

**课程名称**  \_ \_\_计算机体系结构 \_\_\_

**学院** \_ 计算机学院 \_\_\_\_\_\_\_\_

**专业** \_ 计算机科学与技术\_\_\_

**班级**  \_ 2014211306\_\_\_\_\_\_\_\_\_

**学号** \_ 2014211292\_\_\_\_\_\_\_\_\_

**姓名** \_ 张有杰\_\_\_\_\_\_\_\_\_\_\_\_\_

实验 1 SimpleScalar 的安装与配置

一、实验目的

安装实验平台并熟悉实验环境,为今后的学习做准备。

二、实验布置及要求

1. 搭建 SimpleScalar(3.0e 版本)模拟器环境,并运行 helloworld,写详细的搭建报告, 截图展示具体过程,用 A4 纸打印交 (http://simplescalar.com/)

2. 利用 SimpleScalar 工具链编译 Mibench‐auotomotive 基准测试包。

三、实验环境

操作系统：Ubuntu 17.04

四、实验过程

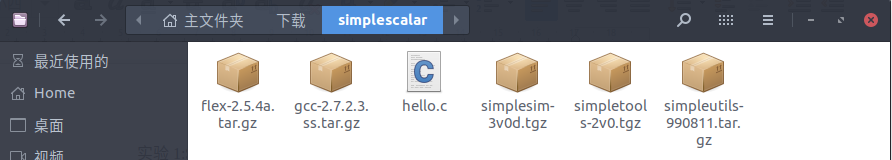
<安装 simplescalar>

1. 下载安装文件

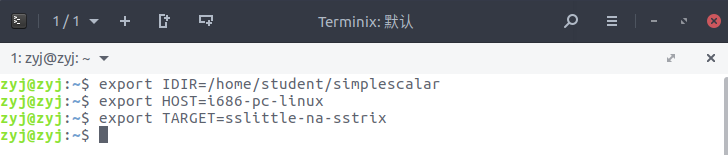
讲义中的部分安装文件在官网上不存在或者链接失效，因此直接从 csdn 上下载，链接如下：

<http://download.csdn.net/download/ldld1717/9348445>

该链接包含的压缩包中包含了实验所需要的依赖文件。下载完成后，解压得到下图所示的文件：

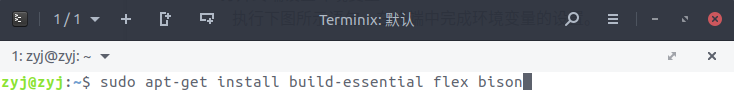


2. 打开终端设置环境变量

 执行下图所示语句，在终端中完成环境变量的设置。

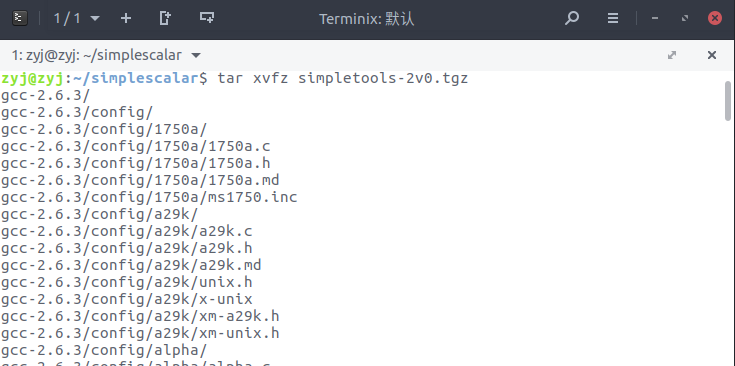
3. 在student主目录下创建目录 simplescalar，并将4个安装包移动到该目录

由于我将安装包都放在了桌面，因此mv命令中第一个参数路径与讲义中不同。

4. 使用命令更新Ubuntu下的软件包build-exxential,flex,bison:

命令正常完成，成功更新相应的软件包。

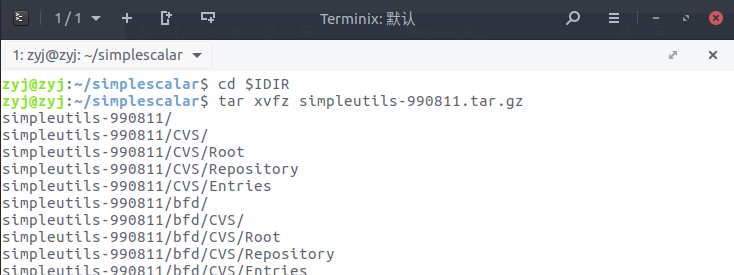
5. 安装SimpleTools

(1)解压 simpletools-2v0.tgz

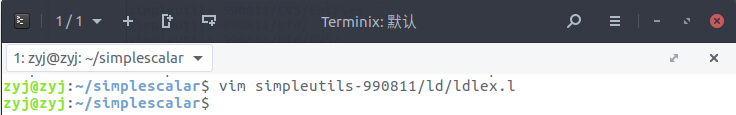
(2)删除旧的 gcc 文件夹



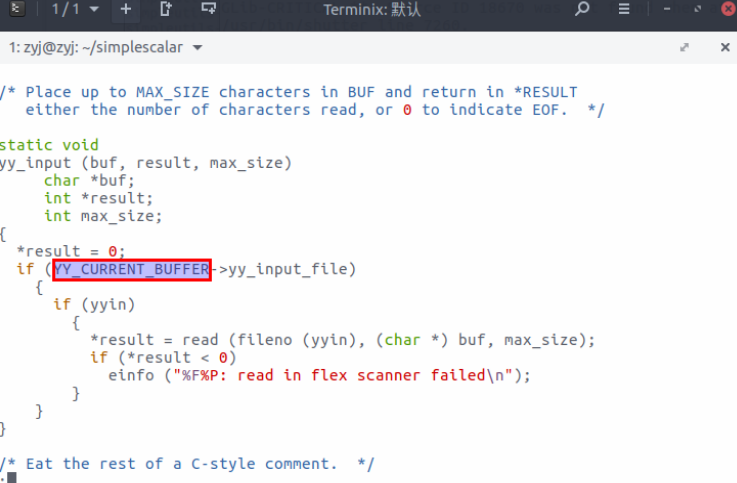
6.安装 SimpltUtils

(1)解压包文件

(2)修改代码错误

使用 vim 打开目录下的 simpleutils-990811/ld/ldlex.l 文件

将 yy\_current\_buffer 修改为 YY\_CURRENT\_BUFFER 。



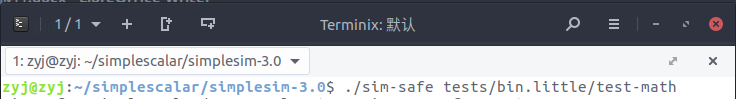
(3)配置、编译

执行 make CFLAGS=-0 以及 make install 之后编译成功。

7.安装模拟器

(1)解压并安装

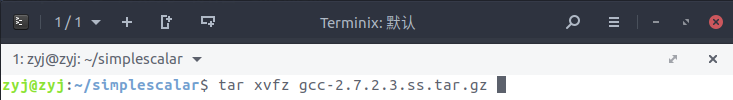
执行 make config-pisa 与 make 语句后安装成功。

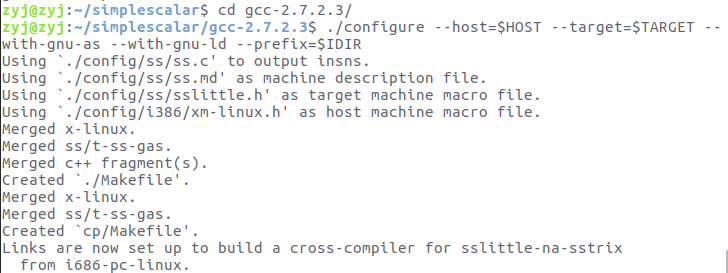
(2)测试

执行以上语句进行测试，测试结果如下：

8. 安装交叉编译器

(1)解压源代码



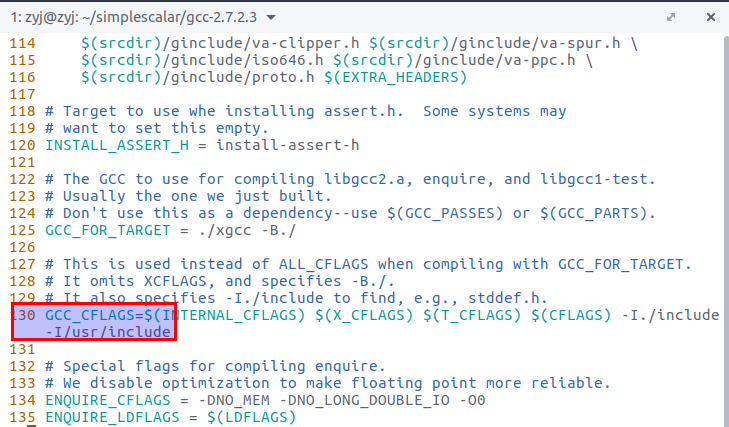
(2)配置安装

(3)修改代码文件读写权限

注意命令后面的 . 符号

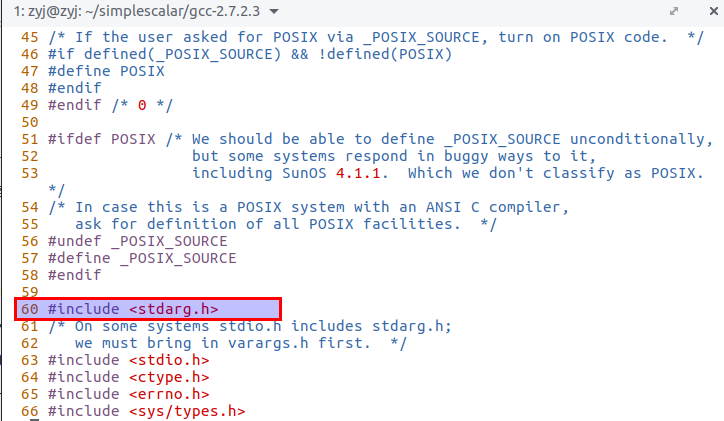
(4)为了使编译顺利进行，做以下修改：

a. 修改 Makefile 的130行 行末加-I/usr/include



b.修改 protoize.c 的第 60 行 , 将“#include <varargs.h>” 替换为“#include

<stdarg.h>”



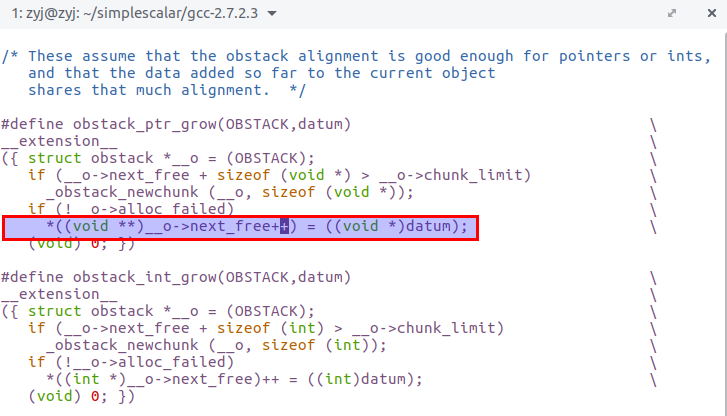
c.为了解决 decl.c 的一处错误信息, 名为“invalid lvalue in increment”, 做以下修改:

编辑 obstack.h 的第 341 行并将:

\*((void \*\*)\_\_o->next\_free)++=((void \*)datum);\

替换为:

\*((void \*\*)\_\_o->next\_free++)=((void \*)datum);\

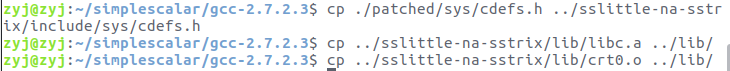


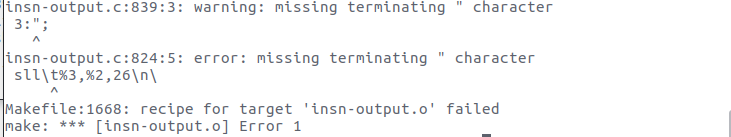
并运行以下命令:

$ cp ./patched/sys/cdefs.h ../sslittle-na-sstrix/include/sys/cdefs.h

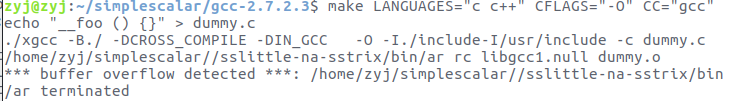
$ cp ../sslittle-na-sstrix/lib/libc.a ../lib/

$ cp ../sslittle-na-sstrix/lib/crt0.o ../lib/



d. 构建编译器

执行 $ make LANGUAGES=c CFLAGS=-O CC="gcc -m32" 命令得到下图错误：

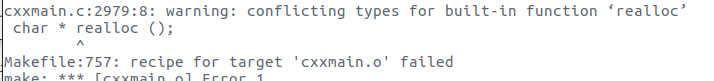
追加 ‘\’到 insn-output.c 的 675, 750 和 823 行末尾后重新构建

再次执行“make LANGUAGES...”后,碰到错误"buffer overflow",这是由于 Ubuntu 版

本较新造成的。下载 ar 和 ranlib 放入$IDIR/sslittle-na-sstrix/bin 目录然后使用命令:

chmod +x ar ranlib

修改 ar 及 ranlib 的权限为可执行。

再次执行“make LANGUAGES...”后，又出现文件关于cxxmain.c的错误，可以通过修改cxxmain.c文件，把里面第2978和2979行注释掉，然后重新编译。

执行

make enquire

../simplesim-3.0/sim-safe ./enquire -f > float.h-cross 以安装交叉编译器

在执行这条命令时，程序很久都没有反应。上网查询后在 > 号后加上感叹号，如下：

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

重新执行后，安装成功

$ make LANGUAGES="c c++" CFLAGS="-O" CC="gcc" install 构建编译器

执行成功，完成构建

编写如下的 hello.c:

#include<stdio.h>

main()

{

printf("Hello World!\n");

}

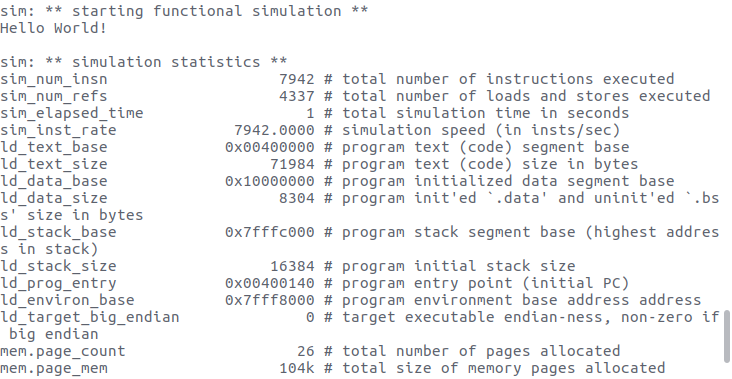
用如下命令编译：

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

用如下命令执行：

$IDIR/simplesim-3.0/sim-safe hello

执行结果如下图所示。



测试成功，安装完成！

<安装mp\_simplesim>

1.移动到 SimpleScalar 安装目录:

$ cd $IDIR

2.将 mp\_simplesim.tar.gz 解压至该目录:

$ tar xvzf mp\_simplesim.tar.gz

3.进入 mp\_simplesim 目录:

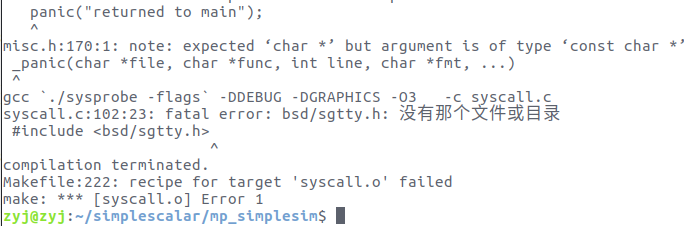
$ cd mp\_simplesim

4.打开 Makefile 文件,将 SS\_BIN\_PATH 设置为交叉编译器的路径:

SS\_BIN\_PATH = /home/student/simplescalar/sslittle-na-sstrix/bin

5.编译:

$ make

遇到错误提示如下：

将 syscall.c 第 102 行中替换为:

#include <sgtty.h>

并保存。

6.继续运行 make,出现新的错误:

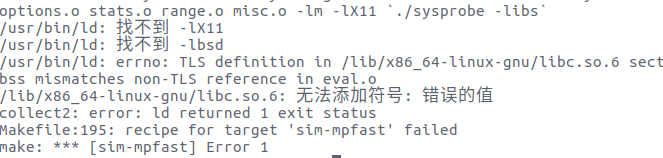
syscall.c:823: 错误: ‘TIOCGETP’未声明(在此函数内第一次使用)

syscall.c:823: 错误: (即使在一个函数内多次出现,每个未声明的标识符在其

syscall.c:823: 错误: 所在的函数内也只报告一次。)

syscall.c:826: 错误: ‘TIOCSETP’未声明(在此函数内第一次使用)

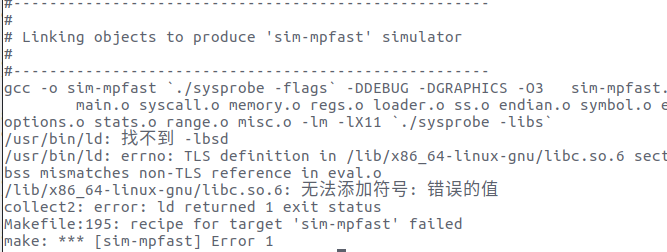
按照实验指导书完成修改，继续 make

7. 继续make 出现以下错误：

解决方法:安装缺失的软件包 libx11-dev:

$ sudo apt-get install libx11-dev

8.继续make 出现以下错误：

解决方法:

打开文件 eval.c,64-66 行内容为:

#if defined(\_\_CYGWIN32\_\_)

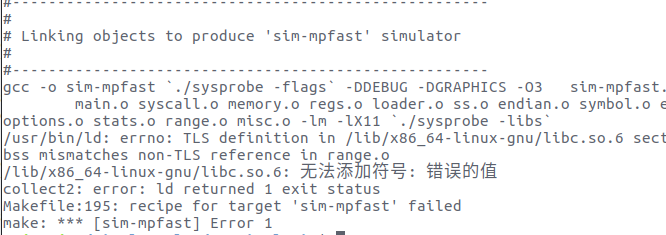
#include <errno.h>

#endif

删除或注释掉 64 及 66 行, 保留:

#include <errno.h>

并保存。

9. 继续 make，出现以下错误：

解决办法：

打开文件 range.c,64-66 行内容为:

#if defined(\_\_CYGWIN32\_\_)

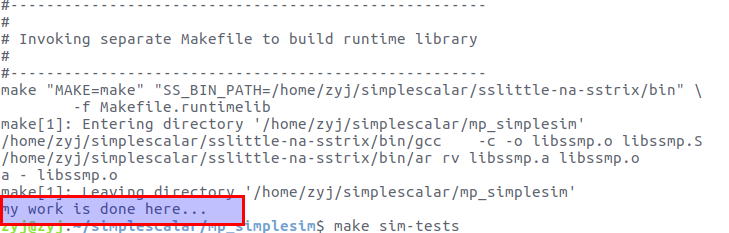
#include <errno.h>

#endif

删除或注释掉 64 及 66 行, 保留:

#include <errno.h>

并保存。

10.继续 make 提示安装成功

五、实验总结

在这次实验中，我在 Ubuntu 17.04 系统上完成了 SimpleScalar 以及 mp\_simplesim 两个实验环境的安装和配置。在整个安装过程中，出现了许多问题，但是都通过实验指导书以及上网查询资料等方式解决了问题，完成了安装。整个环境的安装与配置是比较考验人耐心的一个过程，而且中间出了问题时很可能需要重头再来。但是，只要耐下心来找出并解决问题，最后一定能完成实验。

这次实验锻炼了我的动手能力，考验了我的耐心，为之后的实验打下了基础。