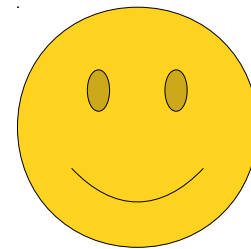
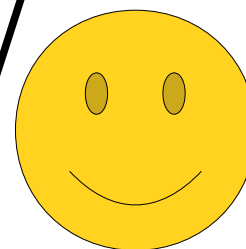


作业 0：使用调试器

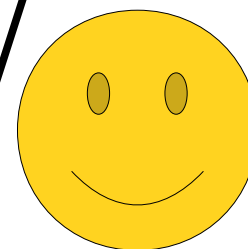
嗨！大家好！



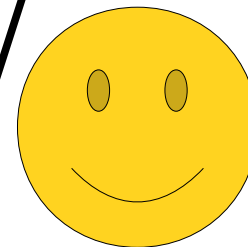
作为作业 0 的一部分，希望大家练习下 Qt Creator
的调试器



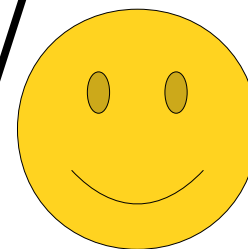
调试器是在编程过程中常用的工具，方便你查看在程序运行时，究竟做了哪些工作。



在查找程序错误时，调试器非常有用。越是熟悉调试工具，就越能更容易地发现程序中的错误。

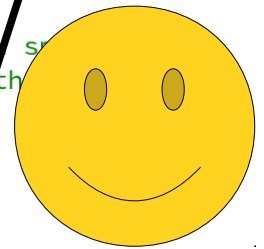


把此份手册当作一个简单的教程，带你熟悉调试器的工作方式，以及在调试过程中你将会看到的东西



首先，打开第二步中的 Name Hash
程序，向下找到 nameHash 函数，如图

```
42  * For t
43  * treat
44  * It th
45  * F_p, wh
46  * some smaller prime number q (you didn't expect a
47  * but we thought it might be fun!)
48  */
49  int nameHash(string first, string last){
50  /* This hashing scheme needs two prime numbers, a large prime and a small
51   * prime. These numbers were chosen because their product is less than 2^31.
52   * 2^31 - kLargePrime - 1.
53   */
54   static const int kLargePrime = 16908799;
55   static const int kSmallPrime = 127;
56
57   int hashVal = 0;
58
59   /* Iterate across all the characters in the first name, then the last
60    * name, updating the hash at each step.
61   */
62   for (char ch: first + last) {
63       /* Convert the input character to lower case. The numeric values of
64        * lower-case letters are always less than 127.
65       */
66       ch = tolower(ch);
67       hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68   }
69   return hashVal;
70 }
71
```



Activities Qt Creator Jan 4 3:09 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

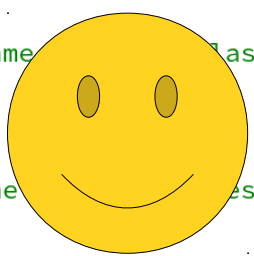
Projects NameHash [main] NameHash.pro Sources NameHash.cpp

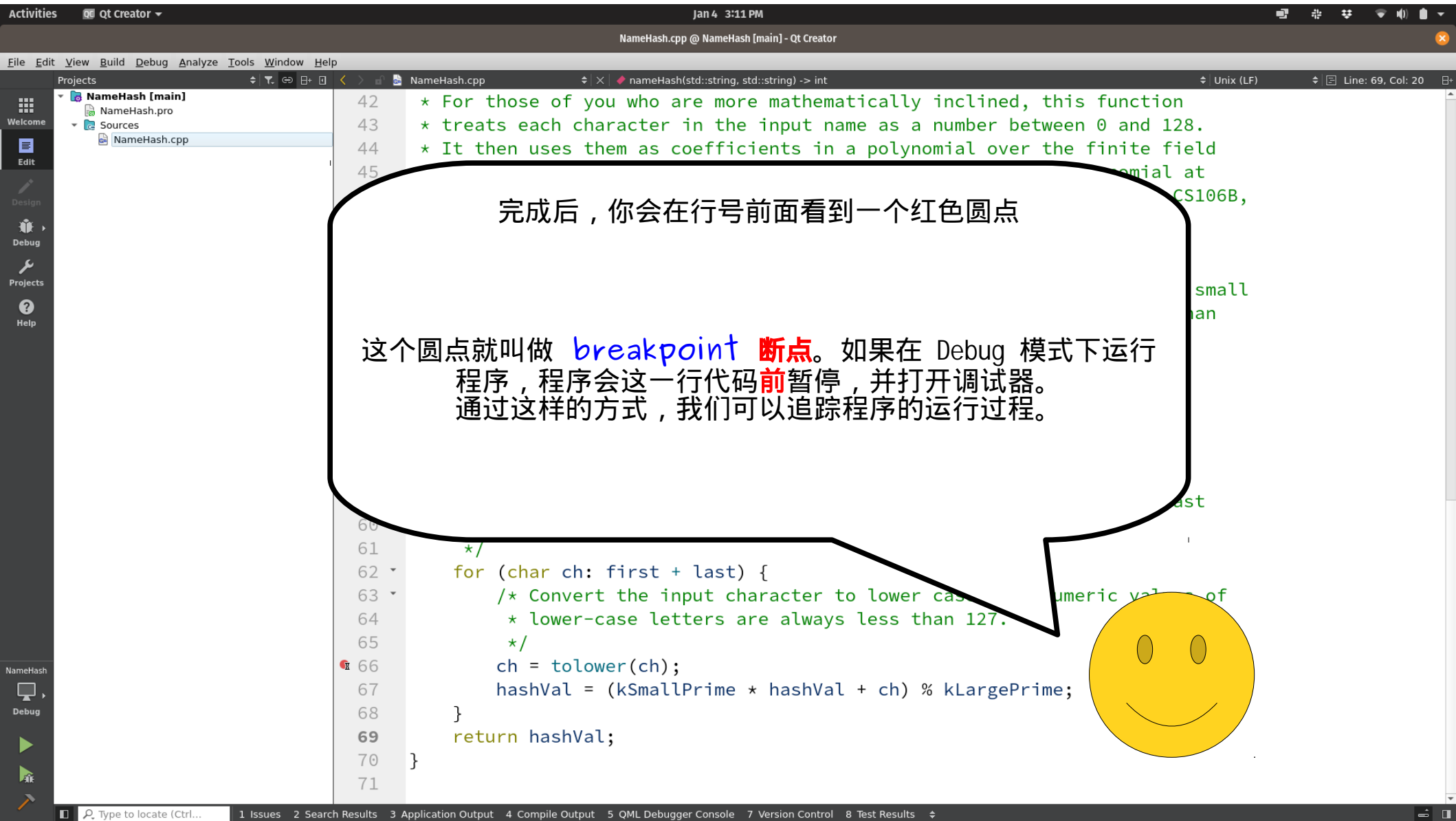
42 * For those of you who are more mathematically inclined, this function
43 * treats each character in the input name as a number between 0 and 128.
44 * It then uses them as coefficients in a polynomial over the finite field
45 * F_p , where p is a large prime number, and evaluates that polynomial at $x = 1$. This is for CS106B,

把鼠标移动到 66 行前面。
此时，点击一下鼠标左键！

56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the input name, from first to last
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case. The values of
64 * lower-case letters are always less than 127.
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71

and a small
less than







.....你将会看到这样的界面。此时，Qt 界面也会弹出一些额外的面板，我们后面会一个个看。

The screenshot shows the Qt IDE interface for a project named 'NameHash'. The main editor displays the following C++ code:

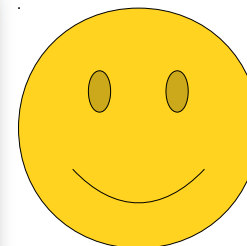
```
45  
46 * some smaller prime number q. (You aren  
47 * but we thought it might be fun!)  
48 */  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66 ch = tolower(ch);  
67 hashVal = (kSmallPrime * hashVal + ch) % kl  
68 }  
69 return hashVal;
```

A 'NameHash Console' window is open, displaying the text 'What is your first name?'. A yellow smiley face is drawn next to the console. The bottom status bar shows the debugger is running, with the message 'Application started.' and a table of threads.

Level	Function	File	Line	Address	Number	Funct File	Line	Address	Condition	Ignore	Threads
1		...g) ...eHash.cpp	66	...5555b6782							(all)

此时，切换到英文输入法，输入 **Ada** 后按回车，**注意不要输入你自己的名字！**

(当然，除非你就叫 Ada)



Activities NameHash

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
45
46 * some smaller prime number q. (You aren't
47 * but we thought it might be fun!)
48 */
49
50
51
52 What is your first name? Ada
53 What is your last name?
54
55
56
57
58
59
60
61
62
63
64
65
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kl
68 }
69 return hashVal;
```

NameHash Console

File Edit Options Help

What is your first name? Ada
What is your last name?

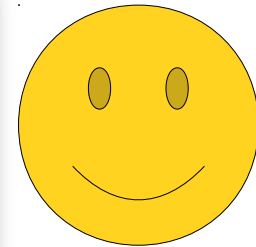
Debugger GDB for "NameHash" Threads: #12 Application started.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1		...g	...eHash.cpp	66	5555b6782				(all)

Type to locate (Ctrl...)

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

接下来，再输入 **Love**lace 但是不要再按回车了



Activities NameHash

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

45
46 * some smaller prime number q. (You aren
47 * but we thought it might be fun!)
48 */
49
50
51
52 What is your first name? Ada
53 What is your last name? Love**l**ace
54
55
56
57
58
59
60
61
62
63
64
65
66 `ch = tolower(ch);`
67 `hashVal = (kSmallPrime * hashVal + ch) % kL`
68 `}`
69 `return hashVal;`

NameHash Console

File Edit Options Help

What is your first name? Ada
What is your last name? Love**l**ace

Debugger GDB for "NameHash" Threads: #12 Application started.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
					1	...	g) ...eHash.cpp	66	...5555b6782			(all)

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

因为一旦你按下回车，Qt 就会弹出一系列对话框。
不要担心，这是正常的。



```
46 * some smaller prime number q. (You aren't
47 * but we thought it might be fun!)
48 */
49
50 NameHash Console
51 File Edit Options Help
52 [Icons] [A] [Icons] [Icon]
53 What is your first name? Ada
54 What is your last name? Lovelace
55
56
57
58
59
60
61
62
63
64
65
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kSmallPrime;
68 }
69 return hashVal;
```

如前所述，你应该做好心理准备了。
准备好.....按回车，一起来看看发生
了什么魔法！

Activities NameHash

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

45
46 * some smaller prime number q. (You aren
47 * but we thought it might be fun!)
48 */
49
50
51
52
53 What is your first name? Ada
54 What is your last name? Lovelace
55
56
57
58
59
60
61
62
63
64
65
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kl
68 }
69 return hashVal;

NameHash Console

File Edit Options Help

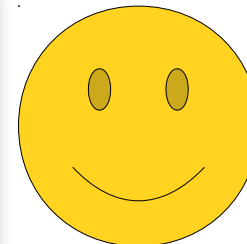
What is your first name? Ada
What is your last name? Lovelace

Debugger GDB for "NameHash" Threads: #12 Application started.

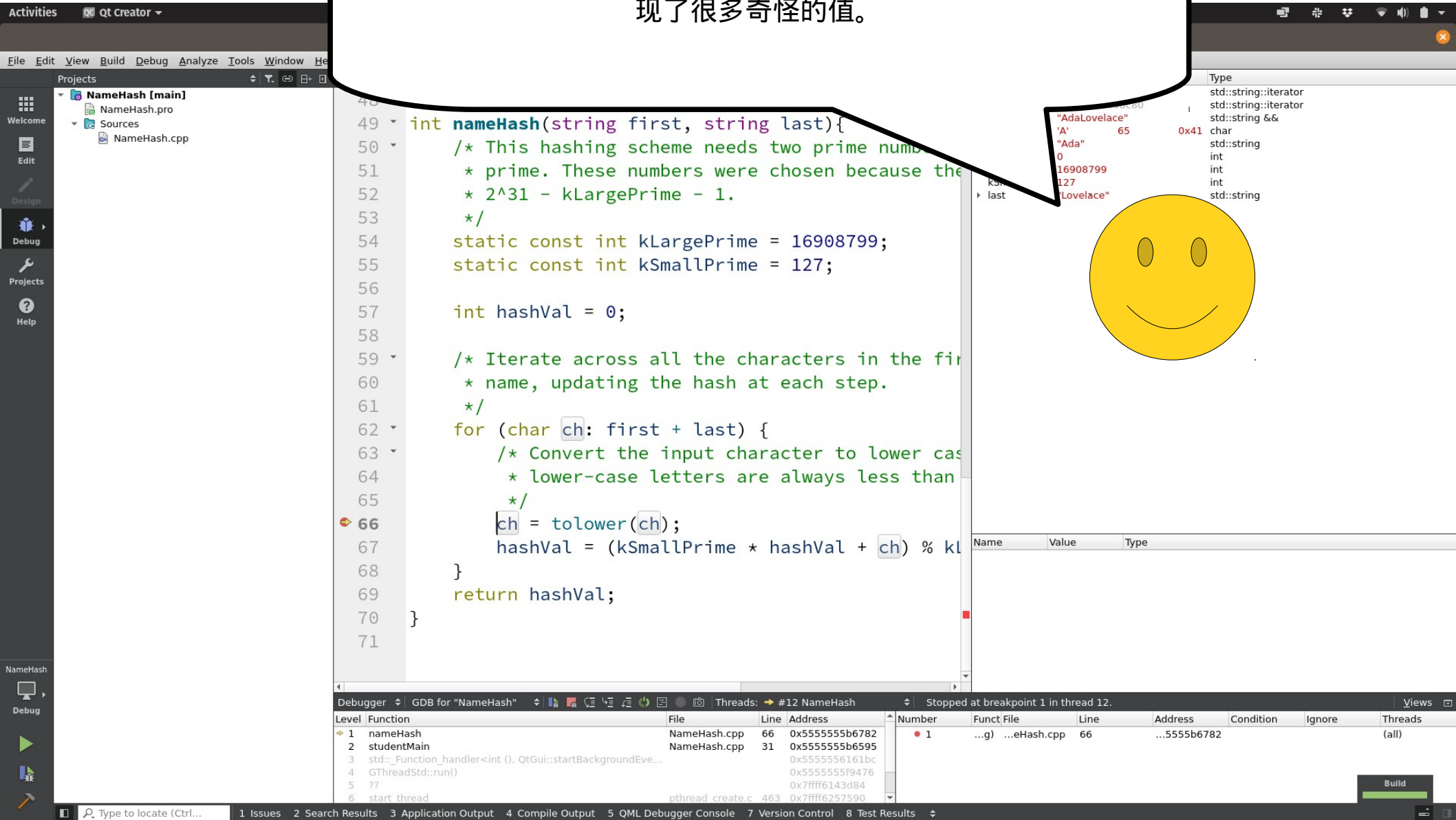
Level	Function	File	Line	Address	Number	Funct File	Line	Address	Condition	Ignore	Threads
1		...g) ...eHash.cpp	66	...5555b6782							(all)

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



哇哦！我们回到了 Qt Creator 并且在弹出的面板中，出现了很多奇怪的值。



The image shows the Qt Creator IDE with a C++ project named "NameHash". The main editor displays the source code for "NameHash.cpp". The code defines a function `nameHash` that takes two strings, `first` and `last`, and returns a hash value. The function uses two prime numbers, `kLargePrime` (16908799) and `kSmallPrime` (127), to calculate the hash. The code is as follows:

```
48
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers
51     * prime. These numbers were chosen because they are
52     * 2^31 - kLargePrime - 1.
53     */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first and last
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64         * lower-case letters are always less than 128.
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

The debugger window at the bottom shows the execution state. It is stopped at breakpoint 1 in thread 12. The stack trace shows the following frames:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop...>::operator()()			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

The right sidebar shows the variable explorer with the following variables:

Name	Value	Type
first	"AdaLovelace"	std::string
last	"A"	std::string
ch	"A"	char
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int

A yellow smiley face is drawn next to the variable explorer.

有很多东西值得说道说道，让我们一起来看看吧。

Qt Creator interface showing the NameHash project and the implementation of the nameHash function.

Project Structure:

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Code Snippet (NameHash.cpp):

```
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers
51      * prime. These numbers were chosen because they
52      * are 2^31 - kLargePrime - 1.
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Variable Watcher:

Name	Value	Type
first	"AdaLovelace"	std::string
last	"A"	std::string
ch	'A'	char
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int

Debugger:

Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Build Status: Build

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - \text{kLargePrime} - 1$ .
```

Variable Inspector:

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

首先，注意之前的红色圆点有了一个黄色箭头！

```
62 for (char ch: first + last)
63     /* Convert the input character to lower case
64      * lower-case letters are always
65      */
66     ch = tolower(ch);
67     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71
```

Debugger: GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555b6782	1	...	g) ...eHash.cpp	66	...	555555b6782		(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results Build

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - \text{kLargePrime} - 1$ .
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

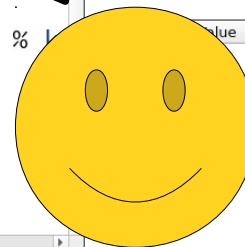
Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x5555555b6782
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
4	GThreadStd::run()			0x55555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Build

黄色箭头表明程序运行到了这一行之前。

这是因为我们之前设置过断点。



Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - \text{kLargePrime} - 1$ .
```

62 for (char ch: first + last)
63 /* Convert the input character to lower case
64 * lower-case letters are always
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68 }
69 return hashVal;
70 }
71 }

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

68 }

69 return hashVal;

70 }

71 }

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

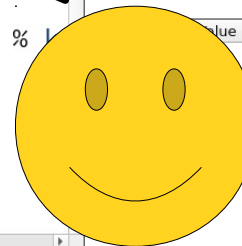
Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"A" 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

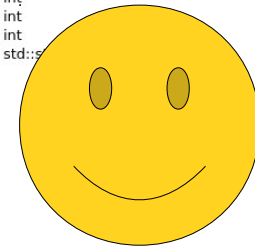
1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Build

每次弹出调试界面，最好确认下是否停在了你预设的断点上。
检查黄色箭头有没有出现是一个很好的习惯。



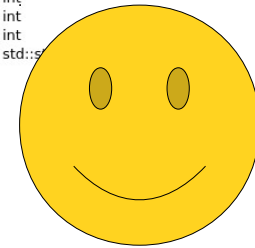
Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



接下来，我们看看下面的面板，我们称之为
call stack 调用栈。

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

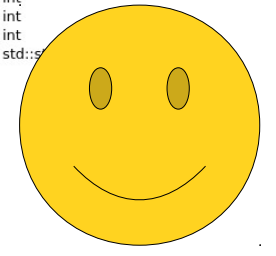
Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string



此刻，根据黄色箭头所指，我们处在 nameHash 函数中。

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

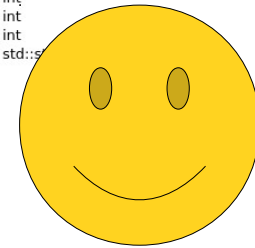
Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



然而，黄色箭头无法告诉我们，程序是如何执行到这个位置的。哪一部分的代码调用了 nameHash 这个函数呢？

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



调用栈可以告诉我们！

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::s



注意，调用栈按顺序列出了一系列不同的函数。
此时，nameHash 函数在顶部（程序暂停的位置），
紧接着下面是 studentMain 函数。

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:15 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

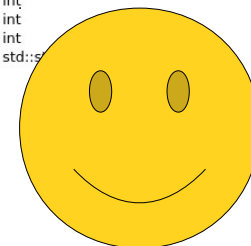
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::thread::id::operator int() const	Qt5Core/src/corelib/thread/qthread.cpp	104	0x55555556161								
4	GThreadSt...			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start threa...	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+K) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results Build

接下来，双击调用栈中的 studentMain 函数名，当你做完后.....



Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h" // for getLine

22 using namespace std;

23 /* Prototype for the nameHash function. This lets u

24 * in main and then define it later in the program.

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name

29 string last = getLine("What is your last name?

30

31 int hashCode = nameHash(first, last);

32

33 cout << "The hash of your name is: " << hashCode

34 return 0;

35 }

36

37 /* This is the actual function that c

38 * to talk more about what hash functi

39 * the meantime, think of it as a func

40 * of the input and produces a number

41 *

42 * For those of you who are more mathem

43 * treats each character in the input name as a num

Name

Name	Value	Type
first	"Ada"	std::string
hashCode	0	int
last	"Lovelace"	std::string

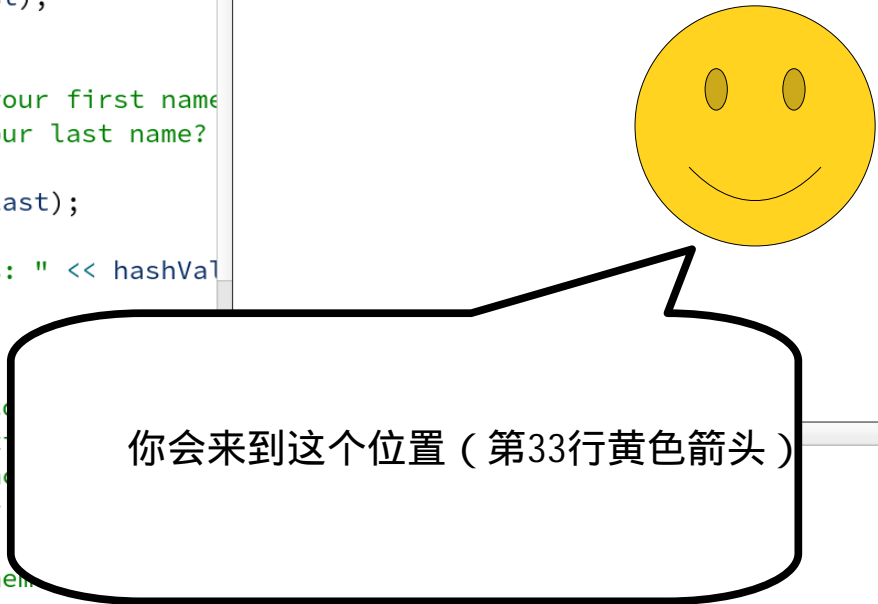
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x5555555b6782
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555555b616c
4	GThreadStd::run()			0x5555555b6476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

Number	Func	File	Line	Address	Condition	Ignore	Threads
1eHash.cpp	66		(all)

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

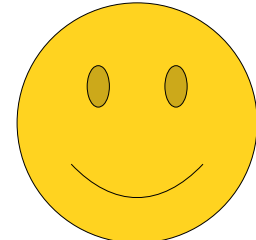
```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVa
34     return 0;
35 }
36
37 /* This is the actual function that c
38 * to talk more about what hash funct
39 * the meantime, think of it as a fun
40 * of the input and produces a number
41 *
42 * For those of you who are more math
43 * treats each character in the input
```

Debugger GDB for "NameHash" Threads: #12 Na

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x55...
2	studentMain	NameHash.cpp	31	0x55...
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55...
4	GThreadStd::run()			0x55...
5	??			0x7ff...
6	start thread	pthread create.c	463	0x7ff...

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Con

Views Threads (all)



注意黄色箭头已经来到 33 行了。
此处包含了一个 nameHash 的函数调用。
这就是该函数的实际调用位置！

由此，我们就知道程序是如何到达断点处的。

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

29 31 32 33 34 35 36 37 38 39 40 41 42 43

```
#include "simpio.h" // for getLine
using namespace std;

/* Prototype for the nameHash function. This lets u
 * in main and then define it later in the program.
 */
int nameHash(string first, string last);

int main() {
    string first = getLine("What is your first name
    string last = getLine("What is your last name?

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVa
    return 0;
}

/* This is th
 * to talk mo
 * the meanti
 * of the inp
 *
 * For those
 * treats each
```

Name

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

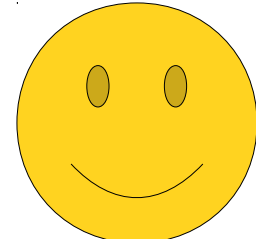
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funcnt	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...555555b6782			(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

通常来说，我们可以通过调用栈来查询，
程序是如何停在断点处的。



Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout <<
34     return
35 }
36
37 /* This is
38 * to talk
39 * the meant
40 * of the input and produces a number.
41 *
42 * For those of you who are more mathematically inc
43 * treats each character in the input name as a num
```

Name Value Type

first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

根据你使用的操作系统，studentMain 下面还会有一些额外的函数。

那些是什么呢？

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555b6782	1	...	g) ...eHash.cpp	66	...	5555555b6782		(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

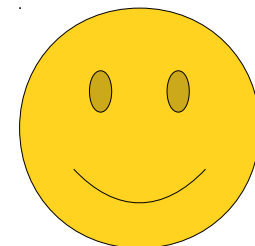
- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28 string first = getLine("What is your first name
29 string last = getLine("What is your last name?
30
31 int hashValue = nameHash(first, last);
32
33 cout <<
34 return
35 }
36
37 /* This is
38 * to talk
39 * the meant
40 * of the input and produces a number.
41 *
42 * For those of you who are more mathematically inc
43 * treats each character in the input name as a num

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



灰色标记的函数栈是我们运行程序时的一些辅助库函数。

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout <<
34     return
35 }
36
37 /* This is
38 * to talk
39 * the meant
40 * of the input and produces a number.
41 *
42 * For those of you who are more mathematically inc
43 * treats each character in the input name as a num
```

Name Value Type

first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

暂时不需要管那些。直接忽略即可。

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:22 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout <<
34     return
35 }
36
37 /* This is
38 * to talk
39 * the meant
40 * of the input and produces a number.
41 *
42 * For those of you who are more mathematically inc
43 * treats each character in the input name as a num
```

Name Value Type

first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

好了，再回到我们的 nameHash 函数。

还记得吗？双击 nameHash ！

完成后.....

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66		(all)
2	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
3	GThreadSt			0x5555555f9476								
4	??			0x7ffff6143d84								
5	start thread	pthread create.c	463	0x7ffff6257590								

Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp


```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

你又回到了刚才的位置。



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

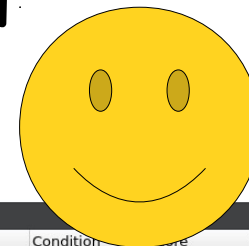
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the string */
60     for (int i = 0; i < last.length(); i++)
61     {
62         char ch = last[i];
63         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
64     }
65
66     return hashVal;
67 }
68
69
70
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

快速回顾下，我们学到了什么！



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

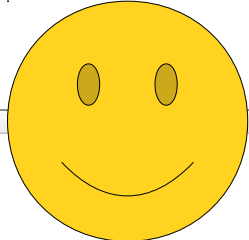
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

设置断点，可以让程序停在我们想要的位置。
设置断点的方式是，点击行号前的空白处。



Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

248 */

249 int nameHash(string first, string last){

250 /* This hashing scheme needs two prime numbers:

251 * prime. These numbers were chosen because the

252 * $2^{31} - kLargePrime - 1$.

253 */

254

255

256

257

258

259

260

261

262 for (char ch: first + last) {

263 /* Convert the input character to lower case

264 * lower-case letters are always less than

265 */

266 ch = tolower(ch);

267 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

268 }

269 return hashVal;

270 }

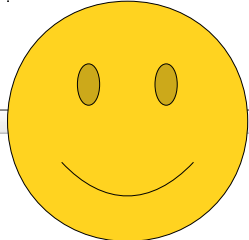
271

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

一旦程序执行到断点处，Qt 会弹出调试面板。



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

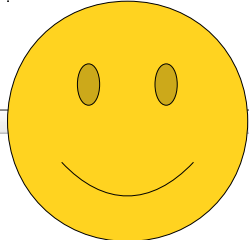
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"A" 65 0x41	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

黄色箭头告诉我们，此时我们所处在的程序中的位置。



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

调用栈可以告诉我们，程序是如何到达这个断点的。

for (char ch: first + last) {
/* Convert the input character to lower case
* lower-case letters are always less than
*/
ch = tolower(ch);
hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
}

return hashVal;

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), Qt...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

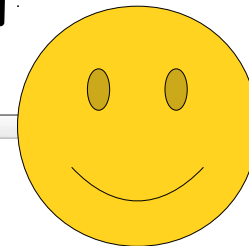
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
55
56
57
58
59
60
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

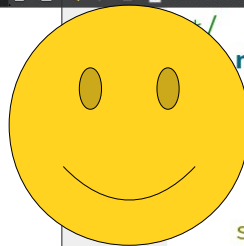
Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

接下来，让我们来看看如何查看函数中的变量。



看看右上角的面板！



Qt Creator IDE interface showing the NameHash.cpp file and the Debug Console.

Code Snippet:

```
nameHash(string first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because the
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Variable Inspector Panel:

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger Console:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

这里面列出的是当前函数内的局部变量。

Activities Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

Welcome

Edit

Design

Debug

Projects

Help

NameHash

Debug

```
1 // NameHash.cpp
2
3 #include <string>
4 #include <string_view>
5 #include <algorithm>
6 #include <cctype>
7
8 using namespace std;
9
10 /* This hashing scheme needs two prime numbers:
11  * prime. These numbers were chosen because the
12  *  $2^{31} - kLargePrime - 1$ .
13  */
14 static const int kLargePrime = 16908799;
15 static const int kSmallPrime = 127;
16
17 int hashVal = 0;
18
19 /* Iterate across all the characters in the first
20  * name, updating the hash at each step.
21  */
22 for (char ch: first + last) {
23     /* Convert the input character to lower case
24     * lower-case letters are always less than
25     */
26     ch = tolower(ch);
27     hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
28 }
29 return hashVal;
30 }
```

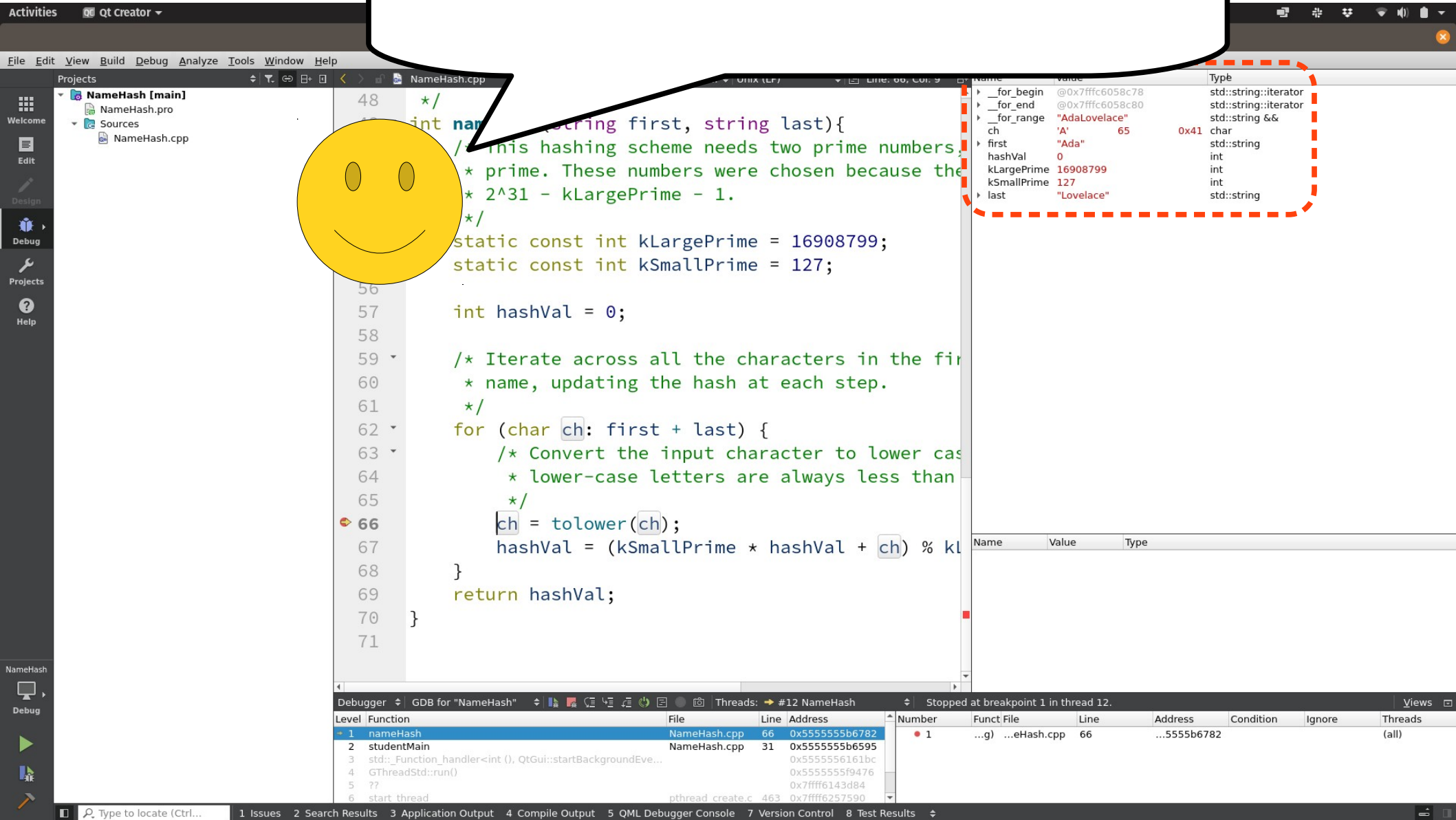
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'A' 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

根据不同的操作系统，名称可能不同。
但我们只关注类似 ch 和 hashVal 这样的名称



The image shows the Qt Creator IDE with a C++ project named "NameHash". The main window displays the source file "NameHash.cpp". The code defines a function `int nameHash(const string& first, const string& last)` that calculates a hash value for a string. The function uses two prime numbers, `kLargePrime` and `kSmallPrime`, and iterates over the characters of the input string to calculate the hash. A yellow smiley face is overlaid on the code. A red dashed box highlights the variable declarations in the code and the corresponding entries in the Variable View.

```
48  */
49  int nameHash(const string& first, string last){
50      /* This hashing scheme needs two prime numbers:
51       * prime. These numbers were chosen because the
52       * 2^31 - kLargePrime - 1.
53       */
54      static const int kLargePrime = 16908799;
55      static const int kSmallPrime = 127;
56
57      int hashVal = 0;
58
59      /* Iterate across all the characters in the first
60       * name, updating the hash at each step.
61       */
62      for (char ch: first + last) {
63          /* Convert the input character to lower case
64           * lower-case letters are always less than
65           */
66          ch = tolower(ch);
67          hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68      }
69      return hashVal;
70  }
71
```

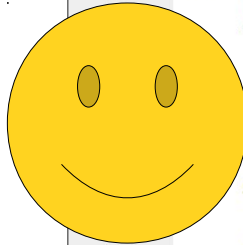
The Variable View on the right shows the following variables:

Name	Value	Type
<code>__for_begin</code>	<code>@0x7fffc6058c78</code>	<code>std::string::iterator</code>
<code>__for_end</code>	<code>@0x7fffc6058c80</code>	<code>std::string::iterator</code>
<code>__for_range</code>	<code>"AdaLovelace"</code>	<code>std::string &&</code>
<code>ch</code>	<code>'A'</code>	<code>char</code>
<code>first</code>	<code>"Ada"</code>	<code>std::string</code>
<code>hashVal</code>	<code>0</code>	<code>int</code>
<code>kLargePrime</code>	<code>16908799</code>	<code>int</code>
<code>kSmallPrime</code>	<code>127</code>	<code>int</code>
<code>last</code>	<code>"Lovelace"</code>	<code>std::string</code>

The Debugger window at the bottom shows the execution state. It is stopped at breakpoint 1 in thread 12. The call stack shows the following frames:

Level	Function	File	Line	Address
1	<code>nameHash</code>	<code>NameHash.cpp</code>	<code>66</code>	<code>0x555555b6782</code>
2	<code>studentMain</code>	<code>NameHash.cpp</code>	<code>31</code>	<code>0x555555b6595</code>
3	<code>std::_Function_handler<int (), QtGui::startBackgroundEve...</code>			<code>0x5555556161bc</code>
4	<code>GThreadStd::run()</code>			<code>0x5555555f9476</code>
5	<code>??</code>			<code>0x7ffff6143d84</code>
6	<code>start_thread</code>	<code>pthread_create.c</code>	<code>463</code>	<code>0x7ffff6257590</code>

忽略那些奇奇怪怪的名字，只关注我们熟悉的名字就好了



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65 0x41
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

Name	Value	Type
------	-------	------

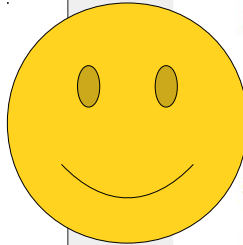
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

比如，你可以看到 kLargePrime 和 kSmallPrime 跟我们程序中定义的一致。



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Debugger variables:

__for_end	@0x7fffc6058c80
__for_range	"AdaLovelace"
ch	'A' 65 0x41
first	"Ada"
hashVal	0
kLargePrime	16908799
kSmallPrime	127
last	"Lovelace"

Debugger variables:

type	std::string::iterator
	std::string::iterator
	std::string &&
	char
	std::string
	int
	int
	std::string

Name	Value	Type
------	-------	------

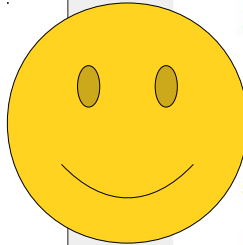
Debugger: GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Number	Funct	File	Line	Address	Condition	Ignore	Threads
1g) ...eHash.cpp	66	...5555b6782			(all)

Type to locate (Ctrl+L) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

我们也会发现 hashVal
的值此时还是 0



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers;
    * prime. These numbers were chosen because they
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

Name	Value	Type
------	-------	------

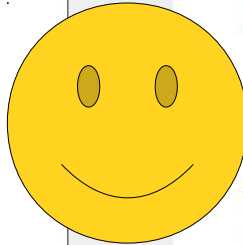
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

当我们一步一步调试程序时，我们会发现这些值的变化。



Qt Creator interface showing the NameHash.cpp file and the GDB debugger.

Code Snippet (NameHash.cpp):

```
48 */
49 int nameHash(const string& first, const string& last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    * 2^31 - kLargePrime - 1.
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Debugger Variables:

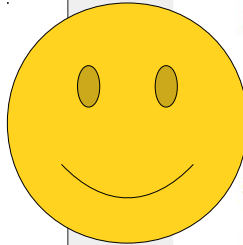
Name	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger Console:

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x5555555b6782
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Debugger Status: Stopped at breakpoint 1 in thread 12.

先看看这个 for 循环。



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case
        * lower-case letters are always less than
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

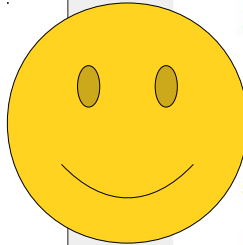
Variable	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger ▾ GDB for "NameHash" ▾ [Icons] Threads: ▸ #12 NameHash ▾ Stopped at breakpoint 1 in thread 12. ▾ Views ▾												
Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

这是一个 **range-based for** 循环，意思是
“对 first + last 中的每一个字符，执行相应的操作。”



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Debugger variables:

__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

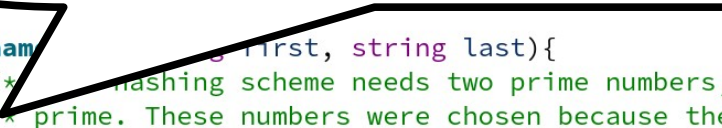
Name	Value	Type
------	-------	------

Debugger: GDB for "NameHash" | Threads: #12 NameHash | Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

回忆一下，我们刚刚输入的名字是
Ada Lovelace

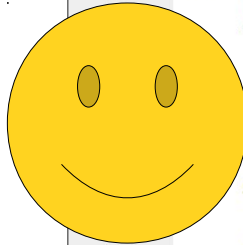


```
for (char ch: first + last) {  
    /* Convert the input character to lower case  
     * lower-case letters are always less than  
     */  
    ch = tolower(ch);  
    hashVal = (kSmallPrime * hashVal + ch) % kSmallPrime;  
}  
return hashVal;
```

Name	Value	Type
------	-------	------

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x5555555b6782	1g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc								
4	GThreadStd::run()			0x5555555f947c								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

如果我们查看 ch 的当前值，会发现此时的值是 A。
这就是 Ada Lovelace 的第一个字母。



```
48  */
49  int nameHash(const string& first, string& last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

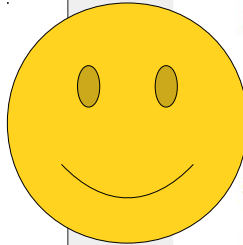
    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Variable view showing the state of variables during execution:

Variable	Value	Type
ch	'A'	char
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Debugger - GDB for "NameHash"										Stopped at breakpoint 1 in thread 12.		
Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1	...	g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

非常好！总结一下：
我们已经知道程序所在的位置（第66行），
还知道程序是如何从 main 函数执行到 nameHash 的，
还知道此刻程序中的变量值。



```
48  */
49  int nameHash(const string& first, const string& last){
    /* This hashing scheme needs two prime numbers:
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string,
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

```
__for_end @0x7ffff6058c80
__for_range "AdaLovelace"
ch 'A' 65
first "Ada"
hashVal 0
kLargePrime 16908799
kSmallPrime 127
last "Lovelace"
```

```
type
std::string::iterator
std::string::iterator
std::string &&
char
std::string
int
int
int
std::string
```

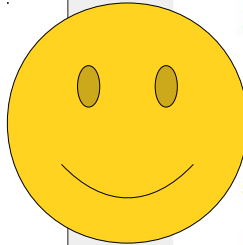
Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

接下来，我们再做一些更酷的事情。
我们将一行一行地执行程序，并观察每一步发生的变化。



```
48  */
49  int nameHash(const string& first, string last){
    /* This hashing scheme needs two prime numbers.
    * prime. These numbers were chosen because they are
    *  $2^{31} - kLargePrime - 1$ .
    */
    static const int kLargePrime = 16908799;
    static const int kSmallPrime = 127;

    int hashVal = 0;

    /* Iterate across all the characters in the first string
    * name, updating the hash at each step.
    */
    for (char ch: first + last) {
        /* Convert the input character to lower case.
        * lower-case letters are always less than 128.
        */
        ch = tolower(ch);
        hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
    }
    return hashVal;
}
```

Variable	Value	Type
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string::iterator
ch	'A'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	66	0x555555b6782	1eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 `*/`

49 `int nameHash(string first, string last){`

50 `/* This hashing scheme needs two prime numbers;`

51 `* prime. These numbers were chosen because the`

52 `* $2^{31} - kLargePrime - 1$.`

53 `*/`

54 `static const int kLargePrime = 16908799;`

55 `static const int kSmallPrime = 127;`

56

57 `int hashVal = 0;`

58

59 `/* Iterate across all the characters in the fir`

60 `* name, updating the hash at each step.`

61 `*/`

62 `for (ch`

63 `/*`

64 `*`

65 `*/`

66 `ch`

67 `has`

68 `}`

69 `return hashVal;`

70 `}`

71

Debugger GDB for "NameHash"

Threads: #12 NameHash


Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Type to locate (Ctrl...

在调用栈的上方，有这么几个按钮。



Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

```
48  */
49  int nameHash(string first, string last){
50  /* This hashing scheme needs two prime numbers;
51   * prime. These numbers were chosen because the
52   *  $2^{31} - kLargePrime - 1$ .
53   */
54   static const int kLargePrime = 16908799;
55   static const int kSmallPrime = 127;
56
57   int hashVal = 0;
58
59   /* Iterate across all the characters in the first
60    * name, updating the hash at each step.
61   */
62   for (char ch : first)
63   {
64       /*
65        *
66        */
67       hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68   }
69   return hashVal;
70 }
71 }
```

Debugger

GDB for "NameHash"

Threads: #12 NameHash

Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590

Debugger

Number

Funct

File

Line

Address

Condition

Ignore

Threads

1

...g

...eHash.cpp

66

...5555b6782

(all)

Type

Value

std::string::iterator

@0x7fffc6058c78

std::string::iterator

@0x7fffc6058c80

std::string &&

"AdaLovelace"

ch

'A'

65

0x41

char

first

"Ada"

std::string

hashVal

0

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

这些按钮可以让你恢复程序、停止程序、单步执行程序.....

Activities Qt Creator

Jan 4 3:30 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (ch

63 /*

64 *

65 */

66 ch

67 has

68 }

69 return hashVal;

70 }

71 }

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start thread	pthread_create.c	463	0x7ffff6257590


1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Name

Name	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	"A" 65	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

把鼠标移动到第三个按钮上，不要点击。

一两秒后，会显示按钮的名称 "step over"



Activities Qt Creator Jan 4 3:30 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```


Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x5555555b6782
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThread...>::operator()()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

先确保你的鼠标在 "Step Over" 上，而不是 "Step Into" 或 "Step Out"。

确认后点一下试试！当你点完后.....



Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers:

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7fffc6058c78 std::string::iterator

__for_end @0x7fffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

NameHash

Debug

Debugger GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), Q

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7ffff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

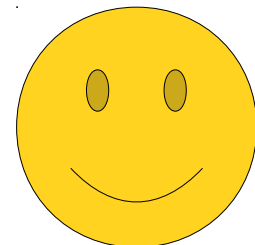
3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



...黄色箭头来到了下一行 (line 67)

ActivitiesQt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

FileEditViewBuildDebugAnalyzeToolsWindowHelp

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

DebuggerGDB for "NameHash"

LevelFunction

1nameHash

2studentMain

3std::Function_handler<int (), Q

4GThreadStd::run()

5??

6start thread

NameValue

Value

Type

for_begin@0x7ffc6058c78std::string::iterator

for_end@0x7ffc6058c80std::string::iterator

for_range"AdaLovelace"std::string &&

ch'a'970x61char

first"Ada"std::string

hashVal0int

kLargePrime16908799int

kSmallPrime127int

last"Lovelace"std::string

Views

Threads

Call

Type to locate (Ctrl...

1Issues

2Search Results

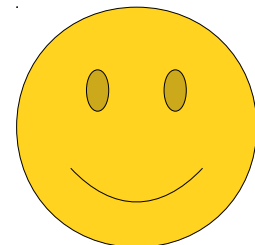
3Application Output

4Compile Output

5QML Debugger Console

7Version Control

8Test Results



非常好！已经有一些东西改变了。让我们一起来看看。

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % k

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7ffc6058c78 std::string::iterator

__for_end @0x7ffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger

GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), G

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7fff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

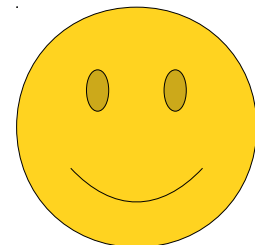
3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



首先，黄色箭头此时在 67 行。

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers;

51 * prime. These numbers were chosen because the

52 * $2^{31} - \text{kLargePrime} - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

for_begin @0x7fffc6058c78 std::string::iterator

for_end @0x7fffc6058c80 std::string::iterator

for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger

GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), Q

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7ffff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

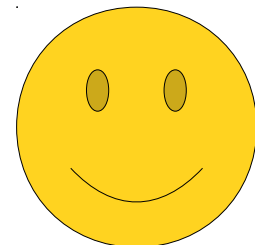
3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



你已经成功地运行了一行代码，相当完美！

Activities

Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

*/

int nameHash(string first, string last){

/* This hashing scheme needs two prime numbers.

* prime. These numbers were chosen because the

* $2^{31} - \text{kLargePrime} - 1$.

*/

static const int kLargePrime = 16908799;

static const int kSmallPrime = 127;

int hashVal = 0;

*/

Iterate across all the characters in the first

* name, updating the hash at each step.

*/

for (char ch: first + last) {

/* Convert the input character to lower case

* lower-case letters are always less than

*/

ch = tolower(ch);

hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;

}

return hashVal;

}

Name

Value

Type

__for_begin

@0x7ffc6058c78

std::string::iterator

__for_end

@0x7ffc6058c80

std::string::iterator

__for_range

"AdaLovelace"

std::string &&

ch

'a'

97

0x61

char

first

"Ada"

std::string

hashVal

0

int

kLargePrime

16908799

int

kSmallPrime

127

int

last

"Lovelace"

std::string

Debugger

GDB for "NameHash"

Level

Function

1

nameHash

2

studentMain

3

std::_Function_handler<int (), QThreadStd::run()

4

GThreadStd::run()

5

??

6

start thread

pthread create.c

463

0x7fff6257590

Type to locate (Ctrl...

1 Issues

2 Search Results

3 Application Output


4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results

那么这行代码做了什么呢？



ActivitiesQt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

FileEditViewBuildDebugAnalyzeToolsWindowHelp

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers:

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7fffc6058c78 std::string::iterator

__for_end @0x7fffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 0 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

DebuggerGDB for "NameHash"

LevelFunction

1 nameHash

2 studentMain

3 std::_Function_handler<int (), G

4 GThreadStd::run()

5 ??

6 start thread

Type to locate (Ctrl...

1 Issues

2 Search Results

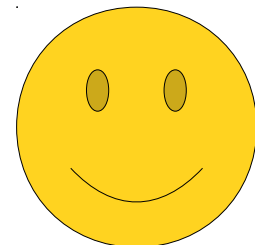
3 Application Output

4 Compile Output

5 QML Debugger Console

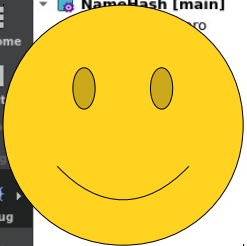
7 Version Control

8 Test Results



这行代码把 ch 转换成了小写。函数 tolower 接收一个字符，并返回它的小写形式。然后覆盖 ch 值，这样就得到了一个小写的版本。

当然，我们也可以通过函数值面板，查看这个过程。



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers:
52      * prime. These numbers were chosen because they
53      *  $2^{31} - kLargePrime - 1$ .
54      */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57
58     int hashVal = 0;
59
60     /* Iterate across all the characters in the first
61      * name, updating the hash at each step.
62      */
63     for (char ch: first + last) {
64         /* Convert the input character to lower case
65          * lower-case letters are always less than
66          */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```

	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

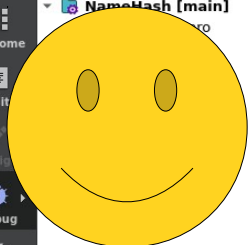
Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

注意到 ch 的值已经从 A 变成了 a



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers;
52     * prime. These numbers were chosen because the
53     *  $2^{31} - kLargePrime - 1$ .
54     */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71 }
```

	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"Ada Lovelace"	std::string &&
ch	'a' 97	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

如果你细心的话，可以发现变量一旦发生改变，字体颜色就会标记成红色。



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers;
52     * prime. These numbers were chosen because the
53     *  $2^{31} - kLargePrime - 1$ .
54     */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60     * name, updating the hash at each step.
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71 }
```

	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"Ada Lovelace"	std::string &&
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1		...g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

这就表明变量值在刚刚那一步中被改变了，非常的方便。



```
50 int nameHash(string first, string last){
51     /* This hashing scheme needs two prime numbers;
52     * prime. These numbers were chosen because the
53     *  $2^{31} - kLargePrime - 1$ .
54     */
55     static const int kLargePrime = 16908799;
56     static const int kSmallPrime = 127;
57
58     int hashVal = 0;
59
60     /* Iterate across all the characters in the first
61     * name, updating the hash at each step.
62     */
63     for (char ch: first + last) {
64         /* Convert the input character to lower case
65         * lower-case letters are always less than
66         */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```

	Value	Type
__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"Ada Lovelace"	std::string &&
ch	'a'	char
first	"Ada"	std::string
hashVal	0	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Name	Value	Type
------	-------	------

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1	...	g) ...eHash.cpp	66		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

start
if (first != 0 && last != 0) {
 int hashVal = 0;

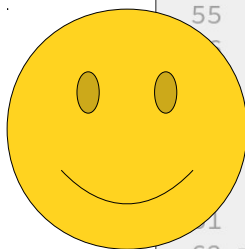
 /* Iterate across all the characters in the first and last names, updating the hash at each step.
 */
 for (char ch: first + last) {
 /* Convert the input character to lower case. Note that
 * lower-case letters are always less than upper-case letters.
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
 }
 return hashVal;
}

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x5555555b6790
2	studentMain	NameHash.cpp	31	0x5555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

接下来，再看看第 67 行。



Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

54
55 start = first + last - 1; // first name = 127;
56
57 int hashVal = 0;
58
59 /* Iterate across all the characters in the first
60 * name, updating the hash at each step.
61 */
62 for (char ch: first + last) {
63 /* Convert the input character to lower case
64 * lower-case letters are always less than
65 */
66 ch = tolower(ch);
67 hashVal = (kSmallPrime * hashVal + ch) % kMaxVal;
68 }
69 return hashVal;
70 }
71

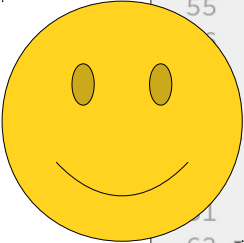
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x5555555b6790	1eHash.cpp	66	...			(all)
2	studentMain	NameHash.cpp	31	0x5555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

不骗你，这是一行比较紧凑的代码。
它执行了一些神奇的数学计算。



Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

start
int hashVal = 0;

/* Iterate across all the characters in the first
* name, updating the hash at each step.
*/
for (char ch: first + last) {
 /* Convert the input character to lower case
 * lower-case letters are always less than
 */
 ch = tolower(ch);
 hashVal = (kSmallPrime * hashVal + ch) % kSmallPrime;
}
return hashVal;

但是我們不必關注那些細節，這裡我們只需要知道
計算出的結果保存在 hashVal 就好了。

67 hashVal = (kSmallPrime * hashVal + ch) % kSmallPrime;

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x555555b6790
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Number	Func	File	Line	Address	Condition	Ignore	Threads
1eHash.cpp	66		(all)

Type to locate (Ctrl...)

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:42 PM NameHash.cpp @ NameHash [main] - Qt Creator

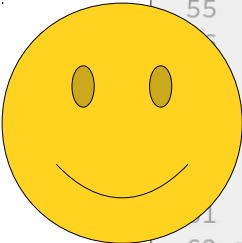
File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

让我们执行一下该行代码，看看会发生什么。



```
54
55
56 start = 127;
57
58 int hashVal = 0;
59
60 /* Iterate across all the characters in the first
61 * name, updating the hash at each step.
62 */
63 for (char ch: first + last) {
64     /* Convert the input character to lower case
65     * lower-case letters are always less than
66     */
67     ch = tolower(ch);
68     hashVal = (kSmallPrime * hashVal + ch) % kMaxVal;
69 }
70 return hashVal;
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790	1eHash.cpp	66	...			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator

Jan 4 3:42 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

48 `*/`

49 `int nameHash(string first, string last){`

50 `/* This hashing scheme needs two prime numbers;`

51 `* prime. These numbers were chosen because the`

52 `* $2^{31} - kLargePrime - 1$.`

53 `*/`

54 `static const int kLargePrime = 16908799;`

55 `static const int kSmallPrime = 127;`

56

57 `int hashVal = 0;`

58

59 `/* Iterate across all the characters in the fir`

60 `* name, updating the hash at each step.`

61

62

63

64

65

66

67 `return hashVal;`

68 `}`

69

70


71

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	67	0x555555b6790								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::function_handler<int (), QtGui::startB...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

先把鼠标放在“Step Over”按钮上，
确认名称正确了再执行。
点击按钮后...



Activities

Qt Creator

Jan 4 3:48 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers:

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7fffc6058c78 std::string::iterator

__for_end @0x7fffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 97 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger GDB for "NameHash"

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), QtGui::startBackgroundDeve...

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7ffff6257590

1 Issues

2 Search Results

3 Application Output

4 Compile Output

5 QML Debugger Console

7 Version Control

8 Test Results



... 你会来到这个界面！

Activities

Qt Creator

Jan 4 3:48 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

NameHash [main]

NameHash.pro

Sources

NameHash.cpp

48 */

49 int nameHash(string first, string last){

50 /* This hashing scheme needs two prime numbers:

51 * prime. These numbers were chosen because the

52 * $2^{31} - kLargePrime - 1$.

53 */

54 static const int kLargePrime = 16908799;

55 static const int kSmallPrime = 127;

56

57 int hashVal = 0;

58

59 /* Iterate across all the characters in the fir

60 * name, updating the hash at each step.

61 */

62 for (char ch: first + last) {

63 /* Convert the input character to lower cas

64 * lower-case letters are always less than

65 */

66 ch = tolower(ch);

67 hashVal = (kSmallPrime * hashVal + ch) % kL

68 }

69 return hashVal;

70 }

71 }

Name

Value

Type

__for_begin @0x7fffc6058c78 std::string::iterator

__for_end @0x7fffc6058c80 std::string::iterator

__for_range "AdaLovelace" std::string &&

ch 'a' 97 0x61 char

first "Ada" std::string

hashVal 97 int

kLargePrime 16908799 int

kSmallPrime 127 int

last "Lovelace" std::string

Debugger

GDB for "NameH

Level Function

1 nameHash

2 studentMain

3 std::_Function_handler<int (), QtGui::startBackgroundDeve...

4 GThreadStd::run()

5 ??

6 start thread pthread create.c 463 0x7ffff6257590

1 Issues

2 Search Results

3 Application Output

4 Compile Output

5 QML Debugger Console

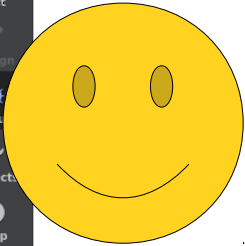
7 Version Control

8 Test Results



看一看发生了什么！

首先 hashVal 的值变成了 97
因为这个值是红色的，所以我们很容易发现它。
其他值的颜色没变化，所以它们的值也没有改变。



```
51  /* This hashing scheme needs two prime numbers;
52  * prime. These numbers were chosen because the
53  *  $2^{31} - kLargePrime - 1$ .
54  */
55  static const int kLargePrime = 16908799;
56  static const int kSmallPrime = 127;
57
58  int hashVal = 0;
59
60  /* Iterate across all the characters in the first
61  * name, updating the hash at each step.
62  */
63  for (char ch: first + last) {
64      /* Convert the input character to lower case
65      * lower-case letters are always less than
66      */
67      ch = tolower(ch);
68      hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69  }
70  return hashVal;
71 }
```

Name	Value	Type
ch	'a'	char
first	@0x7fffc6058c78	std::string::iterator
last	@0x7fffc6058c80	std::string::iterator
hashVal	97	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb	1	...	g) ...eHash.cpp	66	...	5555b6782		(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime$ 
53     */
54     static
55     static
56
57     int has
58
59     /* Iter
60     * name
61     */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kL
68     }
69     return hashVal;
70 }
71
```

Variable Inspector:

Name	Value	Type
__for_begin	@0x7ffc6058c78	std::string::iterator
__for_end	@0x7ffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'a' 97	char
first	"Ada"	std::string
hashVal	97	int
kLargePrime	16908799	int
kSmallPrime	127	int

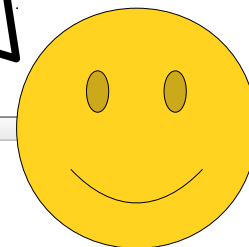
Debugger: GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb	1	...	g) ...eHash.cpp	66	...5555b6782			(all)
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

其次，黄色箭头告诉我们，程序回到了 for 循环的顶部。
这也就意味着，该行是我们接下来将要执行的位置。



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

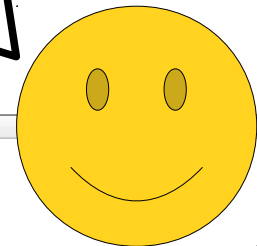
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the string.
60      * nameHash will iterate over the characters in the string.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

非常完美！我们刚刚执行完了一层 for 循环。



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

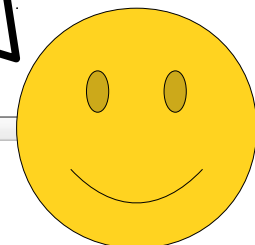
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the strings.
60      * nameHash will iterate over the characters of the strings.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

继续加油！



Activities Qt Creator Jan 4 3:48 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

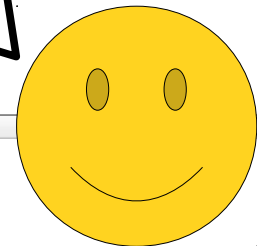
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters of the string
60      * nameHash.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than
65          * 128.
66          */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThread...>::operator()()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7fffff143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

重复刚才的动作，继续点击“Step Over”



Activities Qt Creator Jan 4 3:50 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

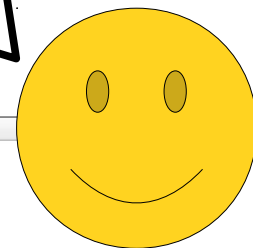
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the first and last names.
60      * nameHash will iterate over the characters in the first
61      * and last names.
62      */
63     for (char ch: first + last) {
64         /* Convert the input character to lower case.
65          * lower-case letters are always less than 128.
66          */
67         ch = tolower(ch);
68         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
69     }
70     return hashVal;
71 }
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop...>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

此时我们看到 ch 的值变成了 d，也就是 Ada 的第二个字符。



Activities Qt Creator Jan 4 3:50 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

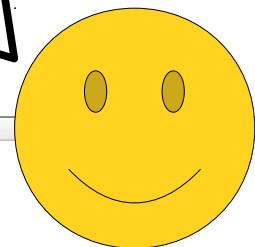
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the string.
60      * nameHash will return the hash value for the string.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped at breakpoint 1 in thread 12.

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	66	0x555555b6782
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundThreadEventLoop...			0x5555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

再次点击 "Step Over" 执行该行代码。



Activities Qt Creator Jan 4 3:52 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

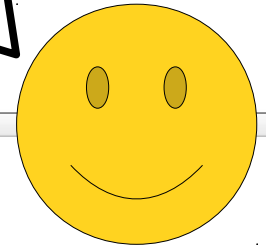
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 100;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the first and last names.
60      * nameHash will calculate the hash value for each character.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x555555b6790
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEventLoop()>::operator()			0x55555556161bc
4	GThreadStd::run()			0x5555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

你应该处在这个位置。注意，没有一个值发生改变。
显然，d 转换成小写还是 d。



Activities Qt Creator Jan 4 3:52 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

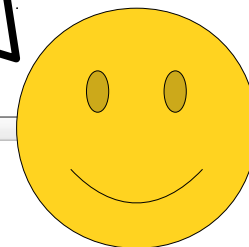
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate over the characters in the string.
60      * nameHash will calculate the hash value for the string.
61      */
62     for (char ch: first + last) {
63         /* Convert the input character to lower case.
64          * lower-case letters are always less than 128.
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	67	0x555555b6790
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEvent...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

好了，再点击一次 "step Over"



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - kLargePrime - 1$ .
53        */
54     static const int kLargePrime = 16908799;
55
56
57
58 }
59 return hashVal;
60 }
61
```


Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Look here!

此时你应该处在这个位置。我们遮住了 hashVal 的值。
如果你操作正确，你应该知道这个值是多少。
对于这个特殊的值，请记录下来，在作业中提交。



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

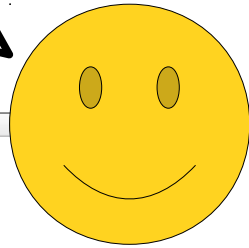
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64          * lower-case letters are always less than
65          */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

Name	Value	Type
__for_begin	@0x7ffff6058c78	std::string::iterator
__for_end	@0x7ffff6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'd' 100	char
first	"Ada"	std::string
hashVal	????	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

最后，我们还会介绍两个小技巧，在调试中将会非常有用。



Activities Qt Creator Jan 4 3:53 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

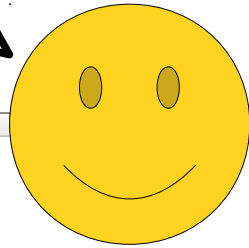
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54
61
62     for (char ch: first + last) {
63         /* Convert the input character to lower case
64         * lower-case letters are always less than
65         */
66         ch = tolower(ch);
67         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	nameHash	NameHash.cpp	62	0x555555b67cb
2	studentMain	NameHash.cpp	31	0x555555b6595
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc
4	GThreadStd::run()			0x555555f9476
5	??			0x7ffff6143d84
6	start_thread	pthread_create.c	463	0x7ffff6257590

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

首先，再次点击之前设置过的断点。如果你这么做的话.....



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51     * prime. These numbers were chosen because the
52     *  $2^{31} - kLargePrime - 1$ .
53     */
54
55     for (char ch: first + last) {
56         /* Convert the input character to lower case
57         * lower-case letters are always less than
58         */
59         ch = tolower(ch);
60         hashVal = (kSmallPrime * hashVal + ch) % kLargePrime;
61     }
62     return hashVal;
63 }
```

.....断点将会消失。

此时，如果我们再次在 Debug 模式下运行程序，程序将不会停止。

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Func	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

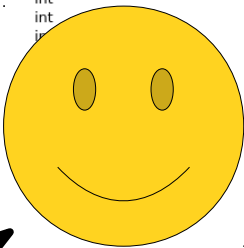
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555555616c								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+K) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

再回来看看这些按钮。



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because they are
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Name Value Type

__for_begin	@0x7fffc6058c78	std::string::iterator
__for_end	@0x7fffc6058c80	std::string::iterator
__for_range	"AdaLovelace"	std::string &&
ch	'd' 100	char
first	"Ada"	std::string
hashVal	????	int
kLargePrime	16908799	int
kSmallPrime	127	int
last	"Lovelace"	std::string

把鼠标悬停在最右边的按钮上。

此时会提示 "step out"

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+K) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers:
51      * prime. These numbers were chosen because the
52      *  $2^{31} - kLargePrime - 1$ .
53      */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60      * name, updating the hash at each step.
61      */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

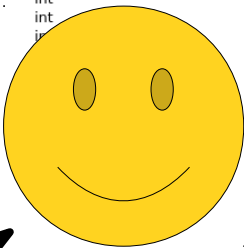
Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackgroundEve...			0x5555556161bc								
4	GThreadStd::run()			0x555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+K)

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

先不要点击。
当你决定点击的时候，程序会执行完 nameHash 函数，并返回值。



Activities Qt Creator Jan 4 3:57 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

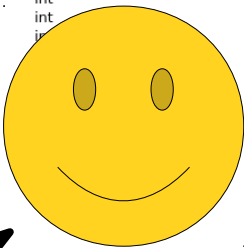
```
48 */
49 int nameHash(string first, string last){
50     /* This hashing scheme needs two prime numbers;
51        * prime. These numbers were chosen because the
52        *  $2^{31} - kLargePrime - 1$ .
53        */
54     static const int kLargePrime = 16908799;
55     static const int kSmallPrime = 127;
56
57     int hashVal = 0;
58
59     /* Iterate across all the characters in the first
60        * name, updating the hash at each step.
61        */
62     for (char ch : first)
63     {
64         /*
65          *
66          */
67         hashVal = (hashVal * kLargePrime + ch) % kSmallPrime;
68     }
69     return hashVal;
70 }
71
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address	Number	Funct	File	Line	Address	Condition	Ignore	Threads
1	nameHash	NameHash.cpp	62	0x555555b67cb								
2	studentMain	NameHash.cpp	31	0x555555b6595								
3	std::_Function_handler<int (), QtGui::startBackground...			0x55555556161bc								
4	GThreadStd::run()			0x5555555f9476								
5	??			0x7ffff6143d84								
6	start_thread	pthread_create.c	463	0x7ffff6257590								

Type to locate (Ctrl+Shift+F) 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

现在尝试点击一下吧！当你正确操作后...



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

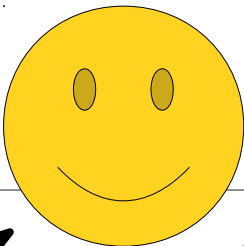
```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVa
34     return 0;
35 }
36
37 /* This is
38 * to talk
39 * the mean
40 * of the i
41 *
42 * For thos
43 * treats e
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

... 你会来到这个界面。



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21

```
#include "simpio.h" // for getLine
using namespace std;

/* Prototype for the nameHash function. This lets u
 * in main and then define it later in the program.
 */
int nameHash(string first, string last);

int main() {
    string first = getLine("What is your first name
    string last = getLine("What is your last name?

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVa
    return 0;
}

/* This is
 * to talk
 * the mean
 * of the i
 *
 * For thos
 * treats e
```

31

first "Ada" std::string
hashValue 0 int
last "Lovelace" std::string

returned value 1967457 int

让我们分析一下。我们究竟处在哪里？

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address	Number	Funcnt	File	Line	Address	Condition	Ignore	Threads
1	studentMain	NameHash.cpp	31	0x555555b6595								
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc								
3	GThreadStd::run()			0x5555555f9476								
4	??			0x7ffff6143d84								
5	start_thread	pthread_create.c	463	0x7ffff6257590								
6	clone	clone.S	95	0x7ffff5e30223								

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVa
34     return 0;
35 }
36
37 /* This is
38 * to talk
39 * the mean
40 * of the i
41 *
42 * For thos
43 * treats e
```

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

Debugger Variables:

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

Debugger Console:

returned value 1967457 int

黄色箭头表明我们已经回到了 main 函数。
很棒！

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name? ");
29     string last = getLine("What is your last name? ");
30     int hashValue = nameHash(first, last);
31
32     cout << "The hash of your name is: " << hashValue << endl;
33     return 0;
34 }
35
36
37 /* This is the
38 * to talk m
39 * the meant
40 * of the in
41 *
42 * For those
43 * treats ea
```

Debugger GDB for "NameHash"

Level Function

- 1 studentMain
- 2 std::Function_handler<int (), QtG
- 3 GThreadStd::run()
- 4 ??
- 5 start_thread
- 6 clone

clone.S 95 0x7ffff5e30223

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

Name	Value	Type
first	"Ada"	std::string
hashValue	0	int
last	"Lovelace"	std::string

returned value 1967457 int

在这个框内我们可以看到 nameHash 函数的返回值 1967457。感谢调试器！

(A note: it seems like on some Macs, this number doesn't display. Don't worry if you don't see it - just continue on as usual.)

Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
19 #include "simpio.h" // for getLine
20 using namespace std;
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVa
34     return 0;
35 }
36
37 /* This is the actual function that computes the ha
38 * to talk more about what hash functions do later
39 the meantime, think of it as a function that scr
40 the input and produces a number.
```

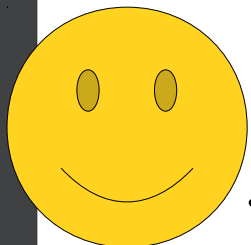
Name	Value	Type
hashValue	0	int

returned value 1967457 int

Name	Value	Type
------	-------	------

Line Address Condition Ignore Threads

Type to locate (Ctrl...



但是上面的 hashValue 却没有存储 1967457 , 虽然已经返回了

(You might see a number other than 0 on your system - that's okay.)



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for nameHash()
23  * in main and
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first;
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash value.
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

Line: 31, Col: 5

Name first Value "Ada" Type std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x5555555b6595
2	std::Function_handler<int (>, QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

但是看起来，我们想让 hashValue
的值等于 nameHash 的返回值。
发生了什么呢？

returned value 1967457 int

Name	Value	Type
------	-------	------



Activities Qt Creator Jan 4 4:02 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

Welcome Edit Design Debug Projects Help

```
19 #include "simpio.h"
20 using namespace std;
21
22 /* Prototype for nameHash()
23  * in main and
24  */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name? ");
29     string last = getLine("What is your last name? ");
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashValue << endl;
34     return 0;
35 }
36
37 /* This is the actual function that computes the hash value.
38  * to talk more about what hash functions do later
39  * the meantime, think of it as a function that scans
40  * of the input and produces a number.
41  *
42  * For those of you who are more mathematically inclined,
43  * treats each character in the input name as a number.
```

Line: 31, Col: 5

Name	Value	Type
first	"Ada"	std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (>, QtGui::startBackgroundEve...			0x5555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

实际上，这非常的酷！

returned value 1967457 int

Name	Value	Type
------	-------	------



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

29 string last = getLine("What is your last name?");

31 **int hashValue = nameHash(first, last);**

33 cout << "The hash of your name is: " << hashValue;

34 return 0;

35 }

37 /* This is the actual function that computes the hash

38 * to talk more about what hash functions do later

39 * the meantime, think of it as a function that scans

40 * of the input and produces a number.

41 *

42 * For those of you who are more mathematically inclined,

43 * treats each character in the input name as a number.

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x5555555b6595
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

函数 nameHash 确实返回了值。
但此时我们是单步执行的，所以 hashValue 还没有被赋值。

returned value 1967457 int

Name	Value	Type
------	-------	------



Activities Qt Creator

Jan 4 4:02 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21 #include "simpio.h"

22 using namespace std;

23 /* Prototype for nameHash function

24 * in main and

25 */

26 int nameHash(string first, string last);

27 int main() {

28 string first = getLine("What is your first name");

29 string last = getLine("What is your last name?");

30

31 int hashValue = nameHash(first, last);

32

33 cout << "The hash of your name is: " << hashValue;

34 return 0;

35 }

36

37 /* This is the actual function that computes the hash value

38 * to talk more about what hash functions do later

39 * the meantime, think of it as a function that scans

40 * of the input and produces a number.

41 *

42 * For those of you who are more mathematically inclined,

43 * treats each character in the input name as a number.

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "function-finished".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	31	0x555555b6595
2	std::Function_handler<int (>, QtGui::startBackgroundThread)...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

此时点击一下 "Step Over" 完成此行代码的执行。

如果你操作正确...

returned value 1967457 int

Name	Value	Type
------	-------	------



Activities Qt Creator

Jan 4 4:07 PM

NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```
/* Prototype for the nameHash function. This lets u
 * in main and then de... in the program.
 */
int nameHash(s...

int main() {
    string fir
    string las

    int hashValue = nameHash(first, last);

    cout << "The hash of your name is: " << hashVal
    return 0;
}

/* This is the actual function that computes the ha
 * to talk more about what hash functions do later
 * the meantime, think of it as a function that scr
 * of the input and produces a number.
 */
 * For those of you who are more mathematically inc
 * treats each character in the input name as a num
 * It then uses them as coefficients in a polynomial
 * F_p, where p is a large prime number, and evalu
```

... 你将会看到正确的值(注意红色字体)。
此时程序也跳到了下一行。

Name	Value	Type
first	"Ada"	std::string
hashValue	1967457	int
last	"Lovelace"	std::string

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	33	0x555555b65b3
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



Activities Qt Creator Jan 4 4:07 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- NameHash [main]
 - NameHash.pro
 - Sources
 - NameHash.cpp

21
22 `/* Prototype for`
23 `* in main and`
24 `*/`
25 `int nameHash(s`
26
27 `int main() {`
28 `string fir`
29 `string last = ge`
30
31 `int hashValue = nameHash(first, last);`
32
33 `cout << "The hash of your name is: " << hashVal`
34 `return 0;`
35 `}`
36
37 `/* This is the actual function that computes the ha`
38 `* to talk more about what hash functions do later`
39 `* the meantime, think of it as a function that scr`
40 `* of the input and produces a number.`
41 `*`
42 `* For those of you who are more mathematically inc`
43 `* treats each character in the input name as a num`
44 `* It then uses them as coefficients in a polynomial`
45 `* F_p, where p is a large prime number, and evalu`

Debugger GDB for "NameHash" Threads: #12 NameHash Stopped: "end-stepping-range".

Level	Function	File	Line	Address
1	studentMain	NameHash.cpp	33	0x555555b65b3
2	std::Function_handler<int (), QtGui::startBackgroundEve...			0x55555556161bc
3	GThreadStd::run()			0x5555555f9476
4	??			0x7ffff6143d84
5	start_thread	pthread_create.c	463	0x7ffff6257590
6	clone	clone.S	95	0x7ffff5e30223

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

至此，我们已经学到了很多。

接下来我们想让程序持续运行，而不是单步执行。

Activities Qt Creator Jan 4 4:07 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25 int nameHash(string first, string last);
26
27 int main() {
28     string first = getLine("What is your first name
29     string last = getLine("What is your last name?
30
31     int hashValue = nameHash(first, last);
32
33     cout << "The hash of your name is: " << hashVal
34     return 0;
35 }
36
37 /* This is the actual function that computes the h
38 * to talk more about what hash functions do later
39 * the meantime, think of it as a function that scr
40 * of the input and prod
```

Name	Value	Type
first	"Ada"	std::string
hashValue	1967457	int
last	"Lovelace"	std::string

Stopped: "end-stepping-range".

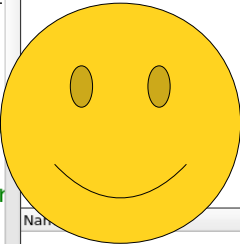
Views

Func	File	Line	Address	Condition	Ignore	Threads
??	pthread_create.c	463	0x7ffff6257590			
clone	clone.S	95	0x7ffff5e30223			

Type to locate (Ctrl...

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results

看一下这个按钮，鼠标放上去会显示“Continue”
这个按钮的意思是，“不要暂停程序，保持程序运行”



Activities NameHash Jan 4 4:08 PM NameHash.cpp @ NameHash [main] - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects NameHash [main] NameHash.pro Sources NameHash.cpp

```
21
22 /* Prototype for the nameHash function. This lets u
23 * in main and then define it later in the program.
24 */
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
```

NameHash Console [Completed]

File Edit Options Help

What is your first name? Ada
What is your last name? Lovelace
The hash of your name is: 1967457

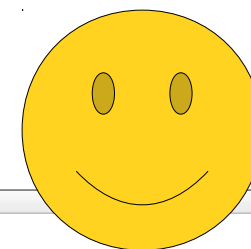
Type

Click一下，你会得到这样的结果。
(如果这个窗口没有自动弹出。请手动打开。)
此时，程序也完成了执行。

Level File Address Condition Ignore Threads

1 student.c 0x5555555555555555
2 std::Function_handler<int (>, QtGui::startBackgroundDe... 0x5555555555555555
3 GThreadStd::run() 0x7ffff6143d84
4 ?? 0x7ffff6143d84
5 start_thread pthread_create.c 463 0x7ffff6257590
6 clone clone.S 95 0x7ffff5e30223

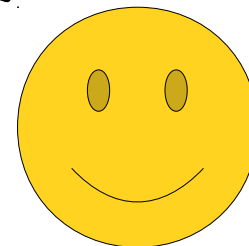
Type to locate (Ctrl... 1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 QML Debugger Console 7 Version Control 8 Test Results



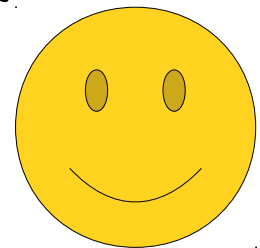
相信至此，你对调试器的使用已经熟悉了。



你知道如何设置断点，让程序停在特殊的位置。



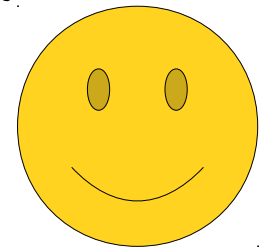
你知道如何查看调用栈和局部变量。



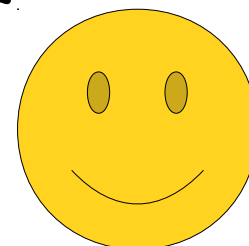
你知道如何单步执行，并观察值的变化。



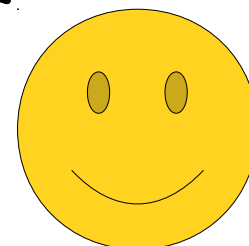
你知道如何执行完一个函数，并知道如何让程序保持运行。



随着课程的深入，你会对调试器越来越熟悉，对程序的运行方式也会越来越了解。



另外，如果你继续学习软件开发，
你会发现，学会使用调试器是个非常有价值的技能。



希望以上信息，可以给你带来帮助！
欢迎参加本期的课程！

