2014-06-19

Work summary

Today, I have:

- Tried to compile a "light" version of Mono which does not include the complete set of configurations (static, dynamic) × (Boehm, sgen).

  It turns out that Mono's current `configure` script is badly bitrotten, and that every combination of the GC flags:

  - `--with-sgen=no`,
  - `--disable-boehm`,
  - `--with-gc=sgen`

  and of the static/dynamic flags:

  - `--with-shared_mono=no`,
  - `--disable-static`,
  - `--disable-shared`

  that I have tried has failed to compile in some way. E.g., when configuring with `--with-sgen=no`:

  ```
  make[3]: *** No rule to make target                   \
      '../../mono/mini/libmonosgen-2.0.la', needed by \
      'libmono-profiler-cov.la'.  Schluss.
  $ ls mono/mini/*.la
  mono/mini/libmini-static.la
  ```

  This is unfortunate, because it makes iteration quite a bit slower; 4 sets of executables have to be rebuilt/relinked after each patch and/or instrumentation!

  I'm sure I will find some shortcuts when working on a specific piece, but such shortcuts are inherently dangerous (good source of human errors)—too bad.

  (I do not intend to work on fixing Mono's build system in the near future, though.)

- Submitted a Mono patch/pull request for the `strtod`/`Double.Parse` problem diagnosed yesterday:

  https://github.com/mono/mono/pull/1112

  Discussion is ongoing.

- Tried to reproduce the hanging test suite on ARM plaforms... but couldn't. I may have been too impatient yesterday (some tests are sloooooow to run on the VMs); we'll see if it happens again;

- Collected `corlib_test_net_4_5.dll` NUnit test results for Tizen 2.2/ARM. (Upside of the previous point.)

  ```
  Tests run: 9911, Failures: 68, Not run: 102
  ```

  The downside is: 68 failures. The re-upside is that a lot of them seem to be related:

  ```
   1) MonoTests.System.IO.DirectoryInfoTest.Symlink :      \
    System.TypeInitializationException : An exception was  \
    thrown by the type initializer for Mono.Unix.Native.Syscall
     ----> System.DllNotFoundException : MonoPosixHelper
   ...
  +63) MonoTests.System.IO.FileStreamTest.WriteWithExposedHandle : \
    System.DllNotFoundException : libc
  ```

  TODO: Investigate;

- Investigated the `marshal7.exe` failure. The bug seems to be in the test, not in the runtime; I have emailed Mono-devel for clarifications:

  ```
  From: Damien Diederen <dd@crosstwine.com>
  To: mono-devel-list@lists.ximian.com
  Subject: Linux x86: marshal7.cs failure
  Date: Thu, 19 Jun 2014 20:19:00 +0200
  Message-ID: <87r42k225n.fsf@mini.crosstwine.com>
  ```

  Currently pending.

- Investigated the `gc-altstack.exe` failure. This is an ugly heisenbug which cannot be reproduced in GDB; timing and/or thread related. Postponed for now.

  TODO: Investigate later;

- Investigated the `block_guard_restore_aligment_on_exit.exe` failure, which seems to be a bad interaction between thread abortion and type constructors:

  ```
  System.TypeInitializationException: An exception was    \
   thrown by the type initializer for System.Console      \
   ---> System.Threading.ThreadAbortException: Thread was \
   being aborted
  ```

Adding `Console.WriteLine("Hello!")` at the beginning of the test prevents the failure. Postponed for now.

TODO: Investigate later;

- Investigated the `bug-10127.exe` failure: the test runs to (successful) completion when standalone, but is killed by the 120s timeout of the test suite runner.

  TODO: Check on hardware device;

- Investigated the `bug-18026.exe` failure, which also hits a timeout (cf. `bug-10127.exe`). A standalone run takes 6min11s on an idle VM.

  TODO: Check on hardware device;

In parallel, I have:

- Pushed the VM infrastructure (scripts + Git-Annex metadata) to a new Kitsilano repository:

  https://github.com/kitsilanosoftware/MonoTizen.VMs

  TODO: Make the images separately downloadable and add their URLs to Git-Annex;

- Cleaned up the Mono build infrastructure and pushed it to a new Kitsilano repository:

  https://github.com/kitsilanosoftware/MonoTizen.BuildScripts

- Confirmed that `strtod`-patched MCS running on the “device” is solid enough to build all of the test cases!

- Discovered and locally removed this Tizen 2.2 gem, which leads to spurious failures and all kinds of strange behaviours:

  ```
  # cat /etc/profile.d/Xorg.sh
  export HOME=/root
  export DISPLAY=:0
  ```

  TODO: Check if present in Tizen 2.2.1, and if so, the impact.

Tomorrow, I plan to:

- Investigate the `finally_block_ending_in_dead_bb.exe` failure;
- Investigate the `DllNotFoundException` issue on the ARM platform;
- Investigate the `SocketException` issue on the Intel platform;

- Add GPL licensing information to the infrastructure repos, and make them public;
- Make the VM images separately downloadable.

Later, I plan to:

- Go back to investigating the `gc-altstack.exe` failure;
- Investigate the `block_guard_restore_aligment_on_exit.exe` without the workaround in place;
- Continue with other test suites once `corlib_test_net_4_5.dll` is clean everywhere;
- Continue packaging my testing environments and scripts;
- Check if Tizen 2.2.1 has the `HOME=/root` bug;
- Follow up on the `marshal7.exe` discussion;
- Follow up on the `strtod` discussion.