

Simplescalar installation instructions

Mats Brorsson, April 15, 2008

Introduction

The following text describes the procedure of installing the simulator Simplescalar (1) on a modern Linux distribution. The procedure has been tested on Ubuntu Linux 7.1. The instructions are modified from (2).

Preparations

Download the necessary source code files:

- `simpletools-2v0.tgz` from <http://www.simplescalar.com/tools.html>
- `simplesim-3v0d-with-cheetah.tar.gz` from <http://www.ict.kth.se/courses/IS2202/software>
- `simpleutils-990811.tar.gz` from <http://www.eecs.umich.edu/mirv/>
- `gcc-2.7.2.3.ss.tar.gz` from <http://american.cs.ucdavis.edu/RAD/gcc-2.7.2.3.ss.tar.gz>

Setup some environment variables (assuming your home directory is `/home/matsbror`):

```
$ export IDIR=/home/matsbror/simplescalar
$ export HOST=i686-pc-linux
$ export TARGET=sslittle-na-sstrix
```

Create the directory “**simplescalar**” under your home directory, and copy all the four tarballs into it:

```
$ mkdir $IDIR
$ mv simplesim-3v0d-with-cheetah.tgz $IDIR
$ mv simpletools-2v0.tgz $IDIR
$ mv simpleutils-990811.tar.gz $IDIR
$ mv gcc-2.7.2.3.ss.tar.gz $IDIR
```

Update Ubuntu (or whatever distribution you are using) with the following packages:

- `build-essential`
- `flex`
- `bison`
- `yacc` (maybe not needed)

You can use the command “`sudo apt-get install <package name>`” to retrieve and install these packages.

Want to run on a Windows computer?

If you don't have access to a Linux computer and you want to use a Windows computer instead, you can use a virtual machine. A model which has been tested to work is to use VmWare-server which can be downloaded for free at <http://www.vmware.com>.

Installing SimpleTools

Just un-pack the package file, and remove the old gcc folder:

```
$ cd $IDIR
$ tar xvfz simpletools-2v0.tgz
$ rm -rf gcc-2.6.3
```

Installing SimpleUtils

First un-pack the package file.

```
$ cd $IDIR
$ tar xvfz simpleutils-990811.tar.gz
$ cd simpleutils-990811
```

Before building the code, you need to fix some sources of errors:

- In directory ld find file ldlex.l and replace all instances of yy_current_buffer with YY_CURRENT_BUFFER

```
$ ./configure --host=$HOST --target=$TARGET --with-gnu-as
--with-gnu-ld --prefix=$IDIR
$ make
$ make install
```

Building the simulators

Un-pack the simulator package.

```
$ cd $IDIR
$ tar xvfz simplesim-3v0d.tgz
$ cd simplesim-3.0
```

Modify the Makefile by changing "-O0" with "-O" to enable compiler optimizations. **Note:** The simulator may not work if you enable higher degree of optimizations.

```
$ make config-pisa
$ make
```

you may test the installation of simplesim by:

```
$ ./sim-safe tests/bin.little/test-math
```

Installation of gcc cross-compiler

Un-pack the source code and configure the installation:

```
$ cd $IDIR
$ tar xvfz gcc-2.7.2.3.ss.tar.gz
$ cd gcc-2.7.2.3
$ ./configure --host=$HOST --target=$TARGET --with-gnu-as
--with-gnu-ld --prefix=$IDIR
$ chmod -R +w .
```

Now, you need to modify a number of files in order for the compilation process to work.

- Change the **Makefile** at line 130, by appending `-I/usr/include` to the end of the line
- Edit line 60 of **protoize.c**, and replace `"#include <varargs.h>"` with `"#include <stdarg.h>"`
- To fix an error message about **decl.c**, saying "invalid lvalue in increment", do the following: Edit **obstack.h** at line 341 and change:
 - `*((void **)__o->next_free)++=((void *)datum);`
with
 - `*((void **)__o->next_free++)=((void *)datum);`

Also run the following commands:

- `cp ./patched/sys/cdefs.h ../sslittle-na-sstrix/include/sys/cdefs.h`
- `cp ../sslittle-na-sstrix/lib/libc.a ../lib/`
- `cp ../sslittle-na-sstrix/lib/crt0.o ../lib/`

Next build the compiler:

```
$ make LANGUAGES="c c++" CFLAGS="-O" CC="gcc"
```

This command will lead to a number of error messages which requires you to edit a couple of files:

- append `'\'` at the end of lines 675, 750 and 823 in file **insn-output.c**
- o Do "make LANGUAGES..." again and then
- remove lines 2978-2979 in file **cxxmain.c**, if needed

Now, run:

```
$ make LANGUAGES="c c++" CFLAGS="-O" CC="gcc"
$ make enquire
$ ../simplesim-3.0/sim-safe ./enquire -f > float.h-cross
$ make LANGUAGES="c c++" CFLAGS="-O" CC="gcc" install
```

That should execute with no troubles.

With that, Simple Scalar has been installed, along with its tools, utils, and compilers. Let's now test compiling a new code and running it on top of simplescalar:

Create a new file, **hello.c**, that has the following code:

```
#include<stdio.h>
main()
{
    printf("Hello World!\n");
}
```

then compile it using the following command:

```
$ $IDIR/bin/sslittle-na-sstrix-gcc -o hello hello.c
```

That should generate a file **hello**, which we will run over the simulator:

```
$ $IDIR/simplesim-3.0/sim-safe hello
```

In the output, you should be able to find the following:

```
sim: ** starting functional simulation **
Hello World!
```

Your work is done!

References

1. **SimpleScalar**. SimpleSclar home page. [Online] [Cited: March 26, 2008.] <http://www.simplescalar.com>.
2. **Al-Issa, Khalid**. A SimpleScalar Installation guide. [Online] den 9 July 2007. [Citat: den 26 March 2008.] <http://www.studiokhalid.com/simplescalar/simplescalar.htm>.