

**实验报告**



**题目： 熟悉Linux系统及其相关软件环境**

**班 级： 2019211315**

**学 号： 2019211608**

**姓 名： 姜彦**

**学 院： 计算机学院**

**2020年 10 月 16 日**

1. 实验目的
2. 熟悉linux操作的基本操作；
3. 掌握gcc编译方法；
4. 掌握gdb的调试工具使用；
5. 掌握objdump反汇编工具使用；
6. 熟悉理解反汇编程序（对照源程序与objdump生成的汇编程序）。

二、实验环境（10分）

简述使用的工具

1. Xshell（服务器：10.120.11.12）：Xshell是一个强大的安全终端模拟软件，可以在Windows界面下用来访问远端不同系统下的服务器，从而达到远程控制终端的目的
2. Linux：一套免费使用和自由传播的类UNIX操作系统，是一个基于POSIX和Unix的多用户、多任务、支持多线程和多CPU的操作系统
3. Gcc编译器：由GNU开发的编程语言译器，其套件包括C、C++、 Objective-C、 Fortran、Java、Ada和Go语言前端，也包括了这些语言的库（如libstdc++，libgcj等。）
4. GDB调试工具：UNIX及UNIX-like下的调试工具
5. Objdump命令反汇编：objdump 有点像那个快速查看之类的工具，就是以一种可阅读的格式让你更多地了解二进制文件可能带有的附加信息。

**报告邮寄（最迟时间：2020年10月28日晚23：59）：**

**大一班（11-15班）：[bywork2019@bupt.edu.cn](mailto:bywork2019@bupt.edu.cn)**

**大二班（1班，16-19班）：[clavicle@bupt.edu.cn](mailto:clavicle@bupt.edu.cn)**

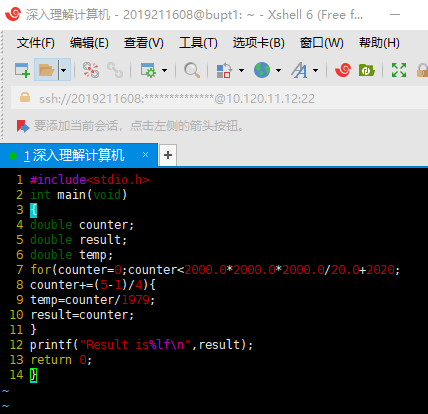
**大三班（临时班）：[2019140578@bupt.edu.cn](mailto:2019140578@bupt.edu.cn)**

三、实验概况

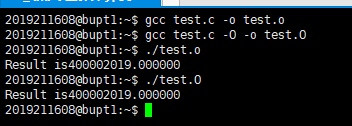
简述实验内容和基本设想

实验内容一（15分）

在linux环境下，编辑课件中源程序（注意程序的完整性）（包含源程序的开发环境截图），

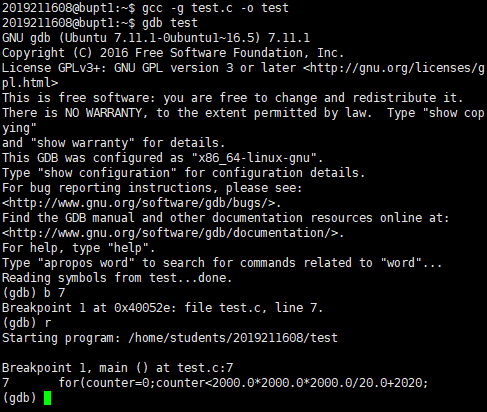


采用gcc编译该程序（要求分别采用-o和-O参数，并比较两者性能，编译指令截图），

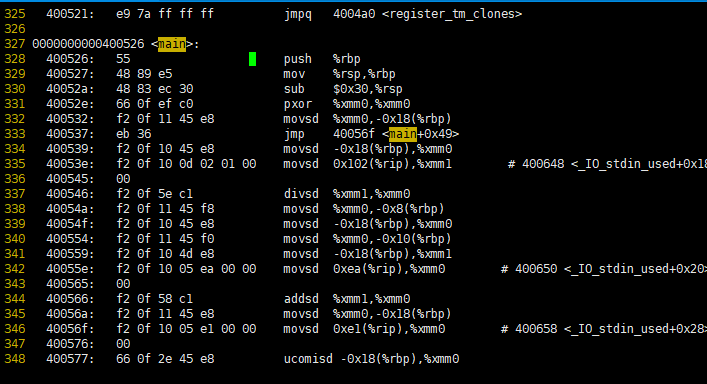


添加-O参数后程序的运行速度有明显的提高

采用gdb进行调试，让程序运行到for函数语句（调试截图），



运用objdump工具生成汇编程序（给出main函数的汇编程序截图）



#include<stdio.h>

int main(void)

{

double counter;

double result;

double temp;

for(counter=0;counter<2000.0\*2000.0\*2000.0/20.0+2020;

counter+=(5-1)/4){

temp=counter/1979;

result=counter;

}

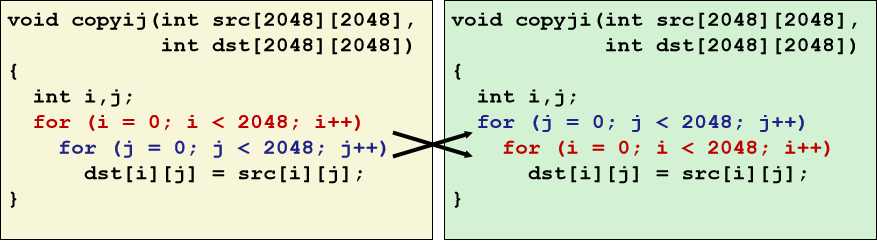
printf(Result is%lf\\n,result);

return 0；

}

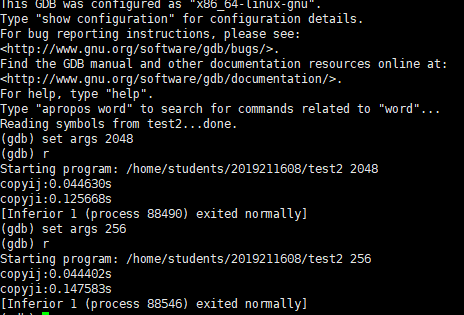
实验内容二（15分）

在linux环境下，分别打印输出如下算法所需时间



分别设置不同优化参数，给出运行时间

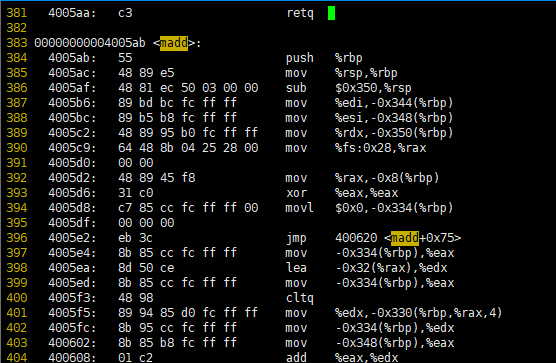




实验内容三（30分）

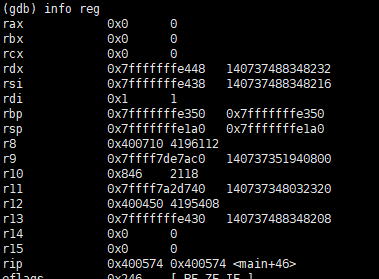
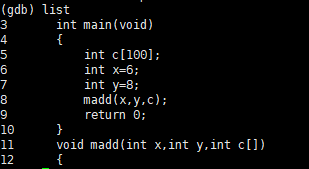
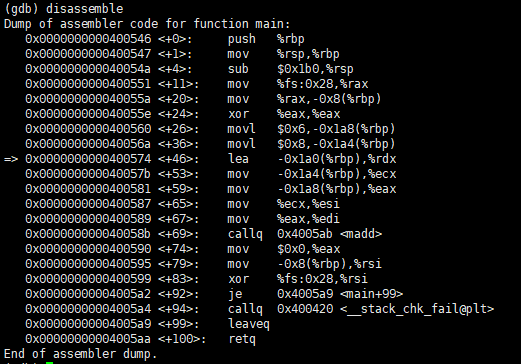
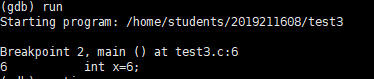
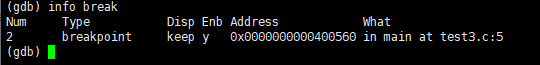
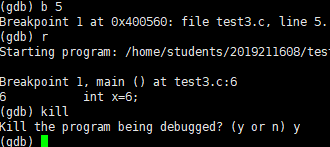
现有int型数组a[i]=i-50,b[i]=i+y，其中y取自于学生本人学号2019211x\*y的个位。登录bupt1服务器，在linux环境下使用vi编辑器编写C语言源程序，完成数组a+b的功能，规定数组长度为100，函数名为madd（），数组a，b均定义在函数内，采用gcc编译该程序（不使用优化选项），

使用objdump工具生成汇编程序，找到madd函数的汇编程序，给出截图；

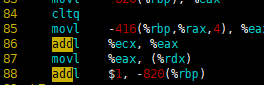


用gdb进行调试，练习如下gdb命令，给出截图；

gdb、file、kill、quit、break、delete、clear、info break、run、continue、nexti、stepi、disassemble、list、print、x、info reg、watch

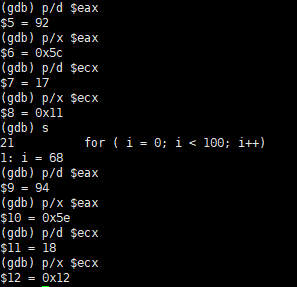


找到a[i]+b[i]对应的汇编指令，指出a[i]和b[i]位于哪个寄存器中，给出截图；



a[i]位于%eax,b[i]位于%ecx

使用单步指令及gdb相关命令，显示a[xy]+b[xy]对应的汇编指令执行前后操作数寄存器十进制和十六进制的值，其中x，y取自于学生本人学号2019211x\*y的百位和个位。



实验内容四（加分项，20分）

任选高复杂度算法（具体算法自选，类型分为高计算量类型和高内存需求类型2类算法），通过设置不同优化参数，分析算法的运行效率

四、实验步骤（60-80分）

实验内容一:

输入touch test.c创建test.c文件

vi test.c编辑代码

：wq保存退出

gcc test.c -o test.o

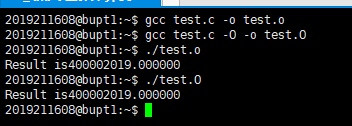
gcc test.c -O -o test.O

分别编译生成test.o，test.O

./test.o

./test.O

分别运行程序

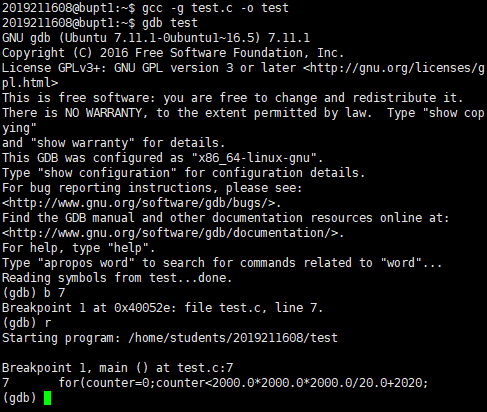


gcc -g test.c -o test

gdb test调试程序

b 7

r



设置断点并运行

Objdump -S -d test > test.txt反汇编

/main找到main函数

实验内容二:

创建编辑test2.c文件

设置时间节点，计算函数所用时间

编译test2.c文件并运行



设置不同参数，多次比较运行时间

实验内容三:

创建编辑test3.c文件

采用gcc（不使用优化选项）编译该程序

使用objdump工具生成汇编程序

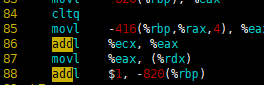
/madd找到对应的madd函数

在gdb调试依次使用练习如下命令

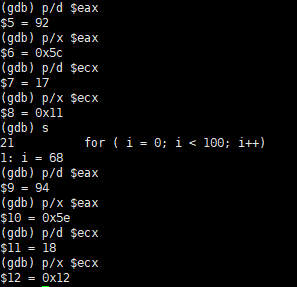
gdb、file、kill、quit、break、delete、clear、info break、run、continue、nexti、stepi、disassemble、list、print、x、info reg、watch

编译生成汇编代码找到a[i]+b[i]对应的汇编指令

并指出a[i]和b[i]位于的寄存器



使用单步指令s,p/d显示a[68]+b[68]对应的汇编指令执行前后操作数寄存器十进制和p/h十六进制的值



五、实验分析（20分）

实验按照编写代码的思路进行，依次创建编辑编译运行等

实验前假设-O是对代码进行优化，copyij速度快都在实验结果的到验证，汇编代码与实验前认为的与源代码一一对应的假设不一样

大部分实验结果与设想一致，小部分数据有些许偏差，但不影响整体结果

六、实验总结（10分）

在使用objdump时一开始使用的-s参数而非-S参数，导致生产的汇编代码产生乱码，在改为-S后恢复正常

编写代码若遇到错误，调试修改相对比较麻烦与不熟悉

gdb的某些指令容易记错，寄存器前要使用$符号

七、诚信声明（不签扣10分）

需要填写如下声明，并在底部给出手写签名的电子版。

在完成本次实验过程中，我曾分别与以下各位同学就以下方面做过交流：

1、简单描述交流内容，例如：来自\*\*\*的建议，采用\*\*\*方式\*\*\*

2、

此外，我还参考了以下资料：

1. 网址等

在我提交的程序中，还在对应的位置以注释形式记录了具体的参考内容。

我独立完成了本次实验除以上方面之外的所有工作，包括分析、设计、编码、调试与测试。

我清楚地知道，从以上方面获得的信息在一定程度上降低了实验的难度，可能影响起评分。

我从未使用他人代码，不管是原封不动地复制，还是经过某些等价转换。

我未曾也不会向同一课程（包括此后各届）的同学复制或公开我这份程序的代码，我有义务妥善保管好它们。

我编写这个程序无意于破坏或妨碍任何计算机系统的正常运行。

我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按照0分计。

（签名）