

# 实验 3 GCC 编译器和 GDB 调试器的使用

## 实验目的

- 掌握 GCC 编译器的使用方法
- 掌握 GDB 调试器的使用方法。

## 实验设备

- 硬件：PC 机
- 软件：WINXP+VMWare 虚拟机+UBUNTU1004-32bit+GCC。

## 实验内容

下表是本实验所涉及到的源文件及其中的函数

源文件名	函数名	描述
main.c	int main()	主函数
factorial.c	int factorial(int n)	计算 n 的阶乘
add.c	int add(int a,int b)	计算 a 和 b 的和
myprn.c	int prn(int x)	输出 x 的值

在主程序中 main.c 中输入一个整数 n，然后调用 factorial(int n)计算  $1! + \dots + n!$  的和，再调用 add(int a,int b)计算  $n!+(n-1)!$ ，最后调用 prn(int x)，输出上述结果。

## 实验原理

参考第 5 章相关内容

## 实验步骤

1. 在虚拟机 LINUX 下，编译源程序，包括如下源文件：main.c、factorial.c、add.c、myprn.c
2. 使用 gcc，编译生成可执行程序，可执行程序名字为 main；
3. 使用 gdb 调试 main 程序，要求如下：
  - a) 在 main 函数的“myfac=factorial(i);”语句位置处插入一个断点；
  - b) 在 main 函数的 sum\_add=add(factorial(n),factorial(n-1)); 语句位置处插入一个断点；
  - c) 执行程序，输入 n=3，当程序运行到该断点处，单步进入 factorial()函数内部；
  - d) 在 factorial()函数内部单步运行程序，每运行一步，查看变量 fac 的值，并记录；
  - e) 当程序运行到第 2 个断点处，单步进入 add()函数内部，
  - f) 在 add 函数内部单步运行，查看变量 sum 的值，并记录
  - g) 查看断点的情况，并记录

## 参考源程序

### 1. main.c

```
#include <stdio.h>

int main()
{
    int n=0;
    int sum_fac=0;
    int sum_add=0;
    int myfac=0;
    int i;
    printf("enter n,please:");
    scanf("%d",&n);

    for(i=1;i<=n;i++)
    {
        myfac=factorial(i);
        sum_fac=sum_fac+myfac;
    }

    sum_add=add(factorial(n),factorial(n-1));

    printf("1!+2!+...+n!=");
    prn(sum_fac);

    printf("(n-1)!+n!=");
    prn(sum_add);

    return 0;
}
```

### 2.factorial.c

```
int factorial(int n)
{
    int fac=1;
    int i;

    for(i=1;i<=n;i++)
    {
        fac=fac*i;
    }
}
```

```
        return fac;
    }
```

3.add.c

```
int add(int a,int b)
{
    int sum=0;
    sum=a+b;
    return sum;
}
```

4.myprn.c

```
#include <stdio.h>
```

```
int prn(int x)
{
    printf("%d\n",x);

    return 0;
}
```