B Technical remarks on the ForkLight compiler package

The technical requirements to run ForkLight on your local system are an ANSI-C compiler like `gcc` and an installation of the P4 library.

B.1 Installation

The ForkLight compiler package can be obtained from the WWW page

`http://www.informatik.uni-trier.de/~kessler/forklight.html`

After unpacking the ForkLight compiler distribution in a directory, the environment variable `FLDIR` has to be set to this directory. Also, `$(FLDIR)/bin` must be added to the search `PATH`.

Furthermore the `Makefiles` in `$(FLDIR)/lib` and `$(FLDIR)/ex` have to be edited in order to set the paths to the P4 installation and to the target compiler appropriately.

Installation is then done just by `make all` in `$(FLDIR)`.

By default, ForkLight uses `gcc -E` as preprocessor. If your system does not support `gcc`, you must change the string `thepreprocessor` in `$(FLDIR)/src/flcc.h` appropriately.

B.2 Compiler options

The ForkLight compiler `flcc` is built in the `$(FLDIR)/bin` subdirectory. To compile ForkLight sources with `flcc` the following flags should be set:

- `-ForkLight` to tell the compiler that the source to be compiled is a ForkLight program rather than ordinary C code.
- `-nprocs`*num* to produce an executable that expects *num* processors, where *num* is a nonnegative integer. If `num` is zero, the run–time system will infer the total number of available processors at program startup.

  If the `-nprocs` option is omitted, the compiler will choose a default value (4).
- `-I$(FLDIR)/include` to find the header files of ForkLight.

As an example, consider the `Makefile` in `$(FLDIR)/ex`.

In addition, some other flags known from standard C compilers, as `-c` (no linking), `-v` (print version), or `-S` (keep assembler file, i.e. here the generated C code) can be used.

The flags for the target compiler and target linker are architecture dependent. They can be looked up in `$(FLDIR)/ex/Makefile` resp. `$(FLDIR)/lib/Makefile`. The target compiler must specify `-DP4` to activate the P4 backend, `-I$(P4_INCLUDE_DIR)` to include the P4 header files, and, if compiling for the SBPRAM, additionally `-DSBPRAM` to activate the SBPRAM–specific optimizations (non-sequentializing atomic operations, fast spawning of parallel processors). The target linker must link the P4 library; for the SBPRAM, additionally the `sbp` library must be linked.

Currently, debugging is only possible on the C level (target compiler flag `-g`) but not on the ForkLight level.