## 实验1：Linux编程基础实验

1. 实验目的
   1. 学会自己安装Linux系统；
   2. 学会配置简单的Linux开发环境；
   3. 在Linux下完成简单编程练习并熟悉各种命令行工具的使用方法。
2. 实验内容
   1. **Linux安装和配置**
      1. 请分别按照[Linux安装](#_二、Linux安装)和[Linux配置](#_三、Linux配置)按步骤完成VirtualBox和Linux系统安装和环境配置。

**要求1：安装32位版本的Ubuntu并配置好所有工具。**

* + 1. 请使用man查询Vim/Git/GCC/AS/OBJDUMP/GDB版本的命令，然后使用查到的命令打印出对应版本，将版本截图贴在实验报告中。
    2. 写下你在安装过程中遇到的问题，并说明你是如何解决的（没有可不写）。
  1. **Linux下的编程实践**
     1. 使用文件管理器或者mkdir在主目录下创建工作目录workspace。

cd ~

mkdir workspace

* + 1. 在workspace目录下创建目录lab01，在lab01下创建目录151220000（改成你自己的学号，后面的同理）。

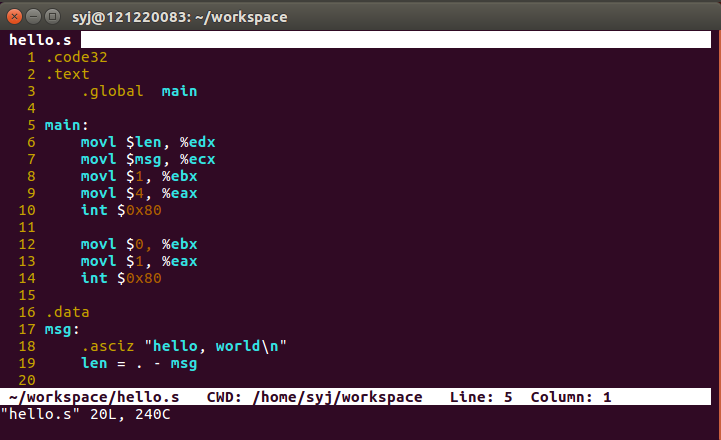
cd workspace

mkdir lab01

cd lab01

mkdir 151220000

* + 1. 将工作目录切换为~/workspace/lab01/151220000，然后使用Gedit或Vim编辑器编写汇编语言的表白程序（可以直接修改下图中的代码让它打印出一个爱心），并保存为heart.S（程序代码参考**[汇编编译工具as](#_汇编编译工具as)**，或下图）。



**要求：爱心形状如下，前后不要有多余的空格，爱心之后紧跟自己的学号，最后一行为空行（请严格遵守格式，助教的自动批改脚本非常不智能）。**

touch heart.S # 创建文件

# 编辑 heart.S

as -o heart.o heart.S # 编译

gcc -o heart heart.o # 链接

./heart # 运行

TIPS：汇编的编译链接命令的详细介绍在教程[汇编编译工具as](#_汇编编译工具as)中。**请提交修改的heart.S代码文件，而不是反汇编文件。正确的输出样例如下：**

\* \*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*

151220000

* 1. **熟悉工具：**
     1. 使用objdump的-D选项反汇编heart.o文件，找到你学号的位置并截图（提示：ASCII码数字的16进制）。

**objdump –D heart.o > dog**

cat dog

**要求：在实验报告中贴出截图，并说明哪里是你的学号。**

* + 1. 编写简单的C语言源程序hello.c，通过预处理、编译、汇编、链接四个步骤将C语言源程序转换为可执行文件，即hello.c -> hello.i -> hello.s -> hello.o -> hello

**要求：在实验报告中贴出截图，每个步骤的命令**。

* + 1. 通过创建或修改~/.vimrc文件，使你的vim支持显示行号。

set nu

**要求：请自行查找如何修改，无需写在实验报告。**

* 1. **数据的表示范围及不同类型的数据长度实验。**

代码：

sqr.c

#include <stdio.h>

int main()

{

int i,j;

i=40000;

j=i\*i;

printf("The 40000\*40000 is %d\n", j);

i=50000;

j=i\*i;

printf("The 50000\*50000 is %d\n", j);

return 0;

}

# gcc –o sqr sqr.c

1. 将输出结果导出，说明发生这种现象的原因？
2. 寻找在该程序中保证结果正确的最大整数值？（不需要提交这部分的代码，请在实验报告中写答案）
3. 应如何修改程序，才能保证结果都正确？（请提交修改后的程序代码）
   1. **矩阵运算执行时间比较**
4. 比较两个矩阵复制函数的执行时间。（请提交可正确编译运行的代码，并将实验结果截图贴在实验报告中）
5. 说明为什么会出现这个差别。

代码：

matrix.c

#include <stdio.h>

#include <time.h>

void copyij(int src[2048][2048], int dst[2048][2048]){

int i, j;

for (i = 0; i < 2048; i++){

for (j = 0; j < 2048; j++){

dst[i][j] = src[i][j];

}

}

}

void copyji(int src[2048][2048], int dst[2048][2048]){

int i, j;

for (j = 0; j < 2048; j++){

for (i = 0; i < 2048; i++){

dst[i][j] = src[i][j];

}

}

}

int src[2048][2048];

int dstij[2048][2048];

int dstji[2048][2048];

int main(){

int t, m;

for (t = 0; t < 2048; t++){

for (m = 0; m < 2048; m++){

src[t][m] = t + m;

}

}

clock\_t startij, finishij, startji, finishji;

startij = clock();

copyij(src, dstij);

finishij = clock();

double durationij = (double)(finishij - startij) / CLOCKS\_PER\_SEC;

printf("copyij %f s\n", durationij);

startji = clock();

copyji(src, dstji);

finishji = clock();

double durationji = (double)(finishji - startji) / CLOCKS\_PER\_SEC;

printf("copyji %f s\n", durationji);

return 0;

}

# gcc -o matrix matrix.c

**【选做】Git相关选做（不加分也不扣分）**

* + 1. 注册并成为GitHub的一员，为Ubuntu系统配置git的name和Email，并在github上添加Ubuntu系统的ssh-key，Git的用法可以通过[上文的教程](#_2.1__源代码版本控制工具git)学会（推荐廖雪峰的教程，通读教程并学会Git的基本操作大约需要2小时）。
    2. 使用Git对实验1进行代码版本管理：
       1. 使用git init初始化~/workspace/lab01/151220000为版本库，然后使用git记录你认为有意义的代码改动（推荐每实现一个功能进行一次提交）。

cd ~/workspace/lab01/151220000

git init

git add <filename>

git commit –m ‘<what you changed>’

* + - 1. 学习使用.gitignore将不需要进行版本追踪的文件（例如pdf，word等）从Git追踪中排除。

**提交要求：**

请在规定时间内提交一个压缩包151220000.zip到课程网站（注意修改学号和压缩格式，不接受过期提交）。压缩包内部应该是一个目录。

**实验报告请精简，尽量不要多于2页。**

151220000.zip解压后获得目录内容如下（注意文件名大小写和每一个文件的提交要求）：

151220000

|----heart.S

|----sqr.c

|----matrix.c

|----report.pdf