Building Schutz.

Rodney Bates

1. Prerequisites

Schutz is mostly written in Modula-3, plus some common linux tools and some specialized languages that it contains implementations of. It has so far only been developed to build and run on linux systems and only compiled for AMD64_LINUX and LINUXLIBC Modula-3 target systems.

Recently, it has been compiled only using the packedVars branch of the Modula-3 CM3 compiler, which can be found at github.com/modula3/cm3. It will probably be compilable by some earlier Modula-3 compilers, but the packedVars branch contains improvements that, at least, will make it occupy less RAM.

You will need:

- The Bash or similar shell, with commands mkdir, cd, rm, cp, source, ls, cat, and ln.
- Addtional tools sed, awk, sort, diff, grep, date, and /lib/cpp.
- A Modula-3 compiler named cm3, including shipped libraries m3core and libm3.
- Compiled and shipped libraries from the comm and ui groups of the Modula-3 distribution. Not all of these are needed, but the exact list is unascertained. Compiling ui will require installation of the X11 Athena widget development library. On Ubuntu and derived Linux distributions, package libxaw7 and libxaw7-dev should suffice.
- A copy of the Schutz distribution, which can be found at github.com/RodneyBates/schutz/master.

2. Build steps

Directory paths in the following are relative to the root of the of the git repository, i.e., parallel to its `.git` directory. The following commands are all issued in the `./scripts` subdirectory.

If the repository is not entirely fresh, issue `veryclean.sh`.

Edit ./sethostdep.sh to have the Modula-3 target you are building on. There is, so far, no automatic detection or setting of this.

Issue `fullboot.sh`. This will go through an elaborate bootstrap process, involving other scripts, creation of directories, compilations, execution of compiled programs, generation of souce code and

internal data, checking for convergence, etc. Do not panic over messages about states not being LALR. At the end, this will have created directory ./boot and many files within. It will have built and shipped library libschutzcommon, built executables LdlBoot and LdlBatch in ./boot, and executable Lbe in ./edit/AMD64_LINUX (substitute your target system).

LdlBoot is used only in the bootstrap process. You will have no further need of it unless you do development work on the process. LdlBatch adds languages to Schutz. You will not need it either, unless you do development on a supported language or add a new language. Lbe is the built editor. At this stage, each of these executables supports only ldl0, a primitive language definition language used only during bootstrapping.

Issue `AddLdl1.sh`. This will alter LdlBatch and Lbe, to additionally support ldl1, the more general language definition language used in Schutz.

Issue `AddM3.sh`. This will alter Lbe, to additionally support Modula-3.

You are done.