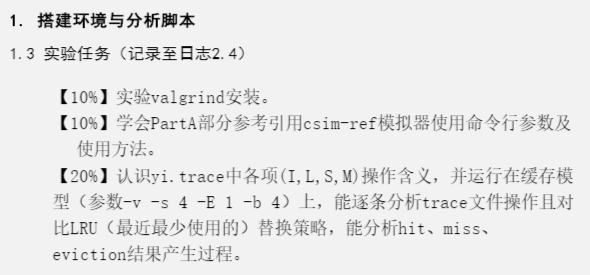
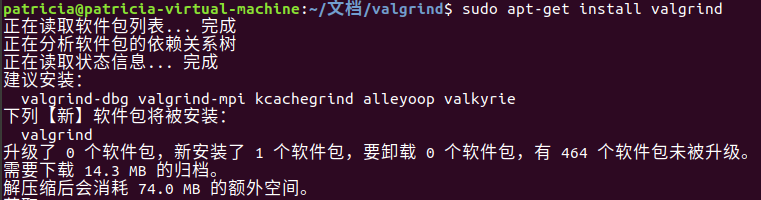
实验日志一

实验要求：



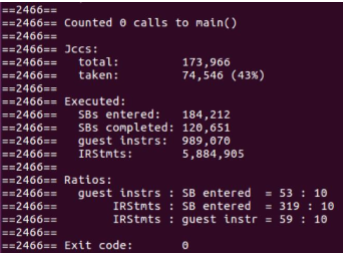
一：valgrind的安装

执行命令：sudo apt-get install valgrind 安装



安装完成 验证一下

执行命令：valgrind --log-fd=1 --tool=lackey -v --trace-mem=yes ls –l



验证成功。

二：学会 PartA 部分参考引用 csim-ref 模拟器使用命令行参数及使用方法

  cache模拟器需要能处理一系列如下的命令：

        Usage: ./csim-ref [-hv] -s <s> -E <E> -b <b> -t <tracefile>

        其中各参数意义如下：

①-h：输出帮助信息的选项；②-v：输出详细运行过程信息的选项；

③-s：组索引的位数(意味着组数S=2^s)；④-E：每一组包含的行数；

⑤-b：偏移位的宽度(意味着块的大小为B=2^b);

⑥-t：输入数据文件的路径(测试数据从该文件里面读取)。

三：认识 I、L、S、M 操作含义，并运行在缓存模型上，逐条分析文件操作且对比 LRU 替换 策略，能分析 hit、miss、eviction 结果产生过程 “

“I”表示指令加载，“L”表示数据加载，“S”表示数据存储，“M”表示数据修改（即数据存储之后的数据加载）。每个“I”前面都没有空格。每个“M”，“L”和“S”之前总是有空格。地址字段指定一个32位的十六进制存储器地址。

大小字段指定操作访问的字节数；

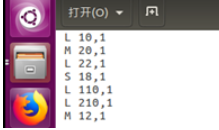
通俗地解释一下各种操作：

①对于‘I’指定地操作，实验说明中提到，我们不需要考虑：

意思就是valgrind运行地时候第一个指令总是为操作‘I’。

②对于‘L’以及‘S’指定的操作，我们简单地可以认为这两个操作都是对某一个地址寄存器进行访问（读取或者存入数据）；

③对于‘M’指定的操作，可以看作是对于同一地址连续进行‘L‘和’S‘操作。



其解释如下：

①对于地址0x10进行访问：

0x10=0000...00010000，偏移值为最低四位，故S=1;访问结果为mis;

②连续对地址0x20进行连续两次访问：

0x20=000...00100000，S=2;结果为第一次mis，第二次hit；

③对地址0x22进行访问：

0x22=000...00100100，S=2;由于操作②以将该块存入高速缓存，故结果为hit;

④对地址0x18进行访问：

0x18=000...00011000，S=1;由于操作①以将该块存入高速缓存，故结果为hit;

⑤对地址0x110进行访问：

0x110=0...000100010000，S=1;虽然操作①使得第一组(只有一行有效)，但是这里的标志位的值Tag为1 故结果为先mis，后eviction;

⑥对地址0x210进行访问：

0x210=0...001000010000，S=1;同操作⑤，但是这里的标志位的值为2，不匹配故结果为先mis，后evicton;

⑦对地址0x12进行连续两次访问：

0x12=000...00000010010，S=1;由于标志位不匹配，故第一次访问时mis，并evicton第二次访问时当然就是hit。

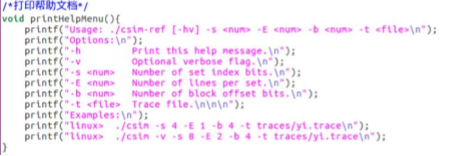
四：编写结构体

get\_Opt()、helpPrint()、checkOptArg()。initCache()、freeCache()

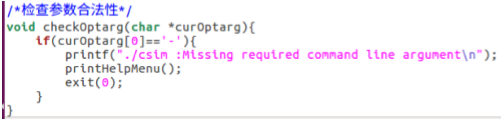
（1）get\_Opt()



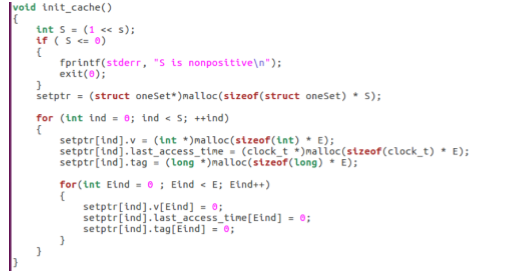
（2）helpPrint()



（3）checkOptArg()



1. initCache()



1. freeCache()

