

2018-2019年度第二学期 00106501

# 计算机图形学



童伟华 管理科研楼1205室

E-mail: [tongwh@ustc.edu.cn](mailto:tongwh@ustc.edu.cn)

中国科学技术大学 数学科学学院

<http://math.ustc.edu.cn/>



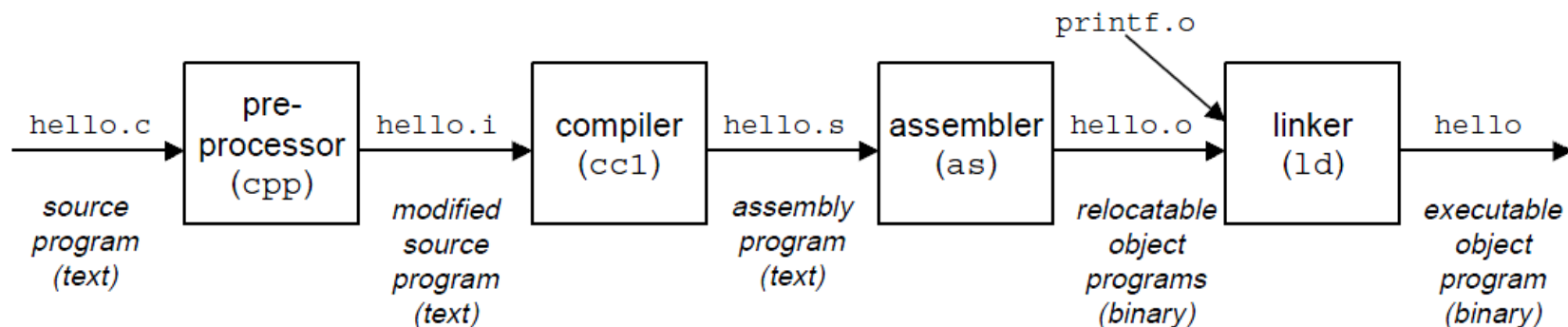


# 附讲一 编程简介

## ■ 什么是编程？

- From Wikipedia : **Computer programming** (often shortened to **programming** or **coding**) is the process of writing, testing, debugging/troubleshooting, and maintaining the [source code](#) of [computer programs](#). This source code is written in a [programming language](#). The code may be a modification of an existing source or something completely new. The purpose of programming is to create a program that exhibits a certain desired behaviour (customization). The process of writing source code often requires expertise in many different subjects, including knowledge of the application domain, specialized [algorithms](#) and [formal logic](#).
- 编程：编写，测试，调试，维护
- 核心：应用领域知识 + 算法 + 数据结构

## ■ 代码编译



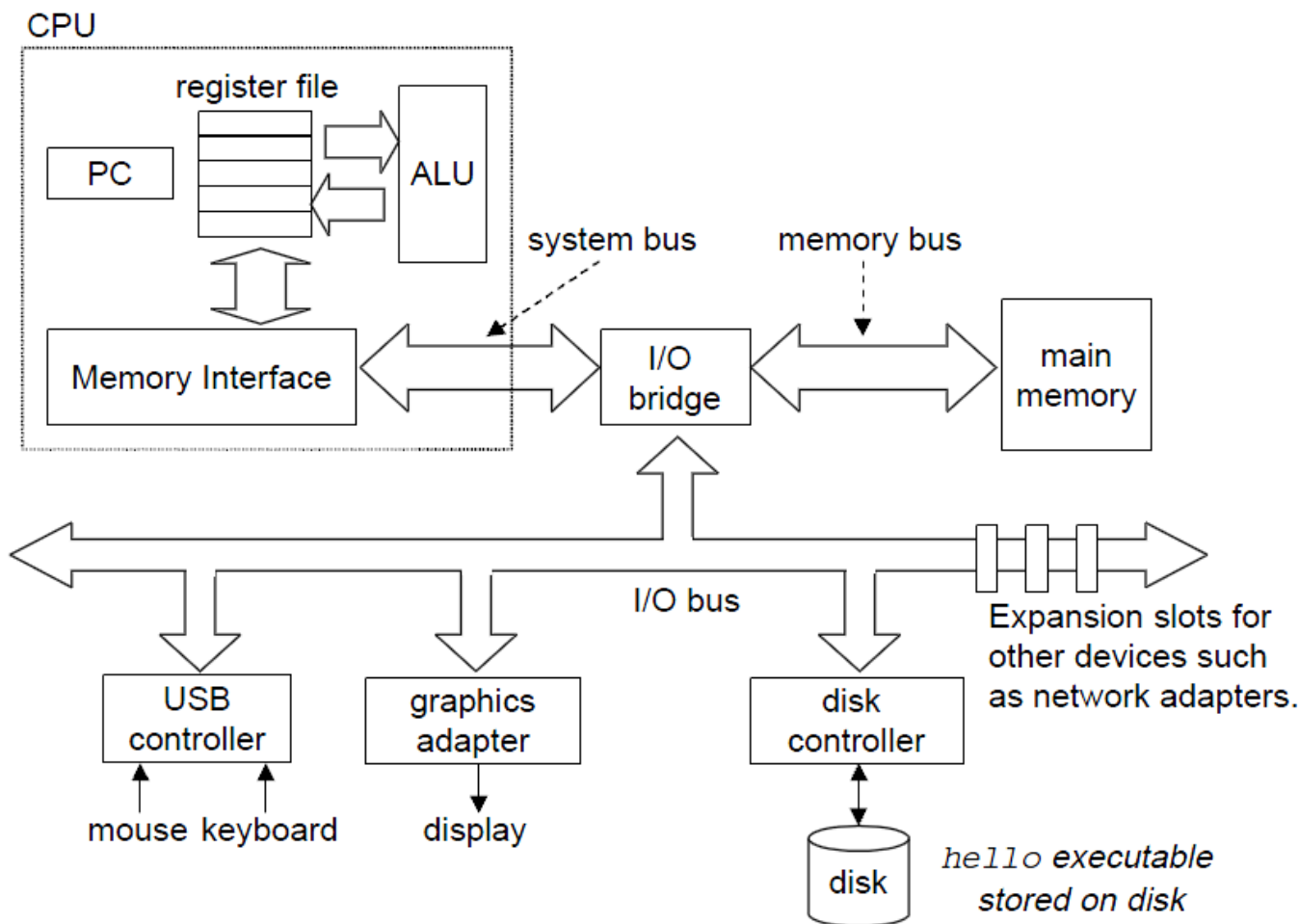
## ■ 为什么需要了解编译系统

- 代码优化：例如switch语句是否总是比if-then-else有效？函数调用的代价是多高？While循环是否总比do循环有效？等等问题，涉及编译器，机器语言
- 理解链接错误信息：link-time errors，例如静态库与动态库的区别？库的次序对链接的影响？
- 避免安全漏洞：例如缓冲区溢出，堆、栈的管理，参数传递及返回等

# 编程模型



## 程序的执行



## ■ 什么是编程语言？

- From Wikipedia : A programming language is an artificial [language](#) designed to express [computations](#) that can be performed by a [machine](#), particularly a [computer](#). Programming languages can be used to create [programs](#) that control the behavior of a machine, to express [algorithms](#) precisely, or as a mode of human communication.

## ■ 编程语言分类

- 目前大概有2500多种编程语言
- 按演化情况（第一代、第二代...第五代）
- 按实际用途（系统编程语言、网络编程语言、数据库语言等等）
- 按语言层次（机器语言，汇编语言，高级语言）
- 按执行情况（编译型、解释型、虚拟机）

## ■ TIOBE世界编程语言排行榜 (2019年2月)

Feb 2019	Feb 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.876%	+0.89%
2	2		C	12.424%	+0.57%
3	4		Python	7.574%	+2.41%
4	3		C++	7.444%	+1.72%
5	6		Visual Basic .NET	7.095%	+3.02%
6	8		JavaScript	2.848%	-0.32%
7	5		C#	2.846%	-1.61%
8	7		PHP	2.271%	-1.15%
9	11		SQL	1.900%	-0.46%
10	20		Objective-C	1.447%	+0.32%
11	15		Assembly language	1.377%	-0.46%
12	19		MATLAB	1.196%	-0.03%
13	17		Perl	1.102%	-0.66%
14	9		Delphi/Object Pascal	1.066%	-1.52%
15	13		R	1.043%	-1.04%
16	10		Ruby	1.037%	-1.50%
17	12		Visual Basic	0.991%	-1.19%
18	18		Go	0.960%	-0.46%
19	49		Groovy	0.936%	+0.75%
20	16		Swift	0.918%	-0.

## ■ 系统编程语言

- C
- C++
- 汇编语言

## ■ 网络编程语言

- Java
- C#
- XML/AJAX /PHP/PERL/Javascript/
- Ruby
- Python

## ■ 数据库编程语言

- SQL (国际标准)

## ■ 人工智能编程语言

- Lisp
- Prolog



## ■ 脚本编程语言

- 通常为解释执行，有专门的解释器
- Unix/Linux shell: bsh/csh等
- 其他如PHP/PERL/RUBY/PYTHON等（通用动态语言）

## ■ 文本处理编程语言

- Tex/Latex等

## ■ 科学计算编程语言

- Fortran
- Matlab

## ■ 符号计算编程语言

- Maple
- Mathematica

## ■ 硬件部分

- CPU指令：Intel 和 AMD的X86指令集（CISC指令集），IBM Power系列的Power指令集（RISC指令集）等
- 系统架构：X86架构，IA和IA-64架构，Power架构等

## ■ 软件部分

- 编译器
- 链接器
- 调试器
- 系统调用API
- 基础类库：Windows下的MFC，跨平台的QT等
- 集成开发环境：Microsoft的Visual Studio系列，Linux下的kdevelop 或 GTK+GONME + glade + vim等

## ■ Microsoft的Visual Studio系列

- ...
- Visual Studio 2013
- Visual Studio 2015
- Visual Studio 2017

## ■ Visual Studio

- Workspace: 工程项目的容器
- Project: 包括源代码、资源文件、编译器选项、子工程等
- Project类型: 譬如Win32 Console Application, Win32 Application, Win32 Static Library, Win32 Dynamic Link Library等 (对于不同工程类型, visual studio生成不同的文件, 配置缺省的编译选项)

# 编程规范



- 编写程序一定要养成良好的代码习惯
- “程序不是写给自己看的，是写给别人看的”
- 高质量代码
  - 提高编程效率
  - 减少调试时间
  - 提高人品：代码是写给他人用的！
- 养成好习惯
  - 从点点滴滴做起
  - 不要光看不做

# 高质量代码的特性



- 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

# 关于代码风格问题



- 代码风格 (Coding Style) 是一种习惯
- 现在许多大公司都对员工书写代码制定了规范
- 开发大项目时由项目管理者制定代码规范
- 程序风格的重要构成因素
  - 程序版式
  - 命名规则
  - 函数设计原则
- 其他
  - 表达式规则
  - 与零比较
  - 常量规则
  - 动态数组
  - 内存管理

# 程序版式



## ■ 程序版式——程序员的书法

- 比书法好学得多，基本不需要特别练习

- 但是坏习惯一旦养成，就像书法一样难以改变

## ■ 不影响程序的功能，但影响程序的可读性

## ■ 追求

- 清晰、整洁、美观、一目了然

- 容易阅读，容易测试

## ■ 不良的风格

```
#include <stdio.h>
#include <math.h>
main()
{int i;
for (i=2;i<100;i++)
{if(isprime(i))
printf("%d\t",i); }
}int isprime(int n)
{int k,i;
if (n == 1) return 0;
k=sqrt((double)n);
for (i=2;i<=k;i++)
{if(n%i==0) return 0;}
return 1;
}
```



# 程序版式



## ■ 良好的风格

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

main()
{
    int i;

    for (i=2; i<100; i++)
    {
        if (isprime(i))
            printf("%d\t", i);
    }
}
```

```
int isprime(int n)
{
    int k, i;

    if (n == 1) return 0;

    k = (int)sqrt((double)n);

    for (i=2; i<=k; i++)
    {
        if (n % i == 0)
            return 0;
    }
    return 1;
}
```

## ■ 注释 (Comments) 的重要性

- 注释对于程序犹如眼睛对于人的重要性一样
- 没有注释的程序对于读者好比眼前一团漆黑，跟拿到一个可执行程序别无二致
- 不规范的注释和好几千度的近视眼没什么区别
  
- 代码本身体现不出价值
- 开发程序的思维才能使其变得有价值
- 这种思维的具体体现就是在于注释和规范的代码本身

## ■ 写注释给谁看？

- 给自己看，使自己的设计思路得以连贯
- 给继任者看，使其能够接替自己的工作

## ■ 在哪些地方写注释？怎样写注释？

- 要站在继任者的角度写
- 简单明了、准确易懂、防止二义性
- 让继任者可以轻松阅读、复用、修改自己的代码

## ■ 注释的风格

## ■ 写注释时的注意事项

# 注释规范



- 好的注释（尤其是算法注释）是对设计思想的精确表述和清晰展现，能揭示代码背后隐藏的重要信息（为便于移植，最好应英文写）

```
/*打开输入文件后判断文件长度是否符合格式要求*/
```

```
if ((fin = fopen("cat.pic","rb") == NULL)
```

```
{
```

```
    puts("打开文件cat.pic失败");
```

```
    return -1;
```

```
}
```

```
.....
```

```
/*
```

```
* 下面是图像转换的算法实现。彩色图像到灰度图像的转换主要利用RGB颜色空间到
```

```
* YUV颜色空间的变换公式来取得灰度值，公式为 $Y = 0.299 * R + 0.587 * G + 0.114 * B$ 
```

```
*/
```

```
for (i=0; i<400; i++)
```

```
    for (j=0; j<400; j++)
```

```
    {.....
```

```
        y = (299 * r + 587 * g + 114 * b) / 1000;
```

```
        .....
```

```
    }
```

```
.....
```

```
fclose(fin);
```

# 程序作业递交



- 作业递交是通过SmartChair投稿软件系统来完成，链接为：
  - [http://www.smartchair.cn/CG\\_2019](http://www.smartchair.cn/CG_2019)
- 递交内容：
  - 源代码 + 作业报告（纯粹的编程作业可省略）
- 所有文件压缩打包成一个文件(zip或rar)，文件命名规则：
  - 压缩包文件的命名方式：学号\_姓名\_作业编号.zip/rar
  - 例如：“PB16001001\_张三\_H01.zip”

# 作业要求



## ■ 代码编写

- 高质量代码
  - 符合一定的规范
  - 代码清晰
  - 尽量多的注释

## ■ 编程环境

- 操作系统：Windows 7及以上
- 集成开发环境：Visual Studio 2013 及以上
- 必须编译通过，且能够正常运行

# 作业要求



## ■ 作业报告（纯粹的编程作业可省略）

- 程序使用的简单说明，可截图说明
- 明确的输入和输出
- 测试结果和分析、小结等
- 若是合作项目，需说明具体分工
- 报告需规范
- 在报告中说明所参考的来源
  - 网站、书籍
  - 致谢他人

# 作业要求



- 通过SmartChair系统来递交作业，不接受Email递交作业
  - 严格按照规则命名
  - 必须在规定的最后期限之前递交
  - 超过最后期限系统将自动不再接受作业
- 独立完成或团队合作
- 绝不允许抄袭！





Thanks for your attention!

