CoCoALib - Slug #792

configure: search for libgmp too slow

02 Nov 2015 14:16 - John Abbott

Status:	Closed	**Start date:**	02 Nov 2015
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	Portability	**Estimated time:**	0.88 hour
Target version:	CoCoALib-0.99700	**Spent time:**	0.85 hour

Description

I have new computer (here in Kassel). I am trying to build CoCoALib/CoCoA-5. The configure script took **ages** to decide that there are many different libgmp available.

Make this (much) faster!

Related issues:

Related to CoCoALib - Bug #593: Temporary directories used during configuration	**Closed**	**20 Jul 2014**

History

#1 - 02 Nov 2015 14:39 - John Abbott

The situation was the following. The new computer did not have GMP installed, so the attempt to use -lgmp failed, so configuration/gmp-find.sh then does a (recursive) find in the normal directories. I am pretty certain that it was the find command which was very slow.

I am installing the newest GMP (v.6.1.0) on the new computer, and note that also the GMP configuration process is very slow -- I think most files are stored on a separate server, and suppose the communication costs a lot of time.

#2 - 02 Nov 2015 14:41 - John Abbott

Since I believe that the find command is the culprit, the solution would be to eliminate it. This could be done by making the list of "usual places" more thorough, so that it includes all directories (not just directory roots).

#3 - 13 Jun 2016 11:32 - John Abbott

- Related to Bug #593: Temporary directories used during configuration added

#4 - 27 Oct 2019 21:10 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- Target version changed from CoCoALib-1.0 to CoCoALib-0.99700

- % Done changed from 0 to 50

I think this may already be almost done. Let's finish it soon!

#5 - 09 Jan 2020 22:02 - John Abbott

- Status changed from In Progress to Closed

- % Done changed from 50 to 100

- Estimated time set to 0.88 h

On my main machine gmp-find.sh (with SSD) takes less than 1s; on the little "netbook" it took about 2.8s.
Faster might be nicer, but it is hardly important.
Closing as the current impl seems to be fast enough...