史文翰 No.2014211218 Cla.2014211304 shiwenhan@bupt.edu.cn

Simplescalar v3.0e 搭建报告

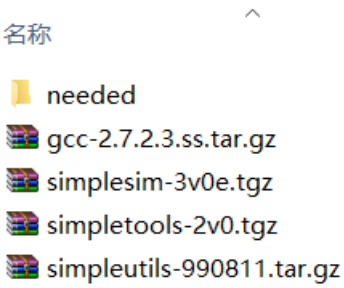
一、 搭建环境

采用 win10 运行虚拟机 VMware 的方法模拟 Linux 环境。虚拟机采用 Ubuntu 16.04 系统，内核版本为 Linux Ubuntu 4.4.0 x86_64。

二、 搭建过程

1、准备所需的文件和环境变量

本次搭建需要如下文件：



其中 needed 为包含 ar 和 ranlib 文件，用于解决安装 gcc 时的溢出错误。其余文件为安装 simplescalar 的必要组件。

设置环境变量如下：

export IDIR=~/shiwenhan/simplescalar/

export HOST=i686-pc-linux

export TARGET=sslittle-na-sstrix

其中 IDIR 表示的是安装目录，此时已经在 shiwenhan 目录下创建了 simplescalar 的工程文件夹。HOST 表示我们需要让模拟器在 Linux 环境下模拟 i686 体系结构。TARGET 指定了系统是小端存储。

用 apt-get 命令确保所需的模块已被安装。

sudo apt-get install build-essential

sudo apt-get install flex

sudo apt-get install bison

sudo apt-get gcc-multilib

sudo apt-get g++multilib

2、解压 Simpletools

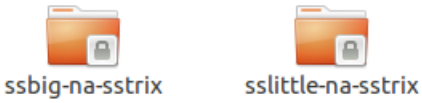
cd $IDIR

tar xvfz simpletools-2v0.tgz

rm -rf gcc-2.6.3

利用 tar 命令在安装目录下解压 simpletools 组件，并删除附带的 gcc 环境，这是由于我们之后会安装新的 gcc 环境。

此时会得到如下两个文件夹。



3、解压并安装 Simpletuils

cd $IDIR

tar xvfz simpleutils-990811.tar.gz

cd simpleutils-990811

解压后进入 simpleutils 目录内，修改 ld 目录下 ldlex.l 文件，将其中的 **yy_current_buffer** 替换为 **YY_CURRENT_BUFFER**。

gedit ldlex.l

```
587   *result = 0;
588   if (YY_CURRENT_BUFFER->yy_input_file)
589     {
590       if (yyin)
591         {
592           *result = read (fileno (yyin), (char *) buf, max_size);
593           if (*result < 0)
594             einfo ("%F%P: read in flex scanner failed\n");
595         }
596     }
597 }
598
```

cd ..

./confing --host=$HOST --target=$TARGET --with-gnu-as --with-gnu-ld --

prefix=$IDIR

make

make install

在此之后，IDIR 目录如下图所示。

Name	Size	Type
share	0 items	Folder
man	1 item	Folder
needed	2 items	Folder
ssbig-na-sstrix	2 items	Folder
include	3 items	Folder
sslittle-na-sstrix	3 items	Folder
lib	8 items	Folder
bin	17 items	Folder
info	30 items	Folder
simpleutils-990811	44 items	Folder
f2c-1994.09.27	62 items	Folder
simplesim-3.0	113 items	Folder
glibc-1.09	162 items	Folder

4、 解压并安装 Simplesim

按照如下指令解压并安装 Simplesim-3v0e.tgz

cd $IDIR

tar xvfz simplesim-3v0e.tgz

cd simplesim-3.0

make config-pisa

make

出现提示信息（由于之前已成功 make，此处只显示了提示信息）

```
shiwenhan@ubuntu:~/shiwenhan/simplescalar$ cd simplesim-3.0
shiwenhan@ubuntu:~/shiwenhan/simplescalar/simplesim-3.0$ make
my work is done here...
shiwenhan@ubuntu:~/shiwenhan/simplescalar/simplesim-3.0$
```

至此，simplesim 的组件安装完毕，可以利用 simplesim 中的 sim-safe 测试一些 test 程序，如下图所示。

./sim-safe tests/bin.little/test-llong

```
shiwenhan@ubuntu: ~/shiwenhan/simplescalar/simplesim-3.0
-max:inst                    0 # maximum number of inst's to execute

sim: ** starting functional simulation **
x+1 = 0x0000000100000001
x-1 = 0x00000000ffffffff
y+1 = 0x0000000200000000
x+y = 0x00000002ffffffff
z*w = 0x01fffffff0000000

sim: ** simulation statistics **
sim_num_insn                 29631 # total number of instructions executed
sim_num_refs                 10185 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate            29631.0000 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                 70464 # program text (code) size in bytes
ld_data_base             0x10000000 # program initialized data segment base
                              8192 # program init'ed `.data' and uninit'ed `.bs
s' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base (highest addres
s in stack)
ld_stack_size                16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
ld_environ_base          0x7fff8000 # program environment base address address
```

LibreOffice Impress

说明 sim-safe 正常工作，至此之前的安装过程正确。

5、 安装 GCC 交叉编译器

运行如下命令加压 gcc 文件。

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

由于版本兼容等问题，需要修改 gcc 源程序来为 make 创造条件。为获取当前目录的写权限，使用如下命令。

chmod -R +w .

- 修改 Makefile 文件，在 130 行的行末添加"**-I/usr/include**"
- 修改 protoize.c 文件，把 60 行处的**<varargs.h>**改为**<stdarg.h>**
- 修改 obstack.h 文件，在 341 行处，把***((void **)__o->next_free)++** 改为 ***((void **)__o->next_free++)**
- 添加补丁，利用如下 cp 命令：

  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/

此时可以开始编译

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

会遇到许多问题，根据操作系统环境的不同，可能会遇到不同的 debug 信息，下面根据在搭建中遇到的信息分类处理。

- insn-output.o 问题：修改 insn-output.c 文件，在第 675 、750、823 行末加上一个反斜杠"\"。由于此文件在 make 之后生成，若执行过 make clean，需要将上述过程重新执行一次。
- sendmsg.c 问题：修改 objc/sendmsg.c 文件，在 35 行处增加宏定义。

  # define STRUCT_VALUE 0
- buffer overflow 问题：将 needed 中的两个文件 ar 和 ranlib 复制（利用 cp 命令）到$IDIR/sslittle-na-sstrix/bin 目录下，并修改它们的权限使其变为可执行文件。

  cd $IDIR/sslittle-na-sstrix/bin/

  chmod +x ar ranlib
- cxxmain.c 问题：修改 cxxmain.c 文件，将 2978 和 2979 行注释掉。
- cc1plus 问题：可能由于机器自带 gcc 版本问题，将 gcc 版本还原为 4.9 版本，利用如下命令：

  sudo apt-get install gcc-4.9

  sudo apt-get install g++-4.9

  sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-4.9 20

  sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-4.9 20

之后重新编译即可通过。

```
root@ubuntu: /home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3
Makefile:940: recipe for target 'libgcc1-test.o' failed
make: *** [libgcc1-test.o] Error 1
shiwenhan@ubuntu:~/shiwenhan/simplescalar/gcc-2.7.2.3$ su root
Password:
root@ubuntu:/home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3# make LANGUAGES="
c c++" CFLAGS="-O" CC="gcc"
./xgcc -B./ -DCROSS_COMPILE -DIN_GCC   -O -I./include-I/usr/include    -c ./libg
cc1-test.c
/tmp/ccUj47iL.s: Assembler messages:
/tmp/ccUj47iL.s:35: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:36: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:37: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:38: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:39: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:40: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:41: Warning: Bignum truncated to 4 bytes
/tmp/ccUj47iL.s:42: Warning: Bignum truncated to 4 bytes
Testing libgcc1.  Ignore linker warning messages.
./xgcc -B./ -DCROSS_COMPILE -DIN_GCC   -O -I./include-I/usr/include libgcc1-test
.o -o libgcc1-test \
  -nostartfiles -nostdlib `./xgcc -B./ --print-libgcc-file-name`
/home/shiwenhan/shiwenhan/simplescalar//sslittle-na-sstrix/bin/ld: warning: cann
ot find entry symbol __start; defaulting to 004000f0
root@ubuntu:/home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3#
```

随后输入命令：

make enquire

../simplesim-3.0/sim-safe ./enquire -f > ！ float.h-cross

make LANGUAGES="c c++" CFLAGS="-O" CC="gcc" install

```
root@ubuntu: /home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3
  /home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3/install.sh -c gcc-cross /ho
me/shiwenhan/shiwenhan/simplescalar//bin/`t='-e s,^,sslittle-na-sstrix-,'; echo
gcc | sed $t`; \
  if [ -d /home/shiwenhan/shiwenhan/simplescalar//sslittle-na-sstrix/bin/. ] ; t
hen \
    rm -f /home/shiwenhan/shiwenhan/simplescalar//sslittle-na-sstrix/bin/gcc; \
    /home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3/install.sh -c gcc-cross /
home/shiwenhan/shiwenhan/simplescalar//sslittle-na-sstrix/bin/gcc; \
  else true; fi; \
else \
  rm -f /home/shiwenhan/shiwenhan/simplescalar//bin/`t='-e s,x,x,'; echo gcc | s
ed $t`; \
  /home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3/install.sh -c xgcc /home/sh
iwenhan/shiwenhan/simplescalar//bin/`t='-e s,x,x,'; echo gcc | sed $t`; \
  rm -f /home/shiwenhan/shiwenhan/simplescalar//bin/sslittle-na-sstrix-gcc-1; \
  ln /home/shiwenhan/shiwenhan/simplescalar//bin/`t='-e s,x,x,'; echo gcc | sed
$t` /home/shiwenhan/shiwenhan/simplescalar//bin/sslittle-na-sstrix-gcc-1 \
    > /dev/null 2>&1 \
    || cp /home/shiwenhan/shiwenhan/simplescalar//bin/`t='-e s,x,x,'; echo gcc |
 sed $t` /home/shiwenhan/shiwenhan/simplescalar//bin/sslittle-na-sstrix-gcc-1; \
  mv /home/shiwenhan/shiwenhan/simplescalar//bin/sslittle-na-sstrix-gcc-1 /home/
shiwenhan/shiwenhan/simplescalar//bin/sslittle-na-sstrix-gcc; \
fi
root@ubuntu:/home/shiwenhan/shiwenhan/simplescalar/gcc-2.7.2.3#
```

完成 gcc 交叉编译器的安装。

三、 输入输出测试

在 IDIR 目录下测试 test.c。

```
Open
#include<stdio.h>
main()
{printf("Hello World!\nI'm Wenhan Shi!\n");}
```

输入如下命令进行 gcc 交叉编译：

./sslittile-na-sstrix-gcc ../test.c

此时 bin 目录下产生 a.out 输出文件，最后我们用 simplesim3.0 中的 sim-safe 解析它。

cd ..

./simplesim-3.0/sim-safe ./bin/a.out

得到如下结果：

```
root@ubuntu: /home/shiwenhan/shiwenhan/simplescalar
sim: ** starting functional simulation **
Hello World!
I'm Wenhan Shi!

sim: ** simulation statistics **
sim_num_insn                  7573 # total number of instructions executed
sim_num_refs                  4109 # total number of loads and stores executed
sim_elapsed_time                 1 # total simulation time in seconds
sim_inst_rate            7573.0000 # simulation speed (in insts/sec)
ld_text_base             0x00400000 # program text (code) segment base
ld_text_size                 71968 # program text (code) size in bytes
ld_data_base             0x10000000 # program initialized data segment base
ld_data_size                  8320 # program init'ed `.data' and uninit'ed `.bs
s' size in bytes
ld_stack_base            0x7fffc000 # program stack segment base (highest addres
s in stack)
ld_stack_size                16384 # program initial stack size
ld_prog_entry            0x00400140 # program entry point (initial PC)
ld_environ_base          0x7fff8000 # program environment base address address
ld_target_big_endian             0 # target executable endian-ness, non-zero if
 big endian
mem.page_count                  26 # total number of pages allocated
mem.page_mem                   104k # total size of memory pages allocated
mem.ptab_misses                 26 # total first level page table misses
```

说明 simplescalar 至此安装正确，且测试结果正确。

主要参考：

1、 http://www.cnblogs.com/blue163/p/4928394.html

2、 http://www.cnblogs.com/darkknightzh/p/6194031.html