# 计算机组成与设计实验任务说明

吴强，彭蔓蔓

2020 年 2 月 4 日

# 目录

# 实验 1　测量程序运行时间

## 1.1　实验目标

1. 学习使用 clock() 函数记录程序执行的 CPU 时间;
2. 学习使用 gettimeofday() 函数记录实际时间。

## 1.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 1.3　实验准备

1. 安装 Linux 操作系统，或 Windows 操作系统;
2. 如果 Linux 或 Windows 没有自带程序开发工具，那么需要安装 GCC（Linux）或 DevC++（Windows）;
3. 利用图书馆或互联网资源，了解标准 C/C++ 运行时库的 clock() 和 gettimeofday() 函数用法（例如: http://rabbit.eng.miami.edu/info/functions/time.html ）。

## 1.4　实验内容

1. 编写一个矩阵相乘的 C 程序，矩阵大小 n 根据实验机器性能自行设置，建议为 2000。相乘部分的参考代码如下所示:

```
for (k=0; k<n; k++) {
    for (i=0; i<n; i++) {
        r = a[i][k];
        for (j=0; j<n; j++) {
```

```
                    c[i][j] += r * b[k][j];
                }
            }
        }
```

编译运行后，检查程序运行是否正确;

2. 用 clock() 和 gettimeofday() 函数获取矩阵相乘程序运行时间（包括 CPU 时间和实际时间），建议重复 30 次并计算平均值作为结果。

## 1.5 思考问题

1. 程序运行的 CPU 时间是什么意思? CPU 时间和实际时间有什么联系和区别;
2. clock() 函数获取的 CPU 时间精度是多少?
3. gettimeofday() 函数获取的实际时间是什么含义，精度是多少?

## 1.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目;
2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，内核版本号，GCC 版本号;（2）矩阵相乘程序源代码，GCC 编译命令文本;（3）clock() 和 gettimeofday() 函数的使用要有程序源代码，实验中设置的具体矩阵大小，重复次数，每次测试的输出结果，计算得到的平均 CPU 时间，平均实际时间。
3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 2　整数的表示

## 2.1　实验目标

1. 学习使用定长整数类型;
2. 学习使用位操作对数据位进行操作。

## 2.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 2.3　实验准备

1. 利用图书馆或互联网资源, 了解标准 C/C++ 头文件 stdint.h 或 cstdint 中定义的 int8_t, int16_t, int32_t, int64_t, uint8_t, uint16_t, uint32_t, uint64_t 等数据类型，以及 (U)INT8(16,32,64)_MIN(MAX) 等预定义常量（例如: http://www.cplusplus.com/reference/cstdint/ ）。
2. 利用图书馆或互联网资源, 了解位操作用法（例如: https://en.wikipedia.org/wiki/Bitwise_operations_in_C ）。

## 2.4　实验内容

1. 编写一个程序，利用位操作检查一个整型数据的各个数据位并输出，或者说，输出一个整型数的二进制表示;
2. 用 8 位，16 位，32 位，64 位的无符号和有符号整型数据 0, ±1(无符号数忽略 -1), 最大值, 最小值测试上述程序，检验结果的正确性。

## 2.5 思考问题

1. C/C++ 语言中无符号整型数的编码是什么？有符号整型数的编码是什么?

2. C/C++ 语言中按位与，按位或，按位非，按位异或对有符号整型数的符号位是如何处理的?

3. C/C++ 语言中移位操作对有符号整型数的符号位是如何处理的?

## 2.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目；

2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；（2）整型数二进制输出程序源代码，GCC 编译命令文本；（3）用 8 位，16 位，32 位，64 位的无符号和有符号整型数据 0, ±1(无符号数忽略 -1), 最大值, 最小值测试上述程序，将输出的二进制与手工计算结果相比较，检验结果的正确性。

3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 3　指令的表示

## 3.1　实验目标

1. 学习了解 C/C++ 语言程序控制结构对应的指令序列;
2. 学习调试器的使用和查看可执行程序的汇编代码。

## 3.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 3.3　实验准备

1. 回顾教材第二章中关于 C/C++ 语言中条件分支和循环结构对应的指令序列;
2. 利用图书馆或互联网资源, 了解调试器用法（例如: https://www.gdb-tutorial.net/ ）。

## 3.4　实验内容

1. 编写一个程序，包含下述代码:

```
int a;

while(1) {

    printf("\n输入一个整数(负数将退出程序):");
    scanf("%d", &a);

    if(a < 0) break;
```

```
}
```

编译并改正可能的错误，使得程序运行正确;

2. 利用调试器查看程序的汇编代码，找到对应 while 循环语句和 if 条件分支语句的汇编代码段，并做记录。

## 3.5 思考问题

1. C/C++ 语言中 while 循环语句通常被编译成什么样的指令序列?
2. C/C++ 语言中 if 条件分支语句通常被编译成什么样的指令序列?
3. 实验中的无限 while 循环和其中的 if 分支语句被编译成了哪些指令?

## 3.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目；
2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；（2）程序源代码，GCC 编译命令文本, 利用调试器查看汇编代码的命令或截图；（3）查看到的程序汇编代码截图。
3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 4　用加减运算实现二进制整数除法

## 4.1　实验目标

1. 学习了解二进制整数除法的运算方法;
2. 学习了解二进制整数除法中对有符号数的处理方法。

## 4.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 4.3　实验准备

1. 回顾教材第三章关于二进制整数除法运算的过程;
2. 利用图书馆或互联网资源, 了解二进制整数除法的运算方法 (例如: https://en.wikipedia.org/wiki/Division_algorithm)。

## 4.4　实验内容

1. 从实验说明文档包中找到 div.cpp 程序，编译并运行;
2. 观察 div.cpp 程序的运行错误，改正其错误并再次编译运行;
3. 测试除数、被除数分别是正数、负数和零的情况，记录测试结果（截图）。

## 4.5　思考问题

1. div.cpp 采用的二进制整数除法是哪一种?
2. div.cpp 的错误在什么地方，如何改正?

3. div.cpp 中对商和余数的符号是如何处理的?

## 4.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目；

2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；（2）改正后的程序源代码，以及 GCC 编译命令文本；（3）测试除数、被除数分别是正数、负数和零的情况，记录测试结果 (请截图)。

3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 5　浮点数加法次序的问题

## 5.1　实验目标

1. 学习了解浮点数运算次序引起的问题;
2. 学习了解浮点数运算产生误差的原因。

## 5.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 5.3　实验准备

1. 回顾教材第三章关于浮点数表示以及浮点数运算尤其是加减法运算的过程;
2. 利用图书馆或互联网资源, 了解浮点运算产生误差的原因和需要注意的问题
   (例如: https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html
   以及 https://en.wikipedia.org/wiki/Floating-point_arithmetic#
   Causes_of_Floating_Point_Error )。

## 5.4　实验内容

1. 从实验说明文档包中找到 floatsum.cpp 程序，编译并运行;
2. 测试 3 个不同种子数 (1, 3, 5) 和 3 个不同的数组大小 (30, 40, 50) 的组合，观察数组累加结果并截图记录, 以便后续分析。

## 5.5 思考问题

1. 浮点数运算产生误差的根本原因是什么?
2. 为什么浮点加法运算的次序会影响误差的大小?
3. 怎么样减小浮点运算的误差?

## 5.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目；
2. 实验记录需要有如下信息：(1) 实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；(2) 程序源代码，以及 GCC 编译命令文本；(3) 测试 3 个不同种子数 (1, 3, 5) 和 3 个不同的数组大小 (30, 40, 50) 组合时程序的输出 (请截图)。
3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 6　栈与过程调用

本实验中使用的 stacktest.cpp 程序是受到通信 1603 班崔威恒同学和通信 1604 班庄义昱同学在讨论课上所演示程序的启发而编写，特此鸣谢！

## 6.1　实验目标

1. 学习掌握栈在过程调用中的作用;
2. 学习了解栈溢出的原理。

## 6.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 6.3　实验准备

1. 回顾教材第二章关于过程调用的内容，了解栈在过程调用中保存返回地址，传递参数以及为局部变量提供存储空间等作用;
2. 利用图书馆或互联网资源, 了解栈溢出的原因及影响(例如: https://en.wikipedia.org/wiki/Stack_buffer_overflow )。

## 6.4　实验内容

1. 从实验说明文档包中找到 stacktest.cpp 程序，编译并运行;
2. 测试程序输入数从 0 到 5 的执行情况并截图记录, 以便后续分析。

## 6.5 思考问题

1. C/C++ 语言的过程调用在利用栈传递参数时各个参数的入栈次序是怎么样的？反映到参数地址上的表现是什么？

2. C/C++ 语言的过程调用在利用栈保存返回地址时，返回地址的保存位置与参数和局部变量保存位置的关系是怎么样的？

3. 利用数组越界修改返回地址时可能有哪些情况？修改成功时程序执行情况会是什么样的？不成功时又会是什么样的？

## 6.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目；

2. 实验记录需要有如下信息：(1) 实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；(2) 程序源代码，以及 GCC 编译命令文本；(3) 输入数从 0 到 5 时程序的输出 (请截图)。

3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 7　矩阵相乘循环次序对执行时间的影响

## 7.1　实验目标

1. 了解高速缓存结构和工作原理;
2. 学习了解访问次序对高速缓存命中率的影响。

## 7.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 7.3　实验准备

1. 回顾教材第五章中关于高速缓存结构和工作原理的知识;
2. 利用图书馆或互联网资源，了解 C/C++ 数组的内存放置（例如: http://c.biancheng.net/cpp/html/2920.html ）。

## 7.4　实验内容

1. 编写一个矩阵相乘（建议矩阵大小 1000x1000）的程序，测试不同循环次序对执行时间的影响。其中循环次序如下所示:

```
// ijk
for(i=0; i<n; i++) {
    for(j=0; j<n; j++) {
        s = 0;
        for(k=0; k<n; k++) {
            s += a[i][k]*b[k][j];
```

```
            }
            c[i][j] = s;
        }
    }

    // ...

    // jik
    for(j=0; j<n; j++) {
        for(i=0; i<n; i++) {
            s = 0;
            for(k=0; k<n; k++) {
                s += a[i][k]*b[k][j];
            }
            c[i][j] = s;
        }
    }

    // ...

    // kij
    for(k=0; k<n; k++) {
        for(i=0; i<n; i++) {
            r = a[i][k];
            for(j=0; j<n; j++) {
                c[i][j] += r*b[k][j];
            }
        }
    }

    // ...
```

```
// ikj
for(i=0; i<n; i++) {
    for(k=0; k<n; k++) {
        r = a[i][k];
        for(j=0; j<n; j++) {
            c[i][j] += r*b[k][j];
        }
    }
}

// ...

// jki
for(j=0; j<n; j++) {
    for(k=0; k<n; k++) {
        r = b[k][j];
        for(i=0; i<n; i++) {
            c[i][j] += a[i][k]*r;
        }
    }
}

// ...

// kji
for(k=0; k<n; k++) {
    for(j=0; j<n; j++) {
        r = b[k][j];
        for(i=0; i<n; i++) {
            c[i][j] += a[i][k]*r;
```

```
            }
        }
    }

    // ...
```

编译并改正可能的错误，使得程序运行正确;

2. 运行程序并记录不同次序循环执行时间（建议每种循环针对不同大小(n) 的矩阵反复连续执行多次，取第二次以后的执行时间计算平均时间，从而减小“预热”时间的影响)。

## 7.5 思考问题

1. 矩阵在内存中是怎么存放的?
2. 不同循环次序计算矩阵相乘的内存访问顺序有什么区别?
3. 不同循环次序导致执行时间差异的原因是什么?

## 7.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目;
2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；（2）程序源代码，GCC 编译命令文本；（3）不同循环次序计算矩阵相乘的平均时间。
3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。

# 实验 8　CPU 功能模拟程序设计

## 8.1　实验目标

1. 学习掌握 CPU 的功能和指令执行过程;
2. 学习了解模拟程序设计方法。

## 8.2　实验条件

1. Linux 操作系统（推荐），或 Windows 操作系统;
2. 程序开发工具 GCC（Linux），或 DevC++（Windows）。

## 8.3　实验准备

1. 回顾教材第四章关于 CPU 的内容，了解 CPU 执行指令的过程;
2. 利用图书馆或互联网资源, 了解模拟程序设计(例如: https://fms.komkon.org/EMUL8/HOWTO.html )。

## 8.4　实验内容

1. 从实验说明文档包中找到 cpu_todo.cpp 程序，了解其中思路，补全其中欠缺部分;
2. 编译运行修改好的程序，观察程序运行结果。如果运行结果有错误，请改正。

## 8.5　思考问题

1. CPU 执行指令的基本步骤是哪几步？分别对应于模拟程序中哪些部分?

2. 模拟程序采用了无限循环 while(1) 来表示 CPU 反复执行指令的过程，那么模拟程序退出循环的条件是什么?

3. 程序计数器（PC）在模拟程序中是怎么更新的? 请说明 beq 指令对 PC 更新的过程;

## 8.6 报告要求

1. 实验报告需要有姓名，学号，班级，实验名称，实验目标，实验条件，实验内容，实验记录，实验分析等项目;

2. 实验记录需要有如下信息：（1）实验机器 CPU 配置，内存配置，操作系统名称和版本号，GCC 或 DevC++ 版本号；（2）补全后模拟程序源代码，以及 GCC 编译命令文本；（3）模拟程序运行结果（如果结果难以截图，请文字描述）。

3. 实验分析要根据实验记录得到的结果，回答思考问题中各个问题。