Subject: Conditional jump or move depends on uninitialised value(s)
Posted by Novo on Mon, 26 Jan 2009 01:35:05 GMT

Linux x86_64 Ubuntu 7.10
Application: HomeBudget

Conditional jump or move depends on uninitialised value(s)

    /home/ssg/dvlp/cpp/upp/svn/uppsrc/Draw/Draw.cpp:100 (if(tonative) {)
    /home/ssg/dvlp/cpp/upp/svn/uppsrc/Draw/Draw.cpp:127 (if(tonative))

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by mirek on Mon, 26 Jan 2009 09:04:44 GMT

Novo wrote on Sun, 25 January 2009 20:35Linux x86_64 Ubuntu 7.10
Application: HomeBudget

Conditional jump or move depends on uninitialised value(s)

    /home/ssg/dvlp/cpp/upp/svn/uppsrc/Draw/Draw.cpp:100 (if(tonative) {)
    /home/ssg/dvlp/cpp/upp/svn/uppsrc/Draw/Draw.cpp:127 (if(tonative))

These are now definitely fixed.

Mirek

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by Novo on Mon, 26 Jan 2009 15:10:19 GMT

Thanks!

I have a lot of problems with x64 platforms. TheIDE is not working with Windows XP x64, my application crashes on x64 Linux.

Hopefully, it is fixed now.

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by mirek on Mon, 26 Jan 2009 16:29:13 GMT

View Forum Message <> Reply to Message

Interesting.

theide not working in win64, that is a new one... Going to check it soon - have not for a while.

OTOH, I am using Ubuntu64 for all Linux development, booting Ubuntu32 really rarely (about as often as Vista64) - have not noticed any problems at all.

What packages is your app using?

Mirek

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by Novo on Tue, 27 Jan 2009 05:06:37 GMT

View Forum Message <> Reply to Message

Thanks. All reported uninitialised values are gone.

luzr wrote on Mon, 26 January 2009 11:29Interesting.

theide not working in win64, that is a new one... Going to check it soon - have not for a while.

I got XP x64 in my current company and tried to build and use 64-bit version of TheIDE. I managed to build and launch it, but got no luck with putting brackpoints.

I'll try to check that again tomorrow.

Quote:
OTOH, I am using Ubuntu64 for all Linux development, booting Ubuntu32 really rarely (about as often as Vista64) - have not noticed any problems at all.

What packages is your app using?

Mirek

After fixing all issues with Draw and SQL the only problem I'm seeing is with GridCtrl. But this is another story.

Another useful fix. This is an uninitialized value too. Using of initialization lists is a good practice ...

Index: uppsrc/Core/Gtypes.h
===
--- uppsrc/Core/Gtypes.h   (revision 812)
+++ uppsrc/Core/Gtypes.h   (working copy)
@@ -60,7 +60,7 @@

        String   ToString() const;

-       Size_() {}
+   Size_() : cx(0), cy(0) {}
        Size_(T cx, T cy) : cx(cx), cy(cy) {}

        Size_(const Size_<int>& sz) : cx((T)sz.cx), cy((T)sz.cy) {}

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by mirek on Tue, 27 Jan 2009 09:25:07 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 27 January 2009 00:06Thanks. All reported uninitialised values are gone.

luzr wrote on Mon, 26 January 2009 11:29Interesting.

theide not working in win64, that is a new one... Going to check it soon - have not for a while.

I got XP x64 in my current company and tried to build and use 64-bit version of TheIDE. I managed to build and launch it, but got no luck with putting brackpoints.

Ah, well. Debugger not working, that is expected, at least sort of. Some more work is needed there.

Index: uppsrc/Core/Gtypes.h
===
--- uppsrc/Core/Gtypes.h   (revision 812)
+++ uppsrc/Core/Gtypes.h   (working copy)
@@ -60,7 +60,7 @@

        String   ToString() const;

-       Size_() {}
+   Size_() : cx(0), cy(0) {}

```
    Size_(T cx, T cy) : cx(cx), cy(cy) {}

    Size_(const Size_<int>& sz)  : cx((T)sz.cx), cy((T)sz.cy) {}
```

Well, that is how it is meant to be. If you do not initialize Size, it is not initialized...

The point is

Buffer<Size> x(1000);

I might not want to have implicit memset(&x, 0, 4000) - what if I know I am going to change these sizes immediately to something else?

Mirek

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by Novo on Wed, 28 Jan 2009 04:43:35 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 27 January 2009 04:25
Well, that is how it is meant to be. If you do not initialize Size, it is not initialized...

The point is

Buffer<Size> x(1000);

I might not want to have implicit memset(&x, 0, 4000) - what if I know I am going to change these sizes immediately to something else?

Mirek

Objections:

1) it looks like you added documentation about that only today. And the word "uninitialized" is not in capital letters, so, it is very easy to overlook that.

2) you are saving two assembler commands per object and causing a lot of problems for other developers. Does it really worth that? It looks like one needs to learn all these tricky side-effects before he can start developing.

There always be problems with Size because of that. Guess how I found this constructor. At some point you will forget all these tricks yourself. Hopefully it happens after you turn 80.

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by mrjt on Wed, 28 Jan 2009 08:46:47 GMT
View Forum Message <> Reply to Message

I personally don't think this is a major issue:
1) All Upp 'basic' types (those with public member variables, Point_, Rect_ etc) exibit the same behaviour, so you only have to learn it once.
2) C++ convention is that variables are uninitialised, so you should be aware of the issue already. Buffer<int> x(2000); would be just as uninitialised.

On the other hand, in probably 99.9% of cases initialisation would be safer and come without any practical performance consequences.

Perhaps there is a case for having initialisation on by default, but available to be switched off with a compiler flag for those who really care?

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by mirek on Wed, 28 Jan 2009 15:21:14 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 28 January 2009 03:46I personally don't think this is a major issue:
1) All Upp 'basic' types (those with public member variables, Point_, Rect_ etc) exibit the same behaviour, so you only have to learn it once.
2) C++ convention is that variables are uninitialised, so you should be aware of the issue already. Buffer<int> x(2000); would be just as uninitialised.

On the other hand, in probably 99.9% of cases initialisation would be safer and come without any practical performance consequences.

Perhaps there is a case for having initialisation on by default, but available to be switched off with a compiler flag for those who really care?

I guess that pretty much sums what I think about the issue...

Except I would not vote for the compiler flag and my wild estimate is more something like 95%....

Mirek

Subject: Re: Conditional jump or move depends on uninitialised value(s)
Posted by Novo on Thu, 29 Jan 2009 01:56:45 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 28 January 2009 10:21
I guess that pretty much sums what I think about the issue...

Except I would not vote for the compiler flag and my wild estimate is more something like 95%....

Mirek

OK. Thanks for the explanation.