Running and Compiling the Compiler

Running

The compiler accepts arguments in the following syntax:

```
Compiler [flags] file
```

where *file* is the source file. If *file* is not provided, the compiler will use standard input as the input source.

The possible flags are:

```
-o=file
```

where *file* is the output file to generate ARM code to. If an output file is not specified, it will default to standard output.

```
-s
```

will force the compiler to generate code to save all registers to their respective memory locations at the end of each scope.

-p

will force the compiler to generate pedantic warnings during compilation.

Compiling

During testing, compilation was done under the following environment:

- Ubuntu 11.10
- GNU Make 3.81
- GCC 4.6.1
- Flex 2.5.35
- Bison 2.4.1

At least version 4.6.0 of GCC is required (for the use of nullptr).

Compilation can be done by invoking the "make" command when the working directory is set to the program directory. "make clean" will clean up the generated files. Finally, "make again" will force a full recompilation.

NOTE: The directory structure should be maintained when running. This is because the compiler expects the file in the "Asm" directly to be named exactly as they are. These files are the skeletal files used to generate the final code.

Emulating

The Keil project file under the "Output" directory should give proper settings to compile the assembly file. The assembly will not emulate well in Keil due to the use of unsupported (by Keil) SWI calls.