## 实验二：C编程环境

返回[目录](#_目录)

### 实验目的

1． 熟悉Linux下C程序设计的环境；

2． 对系统调用有初步了解。

### 实验时间

3学时

### 实验内容

1．Linux下C语言程序的开发过程

a、在用户主目录下用vi编辑C语言源程序（源程序已附后），如：$**vi hello.c**。

b、用gcc编译C语言源程序：$**gcc ./hello.c -o example**

这里gcc是Linux下的C语言程序编译器（GNU C Compiler），.**/hello.c**表示待编译的源文件是当前工作目录下的hello.c，**-o example**表示编译后产生的目标代码文件名为example。

c、若编译不正确，则进入vi修改源程序，否则，运行目标代码：$**./example** 。

注意：

a、这只是gcc最基本的用法，其他常用选项有：-c , -S , -O , -O2, -g 等。

b、调试程序可以用gdb（GNU debugger）。

2．编辑、调试下面c语言程序，说明该程序的功能。

附：hello.c源程序

＃include <stdio.h>

main()

{

int n,a[200],carry,temp,i,j,digit = 1;

printf("Please input n:");

scanf("%d",&n); a[0] = 1;

for( i = 2; i <= n; ++i)

{

for( j = 1, carry = 0; j <= digit; ++j)

{

temp = a[j-1] \* i + carry;

a[j-1] = temp % 10;

carry = temp / 10;

} while(carry)

{ a[++digit-1] = carry % 10; carry /= 10;

}

}

printf("Result is:\n%d ! = ",n);

for( i = digit; i >=1; --i)

{

printf("%d",a[i-1]);

}

printf("\n");

}

**实验报告**

完成2.并写出实验报告，报告应具有以下内容：

* 程序（含注释）
* 实验结果展示
* 实验过程中出现的问题及解决方法

**程序：**

//高精度乘法，求的是n！

#include <stdio.h>

int main()

{

int n,a[200],carry,temp,i,j,dight=1;

printf("Please input n:");

scanf("%d",&n);

a[0]=1;

for (i=2;i<=n;i++)

{

for (j=1,carry=0;j<=dight;++j)

{

//a(乘数1)的第j位与i(乘数2)相乘

temp=a[j-1]\*i+carry;

//a的第j位(结果)取余

a[j-1]=temp%10;

//a的第j+1位获得进位

carry=temp/10;

}

//若存在进位(结果最高位carry大于10),则继续执行下去，

//每次最高位的位数1，数值除以10，直到没有进位

while (carry)

{

a[++dight-1]=carry%10;

carry/=10;

}

}

printf("Result is:\n%d != ",n);

for (i=dight;i>=1;--i)

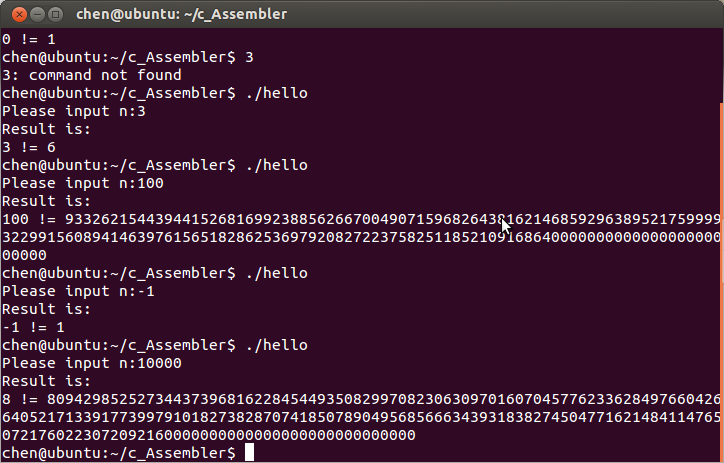
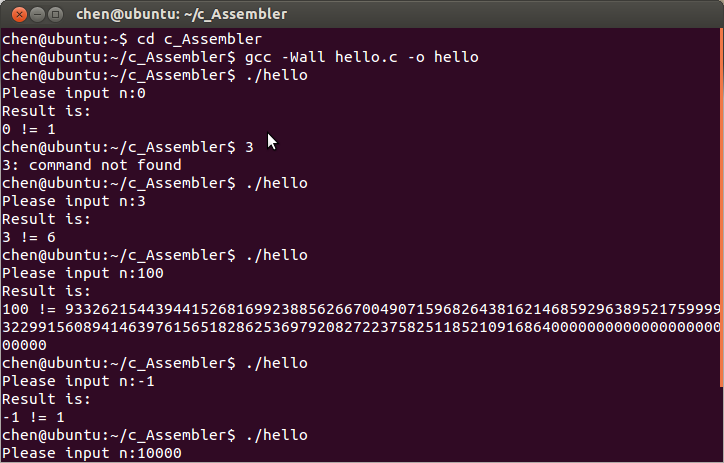
printf("%d",a[i-1]);

printf("\n");

return 0;

}

**结果展示：**



高精度乘法的优势：可处理较大数值的阶层求值。

可看到，输入-1和1000后，结果是错误的，所以要添加语句，**修改后**如下：

//高精度乘法，求的是n！

#include <stdio.h>

int main()

{

int n,a[200],carry,temp,i,j,dight=1;

printf("Please input n:");

scanf("%d",&n);

**if (n<0)**

**{**

**printf("Wrong input\n");**

**return 0;**

**}**

**if (n>120)**

**{**

**printf("the result is too large.\n");**

**return 0;**

**}**

a[0]=1;

for (i=2;i<=n;i++)

{

for (j=1,carry=0;j<=dight;++j)

{

//a(乘数1)的第j位与i(乘数2)相乘

temp=a[j-1]\*i+carry;

//a的第j位(结果)取余

a[j-1]=temp%10;

//a的第j+1位获得进位

carry=temp/10;

}

//若存在进位(结果最高位carry大于10),则继续执行下去，

//每次最高位的位数1，数值除以10，直到没有进位

while (carry)

{

a[++dight-1]=carry%10;

carry/=10;

}

}

printf("Result is:\n%d != ",n);

for (i=dight;i>=1;--i)

printf("%d",a[i-1]);

printf("\n");

return 0;

}

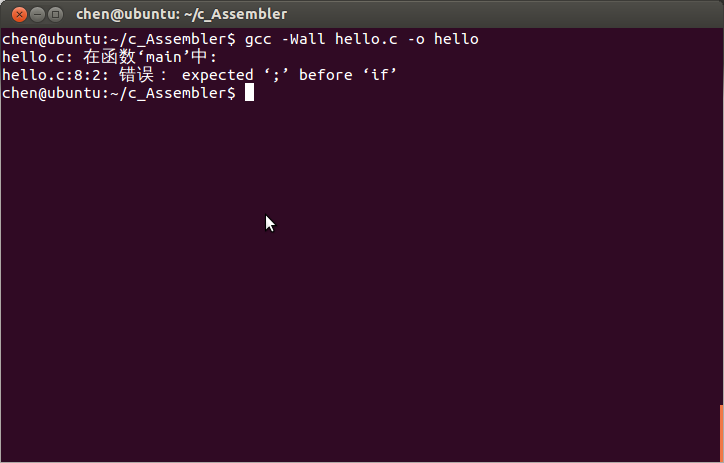
修改后的输出：

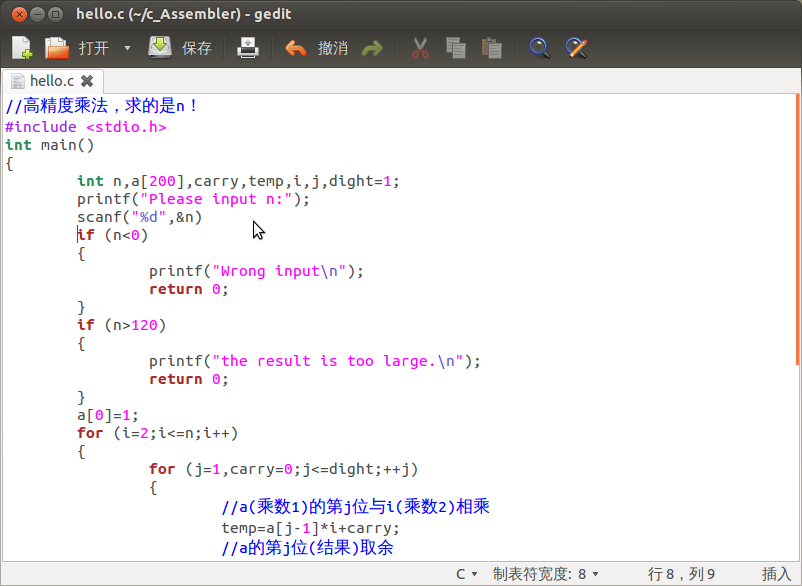


**实验过程中出现的问题及解决方法：**

程序出现语法错误

如：





要结合编译错误信息和程序找出错误之处