Subject: Out of memory panic
Posted by koldo on Thu, 17 May 2012 10:38:48 GMT
View Forum Message <> Reply to Message

Hello all

Although U++ doc. clearly says that: "...We decided to ignore possibility of "out-of-memory" exceptions and recovery...", I would ask to do something .

A possible proposal could be that OutOfMemoryPanic() function could be an user defined function. The user could decide to, for example:
- Show an out of memory message
- Free some memory kept to give breath to the program in this situation
- Save critical data if something goes wrong
- Throw an exception for the main program to handle this gracefully

Now the program just aborts without any possibility of save data processed.

Subject: Re: Out of memory panic
Posted by BioBytes on Fri, 18 May 2012 09:20:25 GMT
View Forum Message <> Reply to Message

Hello Koldo,

Your proposal is interesting. For example, I found that PanicMessageBox freezes application. This problem appears when an error occurs in code dealing with MySQl connection.

Regards

Biobytes

Subject: Re: Out of memory panic
Posted by mdelfede on Sat, 19 May 2012 12:05:17 GMT
View Forum Message <> Reply to Message

IMHO in modern OS with virtual memory, a true out-of-memory error is almost impossible.
What you usually get is a system slowdown because of swapping.

OTOH, memory corruptions often are reported as out of memory, and in my experience, there's nothing you can do about it... most recovery routines need working allocators.

The only solution I can think about is to have a separate block of preallocated memory usable by emergency routine as a pool for newly allocated blocks while trying to close gracefully the application, maybe saving some work.
Or, even if not preallocated, the memory could be requested as a big block from OS and then the

allocators could switch to that block while shutting down.

Max

Subject: Re: Out of memory panic
Posted by Tom1 on Sat, 19 May 2012 18:36:00 GMT
View Forum Message <> Reply to Message

Max, I'm afraid my case is the 'almost impossible' kind: While typical applications are quite happy with the current situation, I have an application that processes large (and I mean really huge) data sets and users tend to load in more data into the process until the address space is exhausted. When we hit the 3 GB allocation limit in the 32-bit windows application address space, there is nothing more to do. In that application, it would certainly be nice to use an allocation function that first tests if the request can be successfully completed. Even having a safety margin before hitting the address space limit would be nice.

Best regards,

Tom

Subject: Re: Out of memory panic
Posted by mdelfede on Sat, 19 May 2012 19:24:14 GMT
View Forum Message <> Reply to Message

Well... maybe there's a way to cope with exhaust virtual address space, which for 32 bit is quite possible... for 64 bit is obviously impossible to reach it.
But anyways I guess that long before of exhausting the virtual address space you'll have so hard swapping slowing down your system to an unusable state.
IMHO your system will become completely irresponsive long before you reach the limit.... what would be the benefit of triggering an emergency routine then ?

Something a bit more useful would be some way to :

1- Check the amount of physical memory / swap space BEFORE start of your application

2- Check same values during app usage and taking some decision when swapped out space will grow too much, maybe popping up an alert box.

But the main problem would be to define the "too much", which may depend on other processes on your machine, which could be sleeping processes (so it's harmless for them to swap out all ram...) or active processes, which can block your system just because you used all physical ram.

So, IMHO also this solution would be almost useless.

Last possibility would be to "fix" the maximum amount you want to be used by your application

and popup an alertbox when reached. On a 32 bit system, you can sau "I want the alert box on 1 GB limit" and you can be almost sure your app will behave good.
I guess that adding this to upp would be not too difficult and could be done in a couple of ways :

1) manual mode... have a function returning TOTAL amount of memory requested by your app, and you'll do a check by yourself before allocating. Easy and maybe (I didn't check...) already embedded in Upp.

2) semi-automatic mode : introduce a function like "SetMemoryLimit()" and let upp throw an exception when allocating past this one. You'll have to surround all your allocation with try... except, but I guess it'll behave bad with constructors.

3) Automatic mode... like step 2, but Upp could pop an alert box and call an user function when accepted, disabling temporary the memory limit. I guess this one would be the best solution. Don't know how much overhead to upp allocators would bring....

Max

Subject: Re: Out of memory panic
Posted by Tom1 on Sat, 19 May 2012 20:28:51 GMT
View Forum Message <> Reply to Message

Actually swapping is not the problem in many cases. We have nowadays machines with 6-8 GB of RAM running 64-bit Windows editions. So, in many cases, the address space for a 32-bit application is really exhausted before physical RAM runs out.

I admit, that the machines with just 2-3 GB of RAM would experience severe swapping before getting to the limit.

I think your solution B1 (the manual one with calling a function to get the current allocation footprint of the process) would best fit my purposes. This way I could skip trying to allocate buffers for the large data sets after meeting the limit (-- which BTW for me is about 1GB less than available physical RAM.) Is there such a function in Core?

--

What I would really love to have, is your Protect -package working with MSC 64-bit target compilation, which would take the address space pain out for good. Currently my application is limited to 32-bit exes...

Best regards,

Tom

Subject: Re: Out of memory panic
Posted by mdelfede on Sat, 19 May 2012 20:41:18 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Sat, 19 May 2012 22:28Actually swapping is not the problem in many cases. We have nowadays machines with 6-8 GB of RAM running 64-bit Windows editions. So, in many cases, the address space for a 32-bit application is really exhausted before physical RAM runs out.

I admit, that the machines with just 2-3 GB of RAM would experience severe swapping before getting to the limit.

I think your solution B1 (the manual one with calling a function to get the current allocation footprint of the process) would best fit my purposes. This way I could skip trying to allocate buffers for the large data sets after meeting the limit (-- which BTW for me is about 1GB less than available physical RAM.) Is there such a function in Core?

I don't know... you should ask Mirek. Probably there's something like that somewhere....

Quote:
What I would really love to have, is your Protect -package working with MSC 64-bit target compilation, which would take the address space pain out for good. Currently my application is limited to 32-bit exes...

eh... me too, but 64 but MSC don't support inline assembler, so I don't see an easy way to support Protect package.
I'm forced to stay with 32 bit too because of that.
I could try with external assembler functions, but I'm afraid those would be by far easier to defeat.
And, the other solution, encrypt the whole stuff using PE data to do it as professional solutions do would be too much work for me now.
I even don't know if you can mix 32 bit and 64 bit modules, but I guess not.... But if you app allows it you could separate it in a couple of executables, one (32 bit) protected and the other one (64 bit) unprotected.

Ciao

Max

Subject: Re: Out of memory panic
Posted by koldo on Sun, 20 May 2012 14:38:55 GMT
View Forum Message <> Reply to Message

Hello all

I am almost agree with all the opinions. This situation is not usual. However it is very possible In my situation it is a program that manages huge amounts of data.

Now if a lack of memory is detected, a message is shown and program is ended. This is U++ internal and out of programmer control.

I think it would be great if U++ would let the programmer to choose what to do. The default solution is OK but maybe important data is lost or safety would be at risk if program is controlling devices.

I think that U++ would have to let the programmer to gracefully end if any unexpected situation would happen. And sometimes unexpected things happen. Exceptions are a good way to catch this situations.

Subject: Re: Out of memory panic
Posted by mirek on Sun, 20 May 2012 15:11:18 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Sat, 19 May 2012 14:36Max, I'm afraid my case is the 'almost impossible' kind: While typical applications are quite happy with the current situation, I have an application that processes large (and I mean really huge) data sets and users tend to load in more data into the process until the address space is exhausted. When we hit the 3 GB allocation limit in the 32-bit windows application address space, there is nothing more to do. In that application, it would certainly be nice to use an allocation function that first tests if the request can be successfully completed.

The problem is not with MemoryAlloc returning NULL or invoking exception instead of panic. The real problem is that to do handle such exception correctly in all cases is something that requires almost inhuman intelligence...

Subject: Re: Out of memory panic
Posted by mirek on Sun, 20 May 2012 15:13:30 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 20 May 2012 10:38Hello all

I am almost agree with all the opinions. This situation is not usual. However it is very possible In my situation it is a program that manages huge amounts of data.

Now if a lack of memory is detected, a message is shown and program is ended. This is U++ internal and out of programmer control.

I think it would be great if U++ would let the programmer to choose what to do. The default

solution is OK but maybe important data is lost or safety would be at risk if program is controlling devices.

I think that U++ would have to let the programmer to gracefully end if any unexpected situation would happen. And sometimes unexpected things happen. Exceptions are a good way to catch this situations.

Well, if adding "OutOfMemoryHandler" function would statisfy you, I have no problem adding it.

However, please notice that you can do very little in such handler.... You e.g. cannot even call Exclamation to report the issue...

Subject: Re: Out of memory panic
Posted by mdelfede on Sun, 20 May 2012 15:19:01 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 20 May 2012 17:13

Well, if adding "OutOfMemoryHandler" function would statisfy you, I have no problem adding it.

However, please notice that you can do very little in such handler.... You e.g. cannot even call Exclamation to report the issue...

I guess that the best would be to add a trigget with an user-settable memory level.
I mean :

```
::SetMemoryLimit(1000000000, THISBACK(myHandler)); // 1 GB

.....
byte *x = new byte[500000000];
byte *y = new byte[500000000];
byte *z = new byte;   <<-- This one triggers the handler
```

Of course, the Upp allocator should, when limit reached :
1) - Disable the handler
2) - Return NULL, just in case....
3) - Allow normal usage, allowing the user handler to do whathever he wants.

Max

Subject: Re: Out of memory panic
Posted by koldo on Sun, 20 May 2012 18:15:24 GMT
View Forum Message <> Reply to Message

Hello all

In my case I have got the problem when trying to add too many rows to a GridCtrl. With matrices an vectors there is no problem because some return NULL and other throw a bad_alloc if size is too big.

Quote:Well, if adding "OutOfMemoryHandler" function would satisfy you, I have no problem adding it.

However, please notice that you can do very little in such handler.... You e.g. cannot even call Exclamation to report the issue...Thanks Mirek. That solution sounds nice. However I do not know why an Exclamation does not work (I know it because I tried it freeing memory before ...).

Quote:I guess that the best would be to add a trigger with an user-settable memory level.Hello Massimo. You are in fact adding a OutOfMemoryHandler too. However I prefer the programmer and the user to use as much memory as it is available.

Subject: Re: Out of memory panic
Posted by mdelfede on Sun, 20 May 2012 18:20:47 GMT
View Forum Message <> Reply to Message

koldo wrote on Sun, 20 May 2012 20:15Hello Massimo. You are in fact adding a OutOfMemoryHandler too. However I prefer the programmer and the user to use as much memory as it is available.

Well, if you trigger the memory callback when mem is completely exhausted, you won't be able to do anything when inside it... OTOH, if you estimate a "right" amount that triggers it, you'll be able to gracefully recover.
Anyways, you could do something like

::SetMemoryLimit(INFINITE, THISBACK(myHandler)); // No preset limit

you could also have the maximum available mem.

Max

Subject: Re: Out of memory panic
Posted by koldo on Sun, 20 May 2012 18:52:11 GMT

View Forum Message <> Reply to Message

Hello Massimo

Yes. I always considered to free memory before.

To add this memory control would require an additional job but, not too much.

Subject: Re: Out of memory panic
Posted by mirek on Mon, 21 May 2012 11:35:17 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sun, 20 May 2012 14:20koldo wrote on Sun, 20 May 2012 20:15Hello Massimo. You are in fact adding a OutOfMemoryHandler too. However I prefer the programmer and the user to use as much memory as it is available.

Well, if you trigger the memory callback when mem is completely exhausted, you won't be able to do anything when inside it... OTOH, if you estimate a "right" amount that triggers it, you'll be able to gracefully recover.
Anyways, you could do something like

::SetMemoryLimit(INFINITE, THISBACK(myHandler)); // No preset limit

you could also have the maximum available mem.

Max

Well, this has the problem that you do not really know how much memory is really available...

Subject: Re: Out of memory panic
Posted by mdelfede on Mon, 21 May 2012 11:39:58 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 21 May 2012 13:35
Well, this has the problem that you do not really know how much memory is really available...

Of course... but if you've the problem of running out of address space, you can "fix" a reasonable limit and alert user when it's reached.... better than nothing
A perfect solution don't exist, I'm afraid.

Max

Subject: Re: Out of memory panic
Posted by mirek on Tue, 22 May 2012 11:11:32 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Mon, 21 May 2012 07:39mirek wrote on Mon, 21 May 2012 13:35
Well, this has the problem that you do not really know how much memory is really available...

Of course... but if you've the problem of running out of address space, you can "fix" a reasonable limit and alert user when it's reached.... better than nothing
A perfect solution don't exist, I'm afraid.

Max

Well, but perhaps you do not need anything really new - you can get memory allocated by MemoryUsedKb, then add some test before big operation... I it is as good as you can go...