libmill

Go-style concurrency in C

Home **Download** **Tutorial** **Documentation** **Development**

Development

Reporting bugs

Report a bug here:

https://github.com/sustrik/libmill/issues

Accessing source code

To clone the repository:

```
$ git clone https://github.com/sustrik/libmill.git
```

Building from source

```
$ ./autogen.sh
$ ./configure
$ make
$ make check
$ sudo make install
```

Debugging

For easy debugging use the following configure options:

```
$ ./configure --disable-shared --enable-debug --enable-valgrind
```

The above will turn the optimisation out, generate debug symbols and link all the tests with the static version of the library. The second option means that the executables in `tests` subdirectory will be actual debuggable binaries rather that wrapper shell scripts. The last option instructs valgrind about where are the coroutine stacks located and thus prevents spurios valgrind warnings.

Continuous integration

build passing

Travis: https://travis-ci.org/sustrik/libmill

Contributing

To contribute to Mill send your patch to the mailing list or, alternatively, send a GitHub pull request. In either case you have to state that your patch is submitted under MIT/X11 license, so that it can be incorporated into the mainline codebase without licesing issues.

If you make a substantial contribution to a file, add your copyright to the file header. Irrespective of whether you do so, your name will be added to the AUTHORS file to indicate you own copyright to a part of the codebase.

Release process

These instructions are intended for the project maintainers:

1. Run `make distcheck` to check whether the packaging process still works.
2. Bump ABI version as appropriate (see here: http://250bpm.com/blog:41). Commit it and push it back to the master on GitHub.
3. Tag the new version and push the tag to GitHub (e.g. `git tag -a 0.3-beta; git push origin 0.3-beta`).
4. Clone a clean repo from GitHub.
5. Build the package (`./autogen.sh; ./configure; make distcheck`).
6. Get the checksum (`sha1sum`) of the package.
7. Add the package to `gh-pages` branch.
8. Adjust the download.html web page in `gh-pages` branch.
9. Commit and push to `gh-pages`.
10. Send the announcement about the release to the project mailing list.