

CoCoALib - Feature #1755

CheckForInterrupt: give more info?

13 Jul 2023 19:52 - John Abbott

Status:	Closed	Start date:	13 Jul 2023
Priority:	High	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Improving	Estimated time:	1.90 hour
Target version:	CoCoALib-0.99850	Spent time:	1.80 hour
Description			
<p>This may be not so important...</p> <p>Today I wanted to interrupt a long loop and know roughly how far the computation had proceeded. Currently CheckForInterrupt does not do this...</p> <p>It is currently possible by calling GetAndResetSignalReceived (defined in SignalWatcher).</p> <p>We should document how a user can more info when a signal is handled.</p>			

History

#1 - 13 Jul 2023 19:54 - John Abbott

Here is how I managed it in my ad hoc program:

```
for (long n=10000001; n < 20000000; n += 2)
{
    if (!IsSqFree(n)) continue;
    //    CheckForInterrupt("main loop"); // does not say how far we reached
    if (GetAndResetSignalReceived()) { cout << endl << "INTR at n=" << n << endl; break; }
    .....
}
```

#2 - 14 Jul 2023 15:11 - John Abbott

While it is not really much shorter, I was thinking of trying to implement an extension of **CheckForInterrupt** which takes a 2nd arg which will be printed if an interrupt occurs. Hmmm, but if I do not use a macro then the 2nd arg will always be evaluated :-/

```
for (long n=10000001; n < 20000000; n += 2)
{
    if (!IsSqFree(n)) continue;
    CheckForInterrupt("main loop", n);
    .....
}
```

It would be best not to use a MachineInt for the 2nd arg, as otherwise the ctor will always be called. Maybe I could make it a template function? (but that tends to expose more impl details than I'd like...)

KISS: for first impl make 2nd arg of type long??

#3 - 14 Jul 2023 17:46 - John Abbott

- Status changed from New to Feedback
- Assignee set to John Abbott
- % Done changed from 0 to 90

After discussing with Anna (by phone), we think that extending CheckForInterrupt as suggested in comment 2 above is not really such a good idea.

Here are some reasons:

- the proposed extension is of limited flexibility; moreover the extra information is printed only when verbosity is at least 11 (which was not really what I had expected)
- the proposed solution/workaround in comment 1 is not so bad, but we must document it properly (with an example!) It also offers the user complete flexibility (while avoiding potentially evaluating expressions which are then simply discarded *e.g.* if the extra arg to CheckForInterrupt is non-trivial)
- how often does one really want the extra information? (JAA suspects only rarely)
- one can also use **ProgressReporter** (which is probably more informative); of course, it won't report exactly where the loop was when interrupted...

We are inclined to **reject** this idea; but for the moment I have put the issue into feedback in case any more thoughts come to mind.

#4 - 26 Jul 2023 13:52 - John Abbott

- Priority changed from Low to High

Create an example of using **GetAndResetSignalReceived**

#5 - 22 Dec 2023 20:56 - John Abbott

- Status changed from Feedback to Closed
- % Done changed from 90 to 100
- Estimated time set to 1.90 h

I have made a new example **ex-interrupt3.C** which illustrates the idea from comment 1.