Subject: Conditional breakpoints
Posted by dolik.rce on Thu, 09 Dec 2010 19:42:40 GMT
View Forum Message <> Reply to Message

Hi everyone,

Today, I finally got angry enough to start implementing one of the features I seriously miss in theide: conditional breakpoints. I guess most of you knows the frustration when you know that the bug is probably in the last iteration of 1000 cycles loop and you just can't tell the debugger to stop when i=999

So I started digging into the code and to my great surprise, I found out that the code is already there, just commented out! Mirek, why are you hiding such a powerful feature from us?!

Then I tried to uncomment it (ide/idebar.cpp:480) to see if it really works... well it didn't but the fix is actually trivial, just change Gdb::SetBreakpoint() in Gdb.cpp to:bool Gdb::SetBreakpoint(const String& filename, int line, const String& bp)
{
 String bi = Bpoint(*host, filename, line);
 if(bp.IsEmpty())
  FastCmd("clear " + bi);
 else if(bp[0]==0xe)
  FastCmd("b " + bi);
 else
  FastCmd("b " + bi + " if " + bp);
 return true;
}

Now the conditional breakpoints work perfectly (with gdb). I am not sure what will happen with MSVC installs (I don't have any to test - help wanted), but I believe it will just ignore the condition, so it should not cause any trouble. If anyone decides to fix this for windows users too I can just point you to this page where you can find some possibly usefull info about msdev syntax.

Is it OK to commit this in current state? Or should I wait till someone writes the M$ part?

Best regards,
Honza

Subject: Re: Conditional breakpoints
Posted by alendar on Fri, 24 Dec 2010 23:06:36 GMT
View Forum Message <> Reply to Message

This looks like a cool feature. I'd love to try it if it gets deployed. I use both the Windows and linux versions.

Subject: Re: Conditional breakpoints
Posted by mirek on Sat, 08 Jan 2011 12:50:05 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Thu, 09 December 2010 14:42Hi everyone,

Today, I finally got angry enough to start implementing one of the features I seriously miss in theide: conditional breakpoints. I guess most of you knows the frustration when you know that the bug is probably in the last iteration of 1000 cycles loop and you just can't tell the debugger to stop when i=999

So I started digging into the code and to my great surprise, I found out that the code is already there, just commented out! Mirek, why are you hiding such a powerful feature from us?!

Then I tried to uncomment it (ide/idebar.cpp:480) to see if it really works... well it didn't but the fix is actually trivial, just change Gdb::SetBreakpoint() in Gdb.cpp to:bool Gdb::SetBreakpoint(const String& filename, int line, const String& bp)
{
 String bi = Bpoint(*host, filename, line);
 if(bp.IsEmpty())
  FastCmd("clear " + bi);
 else if(bp[0]==0xe)
  FastCmd("b " + bi);
 else
  FastCmd("b " + bi + " if " + bp);
 return true;
}

Now the conditional breakpoints work perfectly (with gdb). I am not sure what will happen with MSVC installs (I don't have any to test - help wanted), but I believe it will just ignore the condition, so it should not cause any trouble. If anyone decides to fix this for windows users too I can just point you to this page where you can find some possibly usefull info about msdev syntax.

Is it OK to commit this in current state? Or should I wait till someone writes the M$ part?

Best regards,
Honza

Please commit.

I do not remember the exact issue why it is commented out...

Mirek

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Sat, 08 Jan 2011 16:49:21 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 08 January 2011 13:50
Please commit.

I do not remember the exact issue why it is commented out...

Mirek
OK, committed. I tested with MSVC as well, to make sure there is no problem with it (yep, finally installed it in wine) Unfortunately the debugging under wine doesn't work properly... But from what I saw it appears that the breakpoints set as conditional are treated just like normal ones. I will try to investigate bit further, maybe I'll find out how to make it work.

Honza

Subject: Re: Conditional breakpoints
Posted by mirek on Sat, 08 Jan 2011 18:54:00 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 08 January 2011 11:49mirek wrote on Sat, 08 January 2011 13:50
Please commit.

I do not remember the exact issue why it is commented out...

Mirek
OK, committed. I tested with MSVC as well, to make sure there is no problem with it (yep, finally installed it in wine) Unfortunately the debugging under wine doesn't work properly... But from what I saw it appears that the breakpoints set as conditional are treated just like normal ones. I will try to investigate bit further, maybe I'll find out how to make it work.

Honza

Ah, that is the issue We cannot do this in MSC, so I have commented it out, because MSC is IMO more important...

OK, please, would it be possible to make it debugger dependent?

(Or, of course, if you would like to add conditionals to MSC, that would be a nice solution as well

---

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Sat, 08 Jan 2011 19:52:50 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 08 January 2011 19:54Or, of course, if you would like to add conditionals to MSC, that would be a nice solution as well
I would like to do that, but first I have to make MSC debugging work in wine... (Because I'm not

going to install windows just because of this)

Making the feature debugger dependent is not really simple. Mainly because behavior when user switches build methods would be hard to implement in any logical way (should the breakpoints disappear, lose the condition or what?).

BTW: While looking around I found few weird places in the debugging code, e.g. if you compile theide with mingw, it can't use Pdb debugger. I will post or commit fixes for this as well...

Honza

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Sun, 09 Jan 2011 12:52:00 GMT
View Forum Message <> Reply to Message

I have had a closer look at the Pdb class and related parts of theide and I have to confess that I am seriously confused. At first look, it appears that theide implements it's own debugger. Looking e.g. in Pdb::AddBp(), I see that to add breakpoint theide directly changes the memory of the debugged process. Also the huge size of Pdb class code (compared to Gdb) hints that it is way more complicated.

Mirek, could you give me a quick overview about how this beast works? Just a few sentences about the design and hint where to look at start would be fine. Also, if my idea about how this works is correct, could you tell me why doesn't theide use e.g. cdb.exe, that comes with the SDK, in similar manner as gdb is used? It seems to be much simpler at first glance, so there must be some serious reason...

Honza

Subject: Re: Conditional breakpoints
Posted by mirek on Sun, 09 Jan 2011 14:00:09 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 09 January 2011 07:52I have had a closer look at the Pdb class and related parts of theide and I have to confess that I am seriously confused. At first look, it appears that theide implements it's own debugger.

Yes.

Quote:
Mirek, could you give me a quick overview about how this beast works? Just a few sentences about the design and hint where to look at start would be fine.

Well, there is a M$ supplied dbghelp.dll file that provides functions to extract symbolic debug info from .exe.

Then there is Win32 debugging API.

The rest is me

Quote:
Also, if my idea about how this works is correct, could you tell me why doesn't theide use e.g. cdb.exe, that comes with the SDK, in similar manner as gdb is used? It seems to be much simpler at first glance, so there must be some serious reason...

Because cdb.exe is even more pain in the ass to work with than gdb.

This way, if nothing else, pdb debugger is fast and relatively reliable.

In fact, if I would have similar api to dbghelp.dll (or, more precisely, if I understood existing equivalents to it for posix), I would do posix debugging similar, avoiding gdb.

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Mon, 10 Jan 2011 15:41:27 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 09 January 2011 15:00At first look, it appears that theide implements it's own debugger.
Yes.[/quote]
Well, exactly what I was afraid of Anyway, I read enough of the sources to figure out how to implement the conditional breakpoints. If all goes well there will be a very little code to add, since most of the needed functionality is already there. So the last problem that remains is that I have nowhere to test it (The debugger refuses to work in wine) Looks like I will have to either install windows in emulator or install all the stuff on someone else's computer...

mirek wrote on Sun, 09 January 2011 15:00
Because cdb.exe is even more pain in the ass to work with than gdb.

This way, if nothing else, pdb debugger is fast and relatively reliable.

In fact, if I would have similar api to dbghelp.dll (or, more precisely, if I understood existing equivalents to it for posix), I would do posix debugging similar, avoiding gdb.
I see a major advantage of gdb in fact that it is already written From my point of view, gdb is easy to use, has some really nice features and is installed on almost any computer. But I can't talk for cdb, I never used that. If you say it's pain to work with, I'll have to believe you

Honza

Subject: Re: Conditional breakpoints
Posted by mirek on Mon, 10 Jan 2011 16:58:02 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Mon, 10 January 2011 10:41From my point of view, gdb is easy to use, has some really nice features and is installed on almost any computer. But I can't talk for cdb, I never used that. If you say it's pain to work with, I'll have to believe you

I have not said gdb is bad. It is just difficult to combine it with GUI frontend intended for C++ debugging.

E.g. one of issues is that if you query the value of any widget 'variable', you get about 1.5MB of text to parse...

Plus, all that pushing commands to gdb and parsing results, done via pipes, is extremely painful and quirky...

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Thu, 13 Jan 2011 14:03:14 GMT
View Forum Message <> Reply to Message

I finally got some time with windows computer, so I could test what I blind coded in past few days

I extended the parser in Exp.cpp to allow comparisons, logical and bitwise operations. Hopefully with correct C++ priorities Then I added a Vector to store the conditions and Pdb::ConditionCheck() function that check whether the condition for given breakpoint is fulfilled or not by evaluating it using Pdb::Exp() call. Up to here it appears to work quite well.

I hit a problem when I tried to "cancel" the breakpoints for which condition is not fulfilled in Pdb::RunToException(). The code looks like it should be enough to break instead of returning and wait for next breakpoint:

```
   switch(event.dwDebugEventCode) {
   case EXCEPTION_DEBUG_EVENT: {
// ...
    int bp=bp_set.Find((adr_t)event.u.Exception.ExceptionRecord.ExceptionAddress);
    if(event.u.Exception.ExceptionRecord.ExceptionCode == EXCEPTION_BREAKPOINT && bp
>= 0)
     #ifdef CPU_32
     context.Eip = (adr_t)event.u.Exception.ExceptionRecord.ExceptionAddress;
     #else
     context.Rip = (adr_t)event.u.Exception.ExceptionRecord.ExceptionAddress;
     #endif

    RemoveBp();
    LLOG("Exception: " << FormatIntHex(event.u.Exception.ExceptionRecord.ExceptionCode) <<
```

```
        " at: " << FormatIntHex(event.u.Exception.ExceptionRecord.ExceptionAddress) <<
        " first: " << event.u.Exception.dwFirstChance);
    if(bp>=0 && !ConditionCheck(bp))
     break;     // condition doesn't fit - we don't want to stop at this breakpoint
    return true;       //the condition is true, we should stop
// ...
```

This however still stops at every breakpoint, regardless if it breaks or returns. Could you give me a hint how to do this properly, please?

Later, it would be also good idea to actually do the check before switching theide back to foreground etc., but that will probably require bigger changes. For now I'm just wondering how to skip the unfitting breakpoints.

Honza

PS: I attach all the changed sources, so you can test.

EDIT: Removed attachment, newer version available below.

Subject: Re: Conditional breakpoints
Posted by mirek on Thu, 13 Jan 2011 14:32:57 GMT
View Forum Message <> Reply to Message

Have you tried to debug it?

(Indeed, little is as amusing as debugging debugger, preferably in itself

---

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Fri, 14 Jan 2011 06:54:55 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 13 January 2011 15:32Have you tried to debug it?

(Indeed, little is as amusing as debugging debugger, preferably in itself

I tried - it's hell of a confusing job to debug ide that is debugging another program But I had limited time and couldn't get the hang of it quick enough :-/

Honza

Subject: Re: Conditional breakpoints
Posted by dolik.rce on Fri, 14 Jan 2011 12:28:13 GMT

So, I looked into bit more and get better understanding about how it works, but apparently still not good enough. I was able to correct few errors, so now it almost works. The only problem now is that the debugged program is terminated as soon as first unfulfiled condition is hit.

Unfortunately, I will be pretty busy next few days, so I won't be able to get back to this until after next week If anyone wants to finish it sooner, I won't object The sources are attached.

Honza

File Attachments

1) Debuggers.zip, downloaded 323 times