

# 科大讯飞 内核研发大课堂

THERMOSPHERE  
MESOSPHERE

MESOSPHERE  
STRATOSPHERE

STRATOSPHERE  
TROPOPAUSE

TROPOSPHERE

GROUND

SPACE

## 2016新员工入职培训

PTL-10000

**The Troposphere**  
The troposphere is the lowest layer of the atmosphere. It begins at the surface and extends to between 7 km (22,000 ft) at the poles and 17 km (56,000 ft) at the equator, with some variation due to weather factors. The troposphere has a great deal of vertical mixing because of convection at the base. This mixing makes air masses less dense as they rise. This is the area of A-10's base system.

**The Stratopause**  
The boundary between the stratosphere and troposphere.

**The Stratopause**  
The boundary between the mesosphere and the stratosphere, typically 50 to 55 km (31 to 34 mi; 160,000 to 180,000 ft). The pressure here is 1/1000th sea level.

**The Stratosphere**  
From the Latin word "stratus" meaning spreading out. The stratosphere extends from the tropopause's 7-17 km (4.3-11 mi; 23,850-56,000 ft) range to about 51 km (32 mi; 170,000 ft). Temperature increases with height. The stratosphere contains the ozone layer, the part of the Earth's atmosphere which contains relatively high concentrations of ozone. "Stratos" high means at low levels per million, much higher than the concentrations at the lower atmosphere but still lower compared to the main components of the atmosphere. It is mostly located in the lower portion of the stratosphere from approximately 15-35 km (9.3-22 mi; 49,000-115,000 ft) above Earth's surface, though the thickness varies seasonally and geographically. This is where the weather balloons float around.

**The Ozone Layer**  
Though part of the stratosphere, the ozone layer is considered as a layer of the Earth's atmosphere in itself because its physical and chemical composition is far different from the stratosphere. Ozone (O<sub>3</sub>) in the Earth's stratosphere is created by ultraviolet light striking oxygen molecules (consisting two oxygen atoms, O<sub>2</sub>), splitting them into individual oxygen atoms (atomic oxygen), the atomic oxygen then combines with unbroken O<sub>2</sub> to create O<sub>3</sub>. O<sub>3</sub> is unstable, so the stratosphere, long-lived and when it is broken down by light into more 2 atoms into a molecule of O<sub>2</sub> and an atom of atomic oxygen, a continuing process called the ozone-oxygen cycle. This occurs in the ozone layer, the region from about 15 to 48 km (23,000 to 150,000 ft) above Earth's surface. About 90% of the ozone in the atmosphere is contained in the stratosphere. Ozone concentrations are highest between about 20 and 40 km (65,000 and 130,000 ft), where they range from about 2 to 8 parts per million.

# C/C++编码规范

规范不仅是为了美观

雷琴辉

---

2016.08.08



细节规范

常见错误

不良习惯

方法原则

# 内容大纲

- 1 课程概述
- 2 排版命名
- 3 函数过程
- 4 常量和宏
- 5 C/C++标准
- 6 工程实现



# 何为高质量的 C/C++代码



- 鲁棒 - Solid and Robust Code
- 简洁 - Maintainable and Simple Code
- 高效 - Fast Code
- 简短 - Small Code
- 共享 - Re-usable Code
- 可测试 - Testable Code
- 可移植 - Portable Code



- **编码风格 (Coding Style)**

- 不同的风格很难抽象地界定它的好坏 ?
- 但在具体的环境中，可以界定是否恰当
- 一些具有很多缺点的风格还是要避免的
- 工程化 / freedom ?
- 个人编码风格须保持一致。

- **“嫁鸡随鸡” 原则**

- 如果必须依赖一个很大的库，不妨考虑使用它的风格



# 概述-总体要求

- Your code may outlive you or your memory!
- Think about the maintainer
- Comment your code
  - File Header
  - Function Header
  - Inline Commenting
- Pick a good name for your functions and variables
- Align/Indent your code!
- Less arguments

# 概述-提升的途径

- **集百家之长，归我所用**
  - 多读一些优秀的代码库
  - 内容优先，形式次之
- **适量的翻一翻C++的“武功秘籍”**
  - C/C++基本的语法要熟悉，练好“内功”
  - 几个C++大师的书还是需要读一读
  - Code Review/Inspection Check List





# 排版-“金玉其表”

- A program is a sort of publication.
  - 机器只管你的代码是否正确，没有美观性要求
  - So，代码的美观性带有主观性
  - 原则上，代码的排版格式尽量向English的出版物格式靠近。

*Abstract*—FFTW is an implementation of the discrete Fourier transform (DFT) that adapts to the hardware in order to maximize performance. This paper shows that such an approach can yield an implementation that is competitive with hand-optimized libraries, and describes the software structure that makes our current FFTW3 version flexible and adaptive. We further discuss a new algorithm for real-data DFTs of prime size, a new way of implementing DFTs by means of machine-specific “SIMD” instructions, and how a special-purpose compiler can derive optimized implementations of the discrete cosine and sine transforms automatically from a DFT algorithm.

*Index Terms*—FFT, adaptive software, Fourier transform, cosine transform, Hartley transform, I/O tensor.

## I. INTRODUCTION

**F**FTW [1] is a widely used free-software library that computes the discrete Fourier transform (DFT) and its

of a multi-dimensional array. (Most implementations support only a single DFT of contiguous data.)

- FFTW supports DFTs of real data, as well as of real symmetric/antisymmetric data (also called discrete cosine/sine transforms).

The interaction of the user with FFTW occurs in two stages: planning, in which FFTW adapts to the hardware, and execution, in which FFTW performs useful work for the user. To compute a DFT, the user first invokes the FFTW planner, specifying the problem to be solved. The problem is a data structure that describes the “shape” of the input data—array sizes and memory layouts—but does not contain the data itself. In return, the planner yields a *plan*, an executable data structure that accepts the input data and computes the desired DFT. Afterwards, the user can execute the plan as many times as desired.

The FFTW planner works by measuring the actual run time

# 排版-Issues

- **Tab问题（空格 / Tab）**

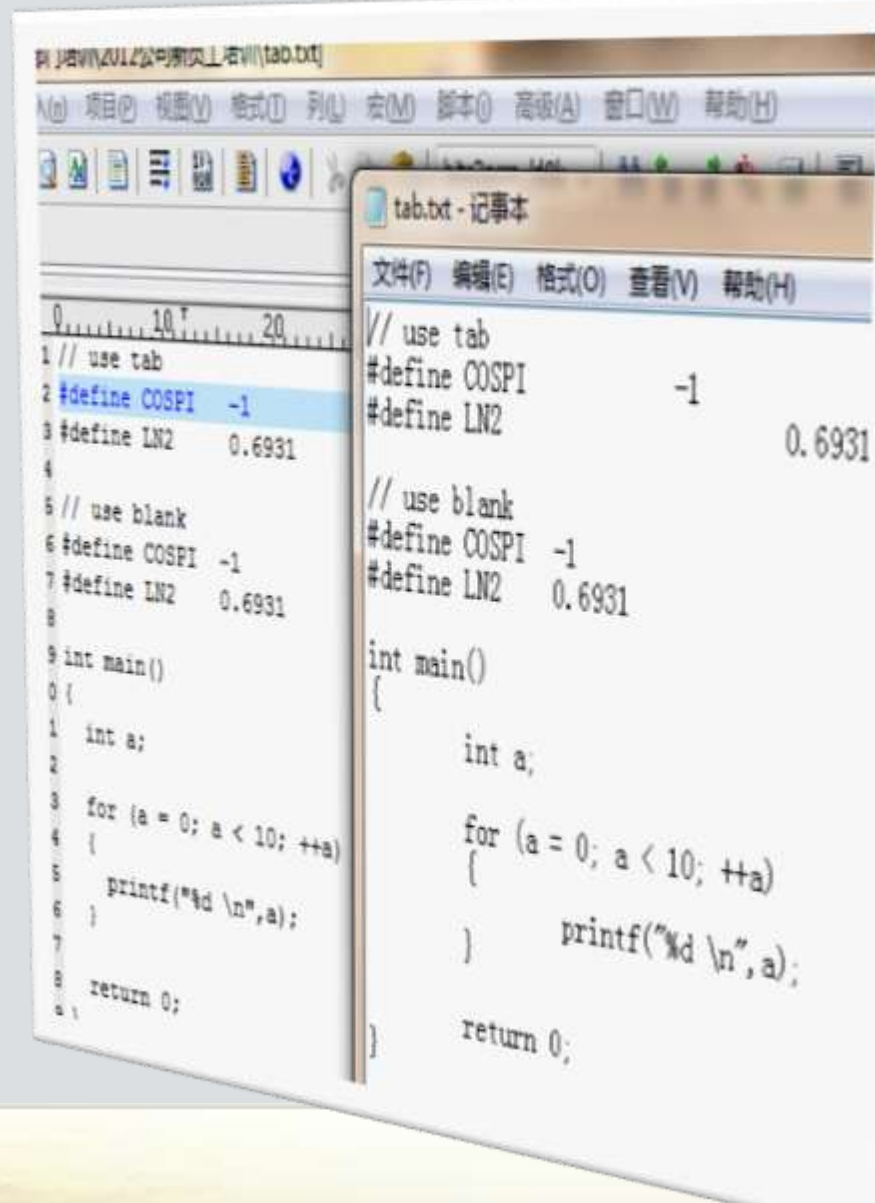
- 建议发布给第三方的头文件使用空格
- 内部使用4位缩进的Tab

- **长语句的切分**

- 到底多长才算长？
- 长表达式要在低优先级操作符处划分新行

- **何处留空？**

- 高优先级的操作符不留空，低的留空



# 排版-烦人 “{ }”

- 简单的符号引起C/C++语言上风格最大的争议
- 某些高级语言干脆就取消了 “{ }”
- 我们的做法
  - 虽然看多了不好看，但在if/for/while/switch等表达式后面的不可省
  - 不能将{ 与缩进的代码写在一行
  - 是否将{或}单独起行，不作限制

```
for (...)  
{  
    if (...)  
    {  
        Program Code  
    }  
}
```


```
for (...) {  
    if (...) {  
        Program Code  
    }  
}
```

# 注释-基本要求

- 一定要保持注释与代码更新，不再有用的注释要删除。
- 适当的文字图是很受欢迎的
  - 下面是一个很好的示例
- 避免在注释中使用缩写，特别是非常用缩写
- 如果能用英文注释最好，很多编译器不支持中文



CPCovlib.h



cssmLh

- 传统的注释风格

```
////////////////////////////////////  
/// File.h – describe this header file  
.....
```

- 文档化的风格

- 对于注释格式作出一定的要求
- 在注释中使用一些特定的标记
- Javadoc / QT

# 注释-文档效果

## 详细描述

简单说明

detail... ( 详细说明 )

## 作者:

yigao

## 版本:

1.0

## 日期:

2004年4月15日

## 参见:

## 版本记录 :

版本	日期	作者	备注
<b>1.0</b>	<b>2004年4月15日</b>	<b>yigao</b>	创建



# 取个好名字 - Use Naming Conventions

- Windows平台下，特别是与基于MFC的代码，建议使用Hungarian Notation。
- 一些需要或未来需要跨平台的代码，建议使用Unix-Like的风格命名方式
  - 使用小写字母 + 下划线的组合。
  - 模板代码不需要支持类型前缀
- 风格需要始终一致
  - add\_user、AddUser、m\_AddUser      OK
  - ~~Add\_User~~      @~#!\$#&\*%

# Hungarian Notation

匈牙利命名法，由查尔斯·西蒙尼发明。这种命名法其实是对于西蒙尼祖籍的一种讽刺。匈牙利人名和大多数其他欧洲人名相比是反过来的，即姓氏在名字的前面。举个例子，英语化的名字“Charles Simonyi”在匈牙利语中原本是“Simonyi Károly”。同样的，在匈牙利命名法中，类型名在实际变量名前，而不是像大多数欧洲的Smalltalk那样，类型放在变量名后，例如aPoint和lastPoint。

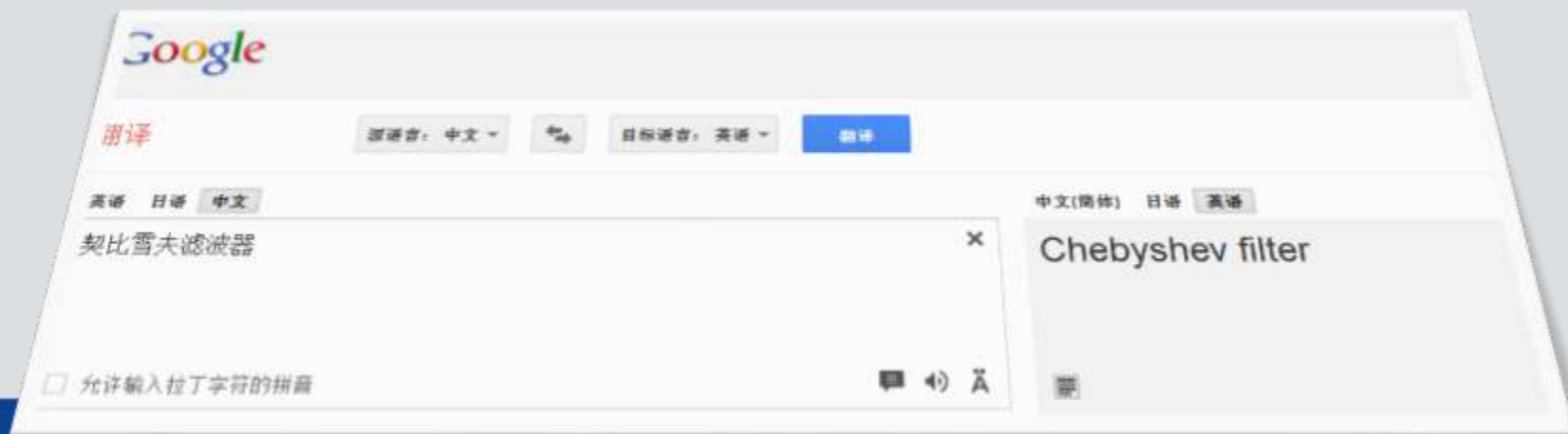


到了2002年，西蒙尼离开微软，与生意伙伴Gregor Kiczales开创“Intentional Software”公司去继续他们的未圆之梦。

到了2005年，西蒙尼被福布斯杂志列入了当年十亿美元超级富翁的名单之内并保持至今，在成名与获利之际西蒙尼热衷于慈善事业。每年捐助一千万美元至五千万美元给各项科学基金、公益事业。如在二〇〇四年他把荣获的年度信息系统奖项（Annual Infosys Award）全数捐给了慈善机构。

# 如何起一个好的名字

- 非English母语的程序员如何起英文名？
  - 不要降低对于命名的要求。
    - Your program is a sort of publication.
  - 根本解决之道，提高英文水平
  - 查阅相关的领域的英文文献，专业名词
  - 借鉴其它类似的代码或库的命名和词语
  - Google翻译之



- **What's the maximum length of a function?**
  - Study in 1986 on IBM's OS/360: the most error-prone routines => longer than 500 lines.
  - Study in 1991 on 148,000-line of code: functions < 143 lines => 2.4 times less expensive to fix than longer functions.
  - Watch out for functions > 200 lines!
- **原则上函数名多数可以使用动宾结构来命名**

# 尽量使用const

- **const能够在语意的层次来约束你的代码或是接口，使用你无法在这一点犯错（编译器不答应）**
- **const char\* p/char const \* p/char\*  
const p**
- **const可以比#define更好地完成常量定义的工作**

- **在C语言中，解决下面问题好方法**
  - 代码简洁
  - 一致性
- **除了条件编译，否则不需要宏了。**
  - 使用const和inline可以完全替掉
  - 不易理解
  - 不利于调试（无法生成调试符号）



- **ANSI C**

- C 的第一个标准是由ANSI发布的。虽然这份文档后来被[国际标准化组织](#)(ISO)采纳并且ISO发布的修订版也被ANSI采纳了，但名称ANSI C(而不是 ISO C)仍被广泛使用。**ISO C/Standard C**。
- C89/C90/C99/C11
- ANSI C几乎被所有广泛使用的编译器支持

- **ISO C++**

- [ANSI-ISO](#)委员会于[1998年](#)对C++标准化 ( [ISO/IEC 14882 : 1998](#) )
- [2005年](#)，一份名为[Library Technical Report 1](#) ( 简称TR1 ) 的技术报告释出，几乎所有目前仍在维护的C++编译器皆已支援TR1。
- 最新的C++标准是2011年9月发布的[ISO/IEC 14882:2011<sup>\[1\]</sup>](#)，又称[C++0x](#)或[C++11](#)

©ISO/IEC

ISO/IEC 9899:1999 (E)

## Contents

Foreword . . . . .	xi
Introduction . . . . .	xiv
1. Scope . . . . .	1
2. Normative references . . . . .	2
3. Terms, definitions, and symbols . . . . .	3
4. Conformance . . . . .	7
5. Environment . . . . .	9
5.1 Conceptual models . . . . .	9
5.1.1 Translation environment . . . . .	9
5.1.2 Execution environments . . . . .	11

# INTERNATIONAL STANDARD

ISO/IEC  
14882

Second edition  
2003-10-15

---

---

## Programming languages — C++

*Langages de programmation — C++*

# C/C++ (标准) 库

- 在C++中，库的地位是非常高的。
  - C++之父 Bjarne Stroustrup先生多次表示了设计库来扩充功能要好过设计更多的语法的言论。
- C runtime-library
- iostream-library
- STL(standard template library)
  - “用过都说好”。
- 准标准库—Boost
  - 提供包括正则表达式、图组件和算法、内存池管理、把C++类和函数映射到Python之中等功能支持。
  - 提供工具支持对库进行剪裁。
- ACE — Adaptive Communication Environment
- QT/GTK/GTK+等

- 注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级。
- 不要使用难懂的技巧性很高的语句，除非很有必要时

\* stat\_poi ++ += 1; // 除非这是C语言考题

- 不要画蛇添足，对语言进行重定义

```
Status = f( ..., &ptr );
```

```
if( Status isnot NO_ERROR or ptr is NULL )  
    goto cleanup;
```

**is??? isnot??? or???**

- **不要忘记最早测试代码的人是你自己，编程时要考虑到单元测试和联合调试的需要，设定一些调试信息的输出和单元测试点**
  - 使用断言来发现软件问题，提高代码可测性
  - 对于复杂的断言提供注释
  - 关键要树立这方面的意识
  - 复杂的系统Log必不可少

- 与操作系统的接口
- 使用可移植的底层库
- 字节序
  - bitfield你真的需要吗？
- 结构体对齐
  - 复杂度超过你的想象



- **编程时要经常注意代码的效率。**
- **在保证软件系统的正确性、稳定性、可读性及可测性的前提下，提高代码效率。**
  - 说明：不能一味地追求代码效率，而对软件的正确性、稳定性、可读性及可测性造成影响。
- **效率优化首先着眼于全局，再到局部。**

# 工程组织-基本形式

- 目标：构建与一个目标运行环境基本一致的目录结构
  - bin/
  - data/
  - include/ <--开放给第三方的开发接口文件
  - source/
    - XXXX.sln
    - Makefile
    - module\_a/
    - .....
  - .....

- **与目标环境一致**
  - 配置一致，易于调试和单元测试
  - 组织结构的不合理性能够较早的暴露
- **考虑到配置管理的需求**
  - 对于内容较多的配置文件建议也纳入到版本管理系统中
  - Debug / Release版本的输出可以xxxd.dll  
xxxd.exe这个方式来区别，而不是靠目录结构来区分。

# 没有规矩 不成方圆



平嵌研发中心  
13866109785  
[qhlei@iflytek.com](mailto:qhlei@iflytek.com)