869/>)
4. Первая программа на Qt: (<http://cppstudio.com/post/11127/>)
5. Синтаксис: определить в C++ адрес в памяти (<http://cppstudio.com/post/5325/>)









Оставить комментарий

Вы должны войти (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1204%2F), чтобы оставить комментарий.

Поиск...

Поиск

Translation

-  (/post/1204/)Русский (/post/1204/)
-  (/uk/post/1204/)Українська (/uk/post/1204/)
-  (/en/post/1204/)English (/en/post/1204/)
-  (/de/post/1204/)Deutsch (/de/post/1204/)
-  (/be/post/1204/)Беларуская (/be/post/1204/)
-  (/kk/post/1204/)Қазақ тілі (/kk/post/1204/)
-  (/uz/post/1204/)O‘zbek tili (/uz/post/1204/)
-  (/tr/post/1204/)Türkçe (/tr/post/1204/)

Новое

► Особенности Qt: слоты и сигналы, описание QObject и

Популярное

Sorry. No data so far.

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/>)

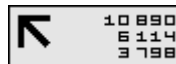
➤ Наследование классов
(<http://cppstudio.com/post/10103/>)

➤ 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>)