

2019年高等院校计算机系统类课程联合研讨会



计算机类专业教学案例库的建设

-----编译教学案例的思考

张莉

北京航空航天大学

lily@buaa.edu.cn



- **案例教学法与教学案例**
- **计算机类专业教学案例的特点---编译课程案例**
- **2018年第一届软件工程教学案例比赛**
- **2019年以及未来的规划和思考**

- 什么是**案例教学法**
- 20世纪初，哈佛大学创造了案例教学法。即围绕一定培训的目的把**实际中真实的情景**加以**典型化处理**，形成供学生**思考分析和决断**的案例（通常为书面形式），通过**独立研究和相互讨论**的方式，来提高学生**分析问题和解决问题的能力**的一种方法。
- **案例教学法能够调动学生的学习主动性**，学生摄取的信息量可大大增加，同时，学生的分析、表达、团队合作等能力都能得到有效锻炼和提高。
- 案例教学的**精髓**不在于让学生去认同和理解某种既定的观点，更重要的是**让学生用批判性的思维，拓宽思路，创造性地寻找解决问题的切入点。**

案例教学法起源于1920年代，由美国哈佛商学院 (Harvard Business School) 所倡导

目前普遍用在：**管理学、法学、临床医学**等学科的教学

中国专业学位教学案例中心

- 在教育部和财政部的支持下，中国专业学位教学案例中心建设工作于2013年5月启动
- 案例中心建设工作由国务院学位委员会办公室和教育部指导，教育部学位与研究生教育发展中心牵头，各相关专业教育指导委员会共同参与
 - 2013年6月，工商管理、公共管理、会计、教育、法律、医学教指委启动第一批案例库建设
 -
- 2016年，工程教指委与案例中心合作，开展工程专业案例建设
 - 区别与目前大多数领域的文字案例，重点建设视频案例库

工程专业的教学案例有自己的特点；软件工程专业教学案例也有自己的特殊性。

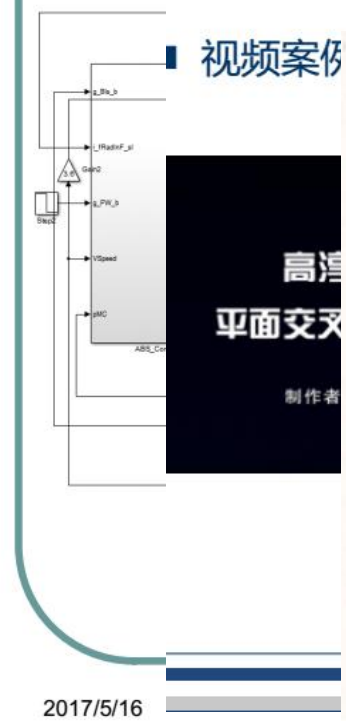
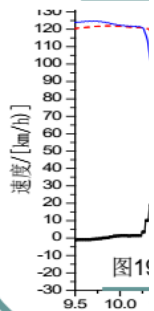
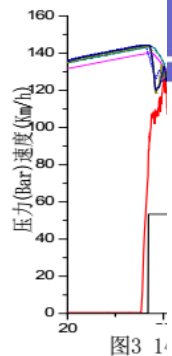
导入案例：伤心的“干细胞”

(CCTV焦点访谈，2015年3月16日)

- 58岁的张英被查出身患**结肠癌晚期**，手术未能阻止癌细胞的扩散。无锡奥斯达公司拥有一种有效的治疗方法。【混淆“研究”与治疗】
- 奥斯达董事长周萱：“一般的癌症根本没有问题。你这个病很有希望的，如果**不来我们这里的就没救了**。”“能治好。**最坏也可以带癌病生存，至少10年**。”【夸大疗效，淡化风险】【患者需要和期望】
- 面对天价治疗费（一个疗程36万元），张英和家人曾想到**放弃**，但若可挽救行将逝去的生命，不算白花，毕竟亲情和**生命高于一切**。女儿火速赶回昆明凑足了36万元。【最佳利益】【风险-受益比】
- 奥思达公司第二天把张英夫妇带到了无锡市滨湖区的一家**社区医院**，为张英进行胚胎干细胞治疗。【资质】【伦理审查】【监管】
 - 治疗结束后，公司说效果很好，可回家休养。但张英的病情急转直下，**三个月后去世**。
 - 为了偿还注射胚胎干细胞所欠下的债务，丈夫卖掉了住了大半辈子的住房，寄居到女儿家，现在还有十多万元外债没有偿还。

案例素材-高速紧急制动仿分析

案例素材-ABS联合仿真模型



拍摄实例

列管式水冷合成气制甲醇合成反应器模拟

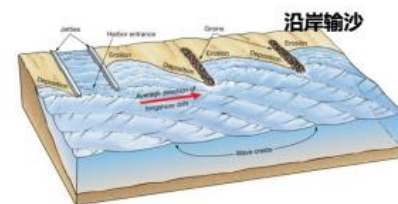
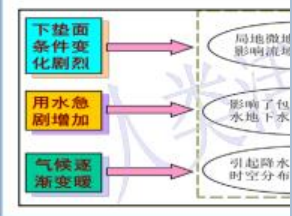


主要内容

- ▶ 水资源动态评价意义
- ▶ 水资源动态评价的
- ▶ 水资源动态评价资
- ▶ 水资源动态评价中
- ▶ 水资源动态评价模
- ▶ 水资源动态评价成

主要内容

- ▶ **海岸波生流**
 - 海岸波浪传播变形、浅化、折射、破碎
 - 海岸波生流流系、向岸质量输移流、沿岸流、裂流... ..
- 沿岸流物理模型试验视频
- 沿岸流现场观测视频
- 裂流现场观测视频
- 砾石海岸整治修复视频



工程专业对可视化的要求更高，更适合视频案例、多媒体案例。

工学专业：案例教学

- 工学专业：强调理论与实践相结合。
- 计算机类专业（计算机、软件工程、物联网等）大部分是工学专业，是实践性很强的专业。
 - 案例可以加深对知识的理解和应用，更好地实现理论和实践相结合
 - 实践不仅是理论的简单应用，真实案例可能更加复杂多样。通过案例教学，可以满足不同的教学目标和定位，可以充分体现工程的复杂性，培养学生解决复杂问题的能力

案例教学与教学案例的关系

➤ 互为依存，又相互独立

- 实施案例教学，必须有教学案例
- 教学案例，是指用于教学的案例

➤ 相互独立

- 同一类案例教学，达到的目的，可以采用不同的案例
- 教学案例，也可以用于不同的案例教学场景

➤ 教学案例库的建设：强调案例的可复用性

- 案例教学中的案例：可复用
- 案例教学的方法：可复用，可以通过文档、视频的形式

教学案例的特点

► 教学案例源于实践，又不同于实践。往往需要加工。

- **真实性原则**：案例内容必须真实。
- **典型性原则**：案例内容要在工程领域中具有一定的代表性，能够反映工程领域主要的理论知识或关键问题。
- **实效性原则**：案例内容应符合当前工程领域的实际情况以及未来发展的方向。
- **完整性原则**：案例应该编写完整，符合案例教学的内容要求、体系完善。
- **启发性原则**：案例的选材和内容应该具有一定的创新性，代表工程领域的理论和实践前沿，可附需要学生讨论的问题，给学生思考的空间，启发学生独立解决问题。
- **适用性原则**：国外案例与国内案例相结合的原则，并结合我国的国情和时代背景。

案例的分类1--按使用场景

➤ 课堂教学使用的案例

- 示范性：形象讲述知识点
- 启发型案例：促进学生思考
- 研讨型案例：从工程问题引入，层层深入，引发讨论，答案往往不唯一
- 代入型案例：从某一个典型事件代入，讲解知识
- 项目型：贯穿课程，将知识点串起来

➤ 实践教学使用的案例

- 实践课程使用的案例，一般以项目的形式
- 系统型案例：相对完整的系统
- 过程型案例：如敏捷过程

➤ 考评使用的案例

- 以考核学生能力的形式

案例的分类2：案例的内容

➤ 项目型案例

- 课堂用，辅助老师讲解知识，增加系统性
- 实践课程用，学生要完成的项目（部分或者全部）

➤ 知识点型案例

- 针对知识点，可大可小，形式多样。视频案例、动画、文字等等。

➤ 评测型案例

- 理论课考试
- 实验课考试

案例的分类3：案例的形式分

➤ 文本形式：

- 案例的文字描述、代码

➤ 模型

- 需求、设计模型等，如UML模型
- 仿真模型

➤ 视频

- 原理讲解型视频
- 真实情况录制/还原视频

➤ 多媒体

- 多种存在形式，如虚拟现实、动画等

编译课堂使用的案例

- 可以是针对某个知识点的小案例
 - ✓ 如何设计好案例
- 可以是贯穿这个编译课程的系统案例
 - 一个小型编译系统，渐进式讲解案例

一个案例

➤ 为了让学生更有兴趣学习编译，我一直在寻找一个例子

一个简单C程序的运行时结构

```
➤ #include <stdio.h>
➤ #include < string.h >
void fun1()
{
    int m=10;
    char num[4];
    strcpy(num, "bbbb" );
}
void fun2()
{
    printf ( "You were attacked!!!\n" );
}
int main()
{
    fun1();
    return 0;
}
```

```

➤ #include <stdio.h>
➤ #include <string.h>
void fun1( )
{
    int m=10;
    char num[4];

    strcpy(num, "bbbbbbbbbbbbbb\x0F\x10\x40\x00" );
}
void fun2( )
{
    printf ( "You were attacked!!!\n" );
}
int main()
{
    fun1();
    return 0;
}

```

- #include <stdio.h>
- #include <string.h>

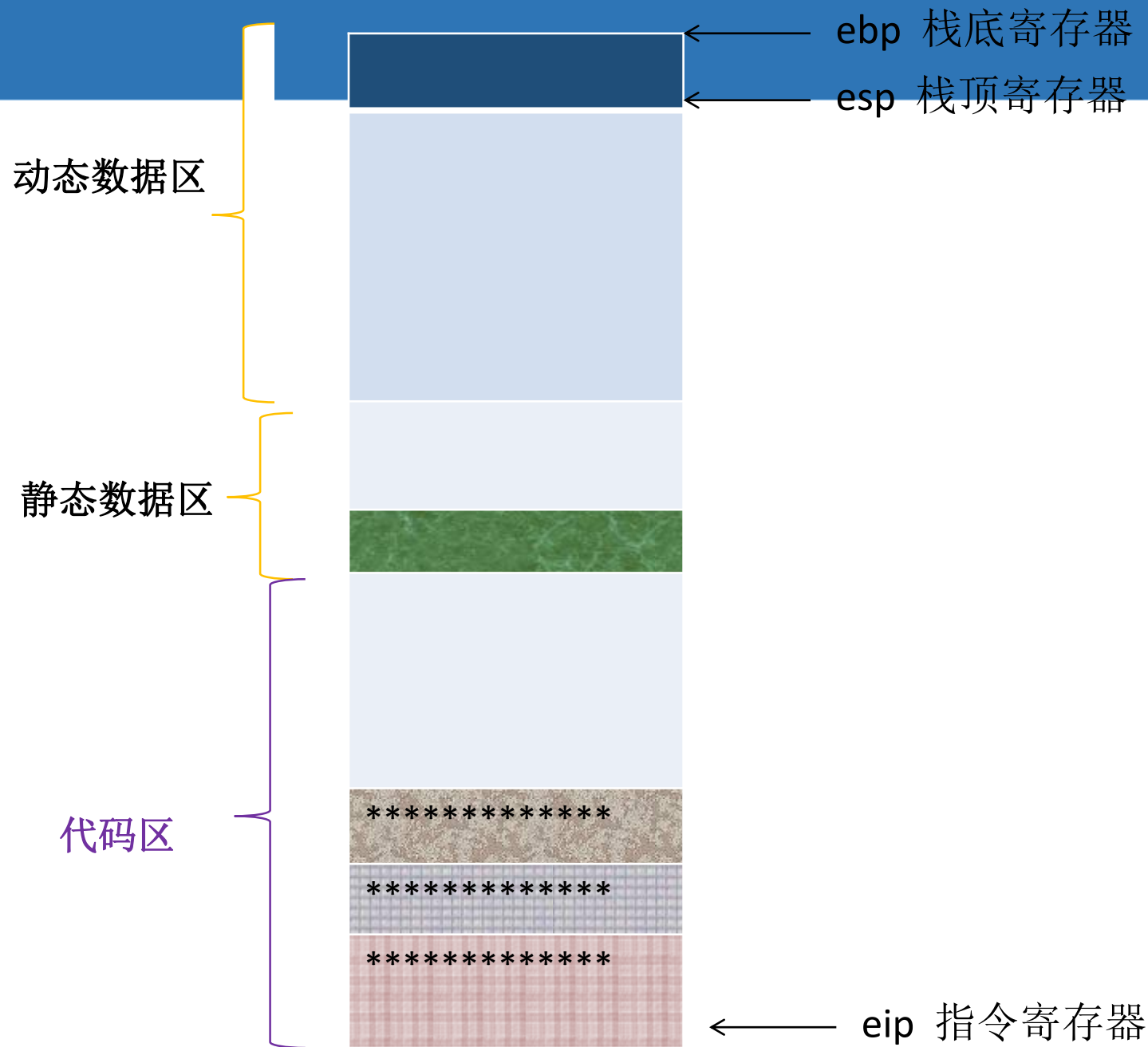
```

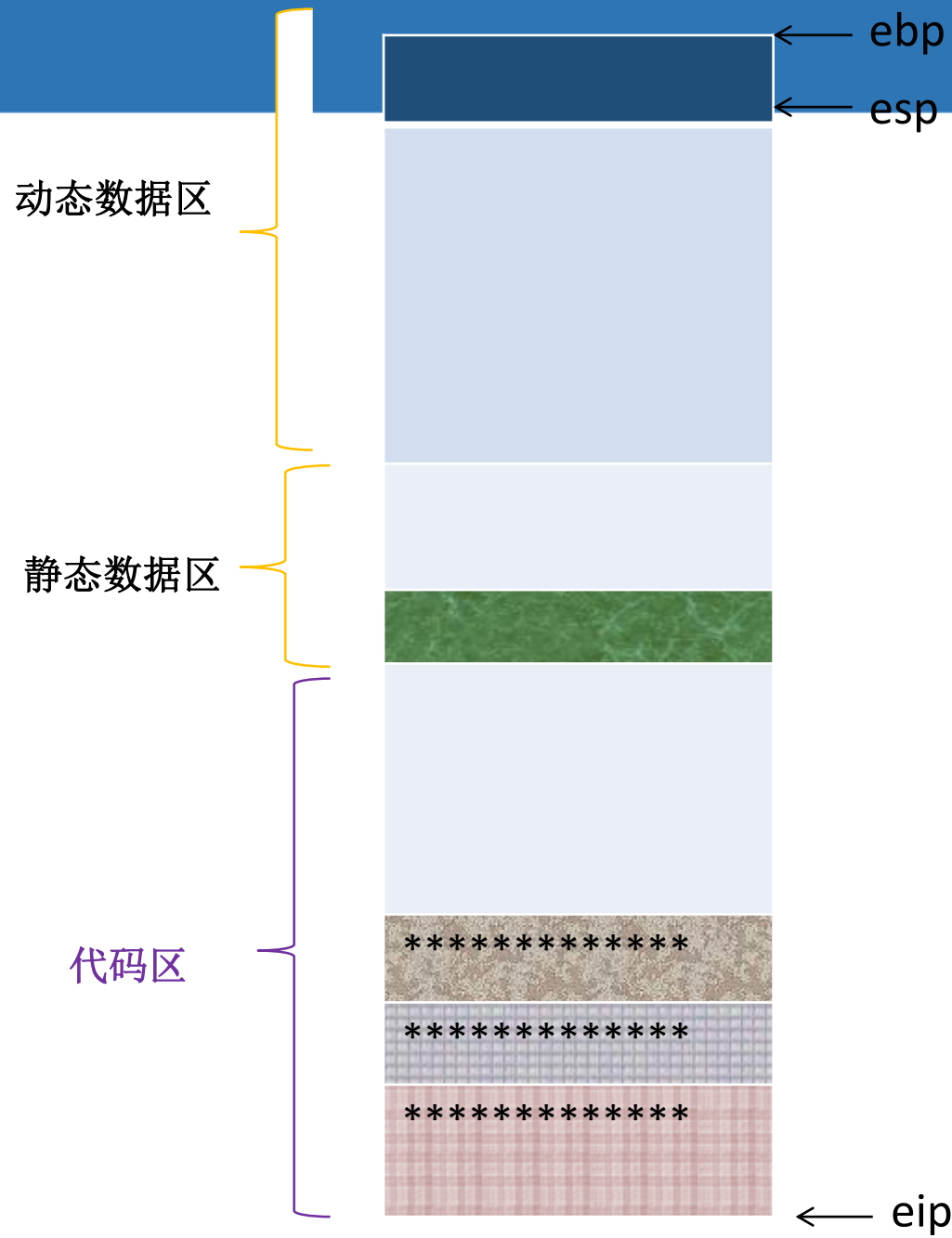
void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}
void fun2( )
{
    printf ("You were attacked!!!\n");
}
int main()
{
    fun1();
    return 0;
}

```


Why?

➤有同学愿意讲讲吗？

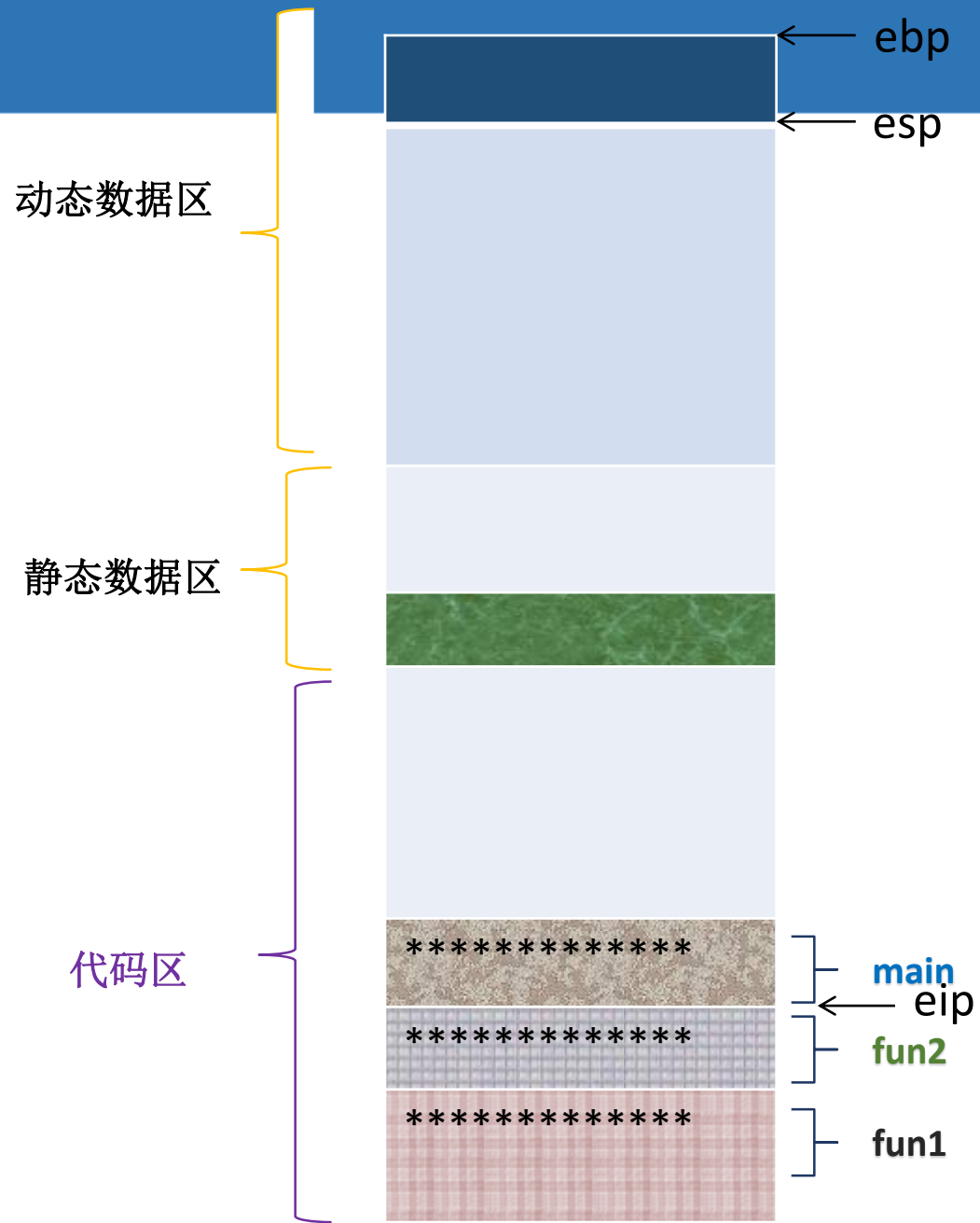




```

• #include <stdio.h>
• #include <string.h >
void fun1( )
{
  int m=10;
  char num[4];
  strcpy(num,"bbbb");
}
void fun2( )
{
  printf ("You were attacked!!!\n");
}
int main()
{
  fun1();
  return 0;
}

```



```

• #include <stdio.h>
• #include <string.h >
void fun1( )
{
  int m=10;
  char num[4];
  strcpy(num,"bbbb");
}
void fun2( )
{
  printf ("You were attacked!!!\n");
}
int main()
{
  fun1();
  return 0;
}

```

fun函数执行后的
返回地址



```

• #include <stdio.h>
• #include < string.h >
void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}
void fun2( )
{
    printf ("You were attacked!!!\n");
}
int main()
{
    fun1();
    return 0;
}
    
```

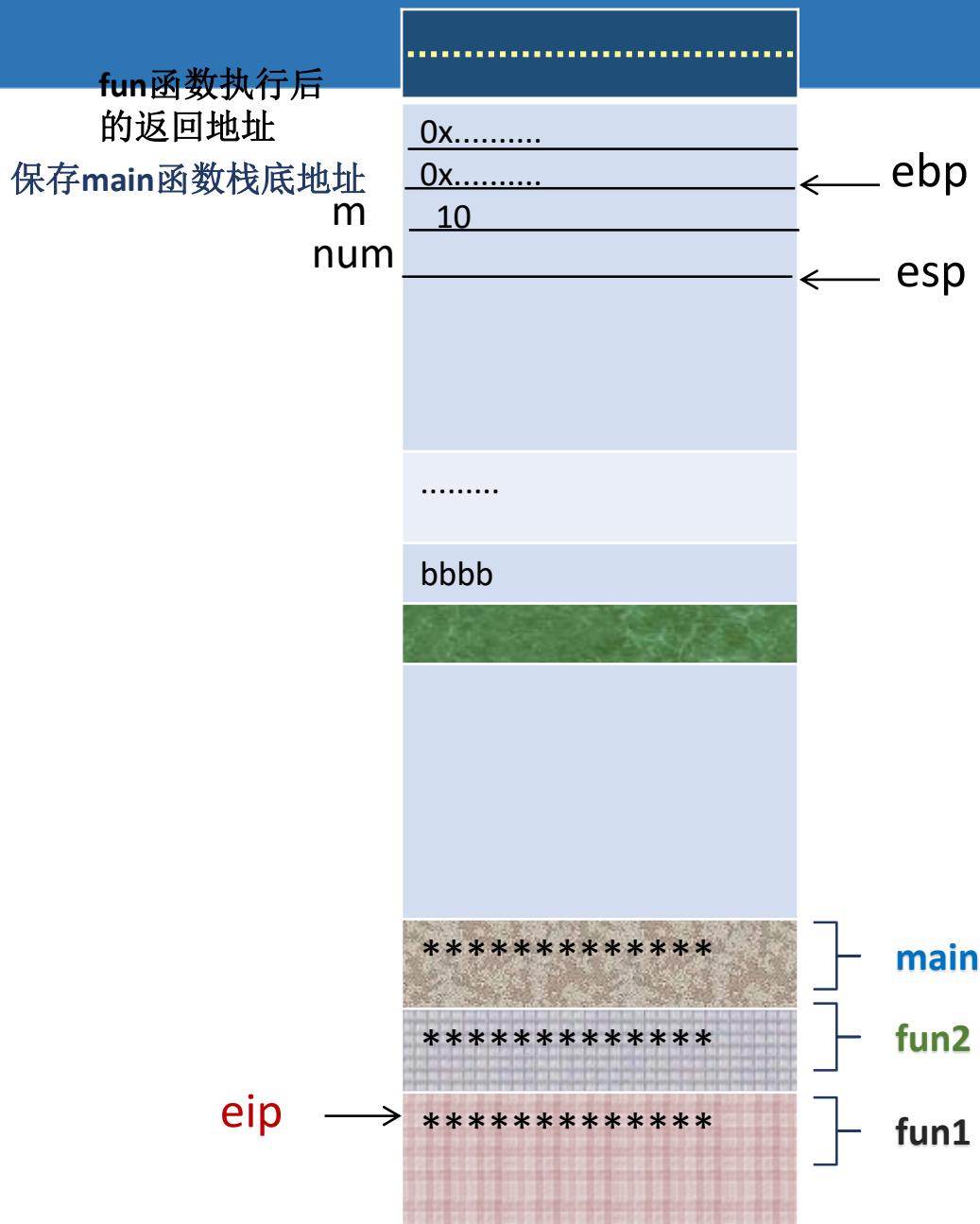
fun函数执行后的
返回地址
保存main函数栈底地址



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

```
void fun1( )
{
  int m=10;
  char num[4];
  strcpy(num,"bbbb");
}
```

```
void fun2( )
{
  printf ("You were attacked!!!\n");
}
int main()
{
  fun1();
  return 0;
}
```



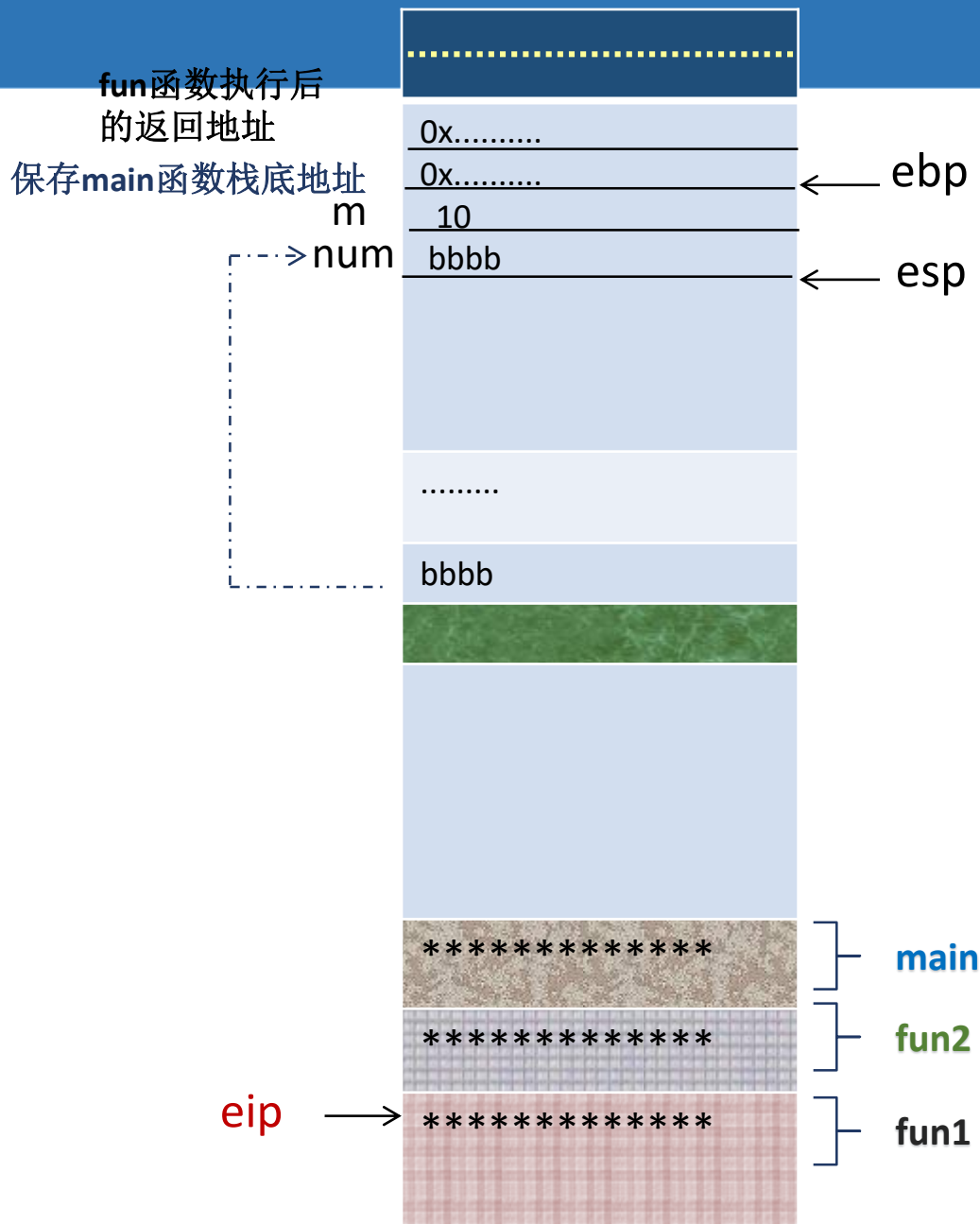
```

• #include <stdio.h>
• #include <string.h >

void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}

void fun2( )
{
    printf ("You were attacked!!!\n");
}

int main()
{
    fun1();
    return 0;
}
    
```



```

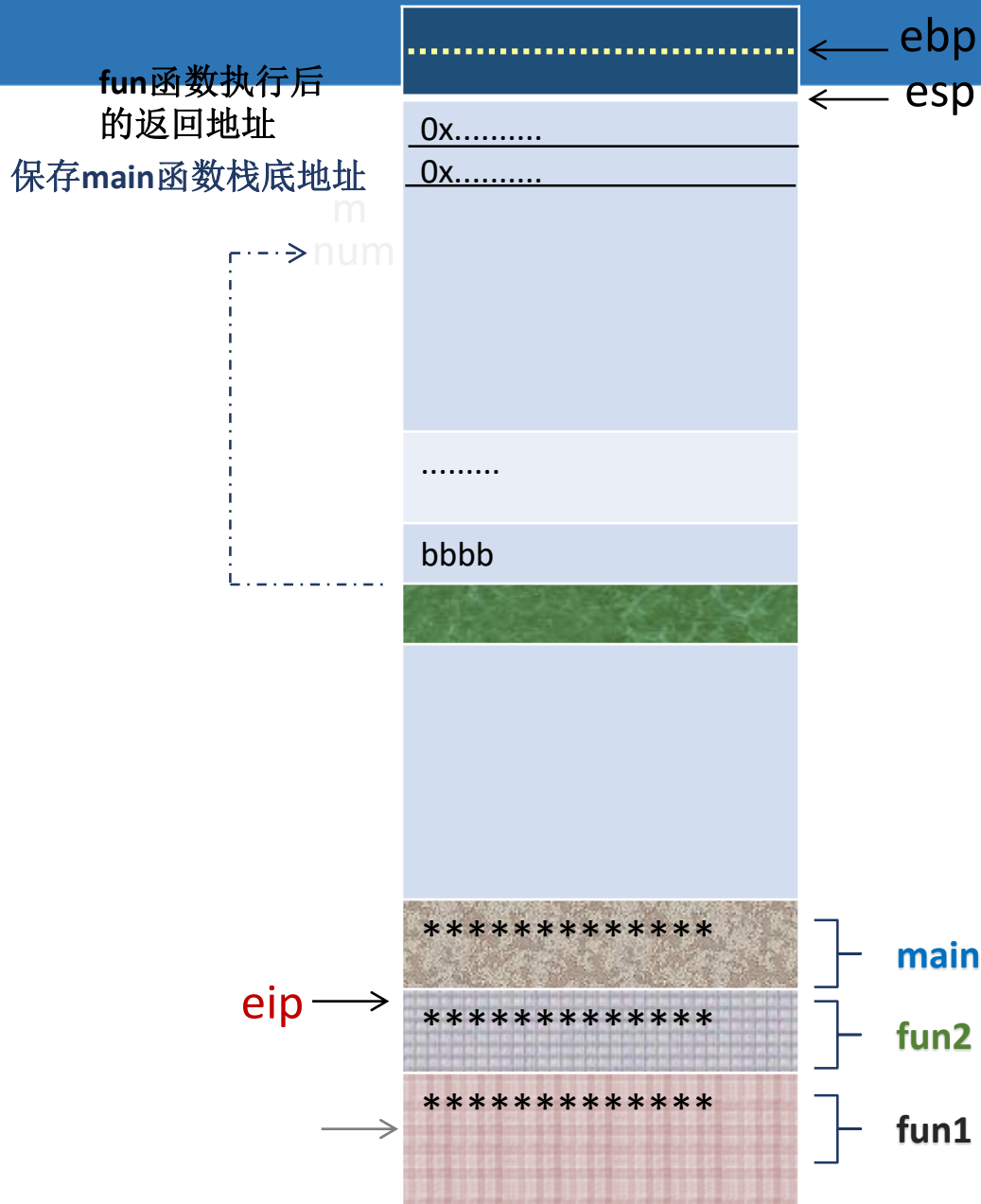
• #include <stdio.h>
• #include <string.h >

void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}

void fun2( )
{
    printf ("You were attacked!!!\n");
}

int main()
{
    fun1();
    return 0;
}
    
```


fun1结束，返回main函数



- #include <stdio.h>
- #include <string.h >

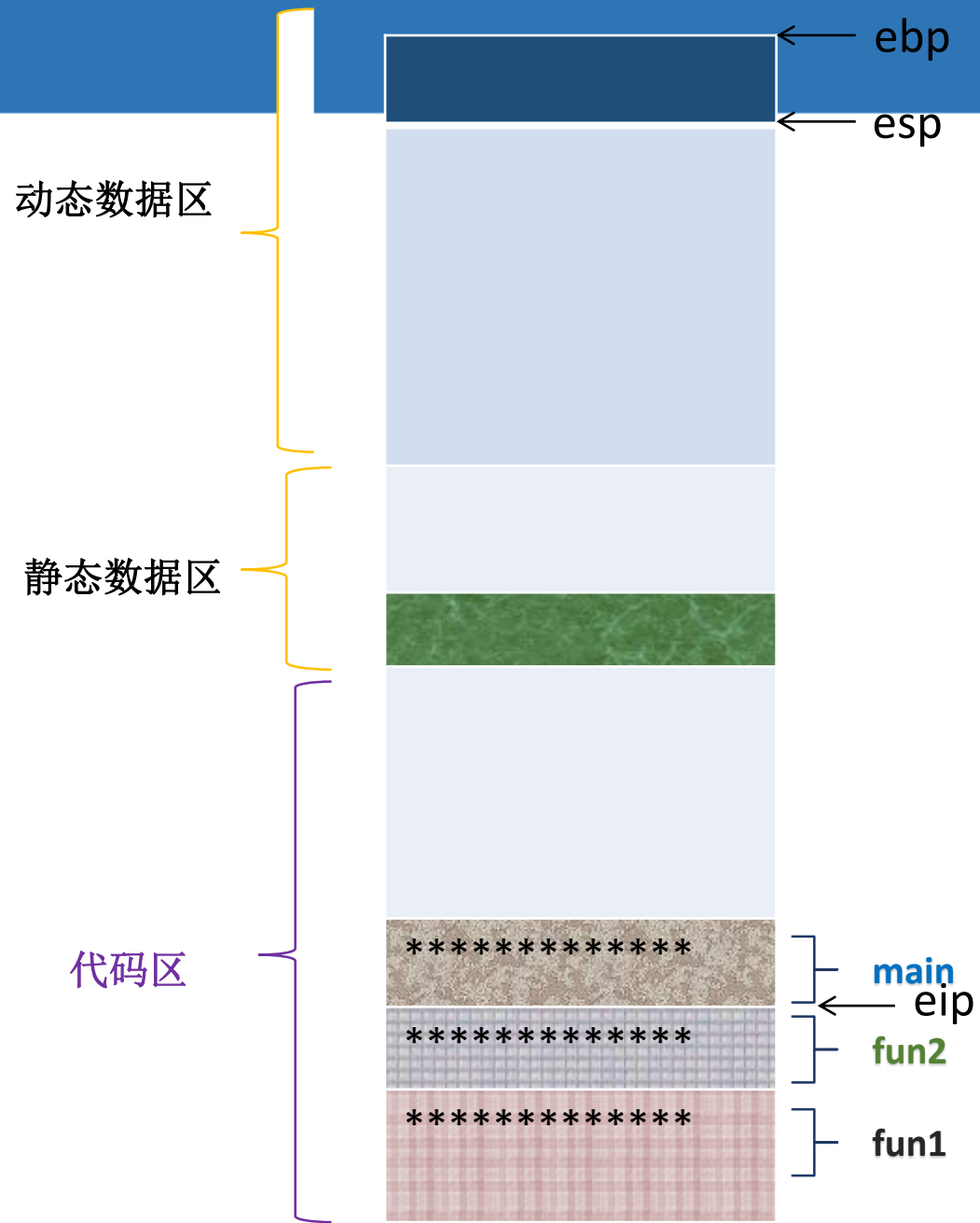
```

void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}

void fun2( )
{
    printf ("You were attacked!!!\n");
}

int main()
{
    fun1();
    return 0;
}
    
```

第二个程序-初始化：程序加载



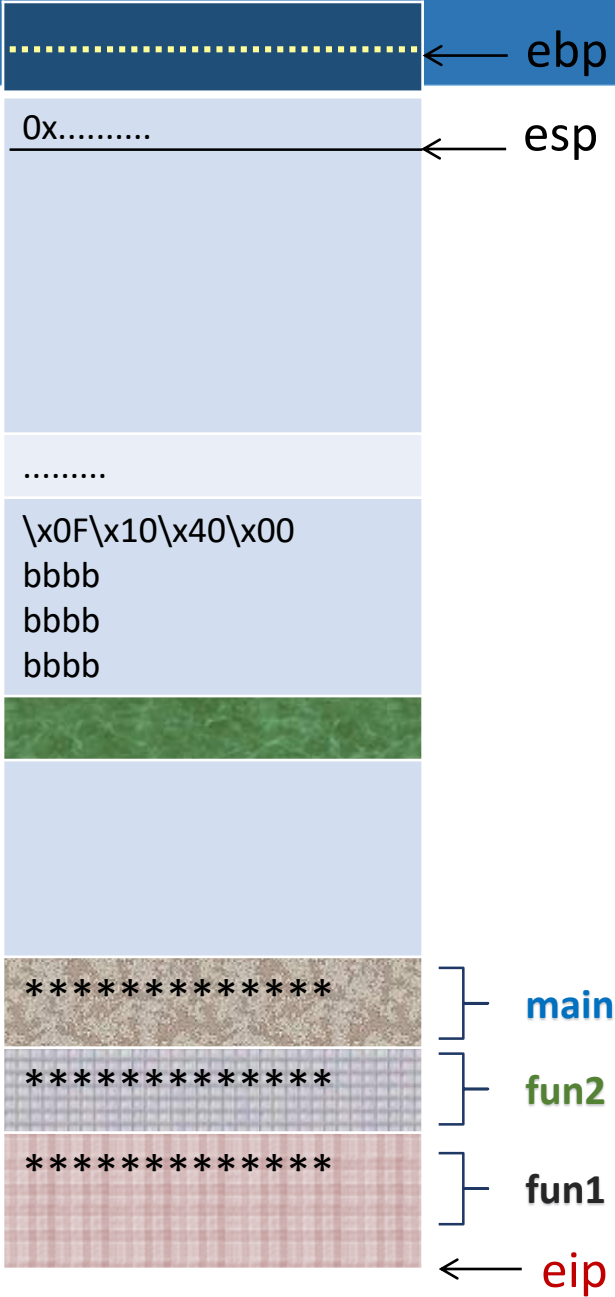
```
• #include <stdio.h>
• #include <string.h >

void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbbbbbbbb\x0F\x10\x40\x00")
;
}

void fun2( )
{
    printf ("You were attacked!!!\n");
}

int main()
{
    fun1();
    return 0;
}
```

fun函数执行后的返回地址

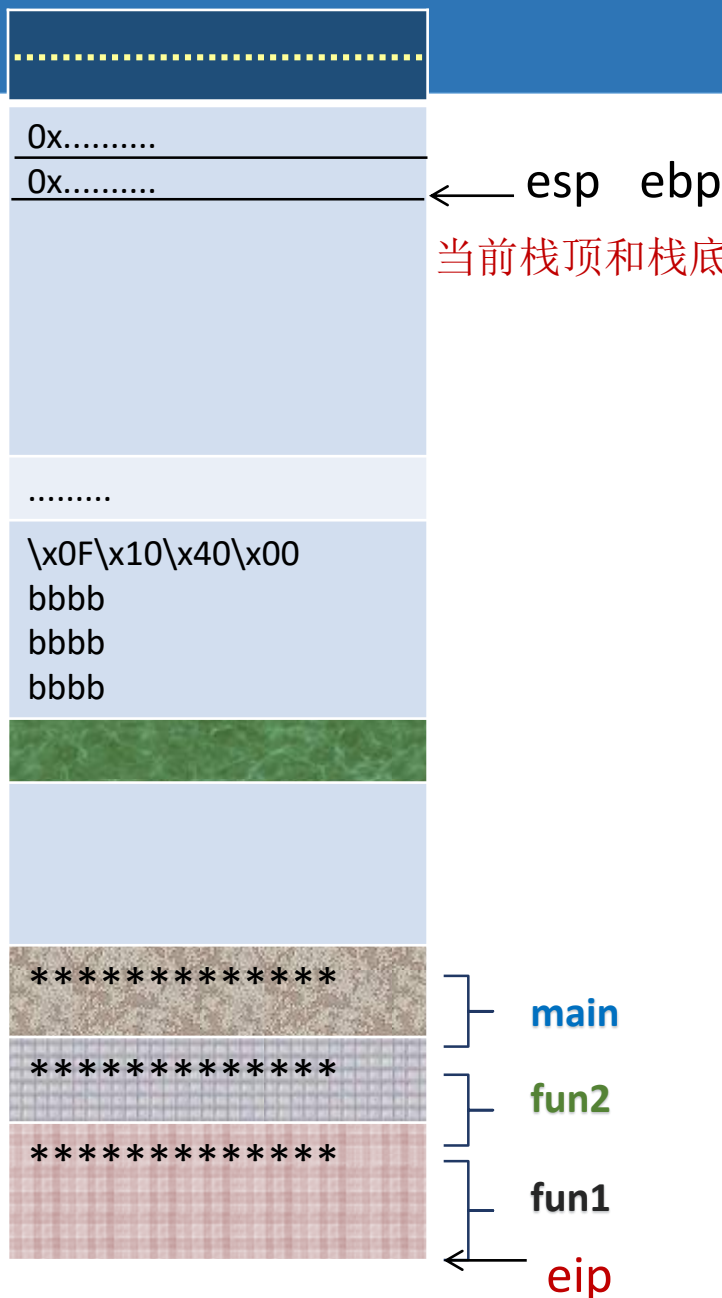


```

• #include <stdio.h>
• #include <string.h >
void fun1( )
{
  int m=10;
  char num[4];
  strcpy(num,"bbbbbbbbbb\x0F\x10\x40\x00")
;
}
void fun2( )
{
  printf ("You were attacked!!!\n");
}
int main()
{
  fun1();
  return 0;
}

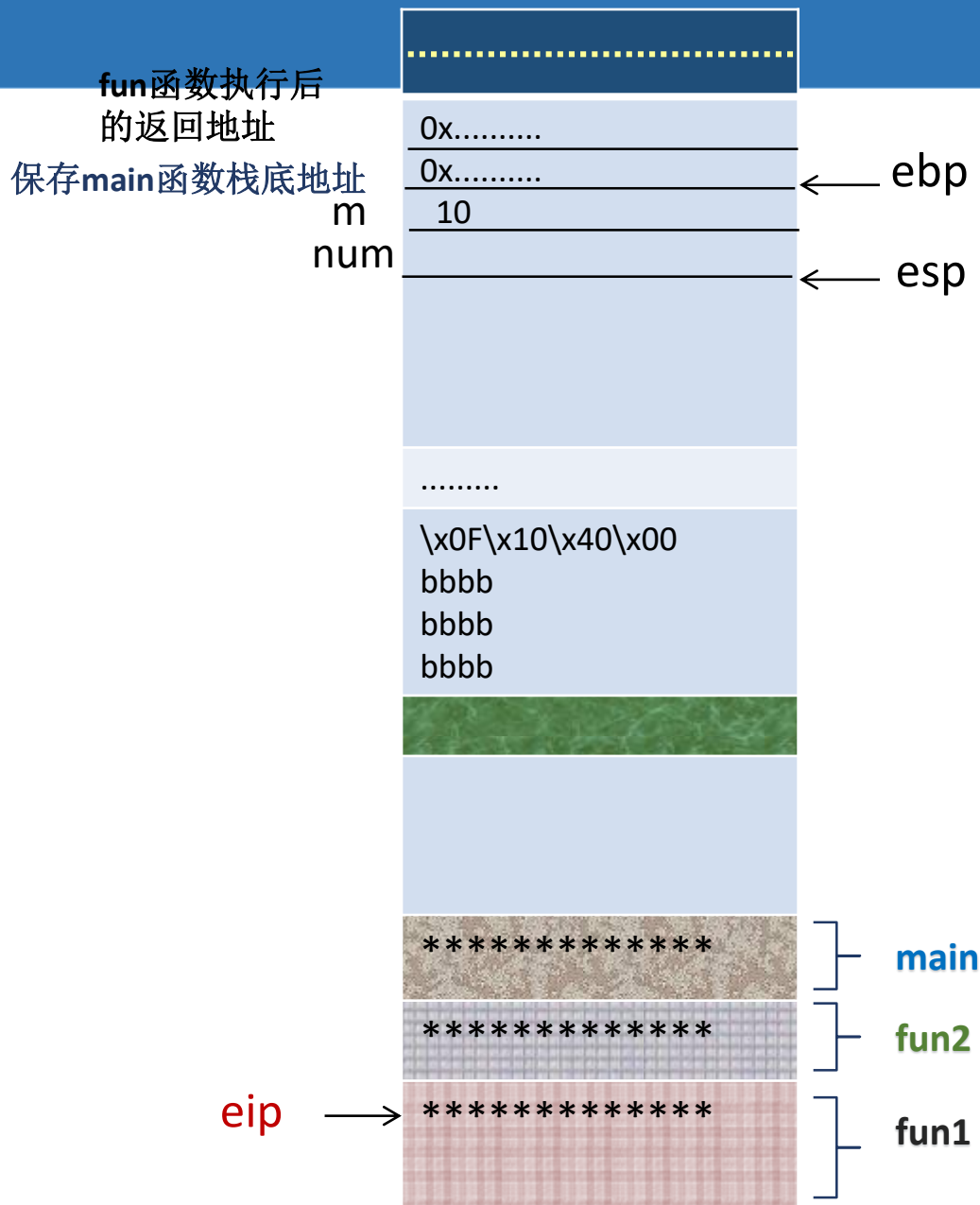
```

fun函数执行后的
返回地址
保存main函数栈底地址



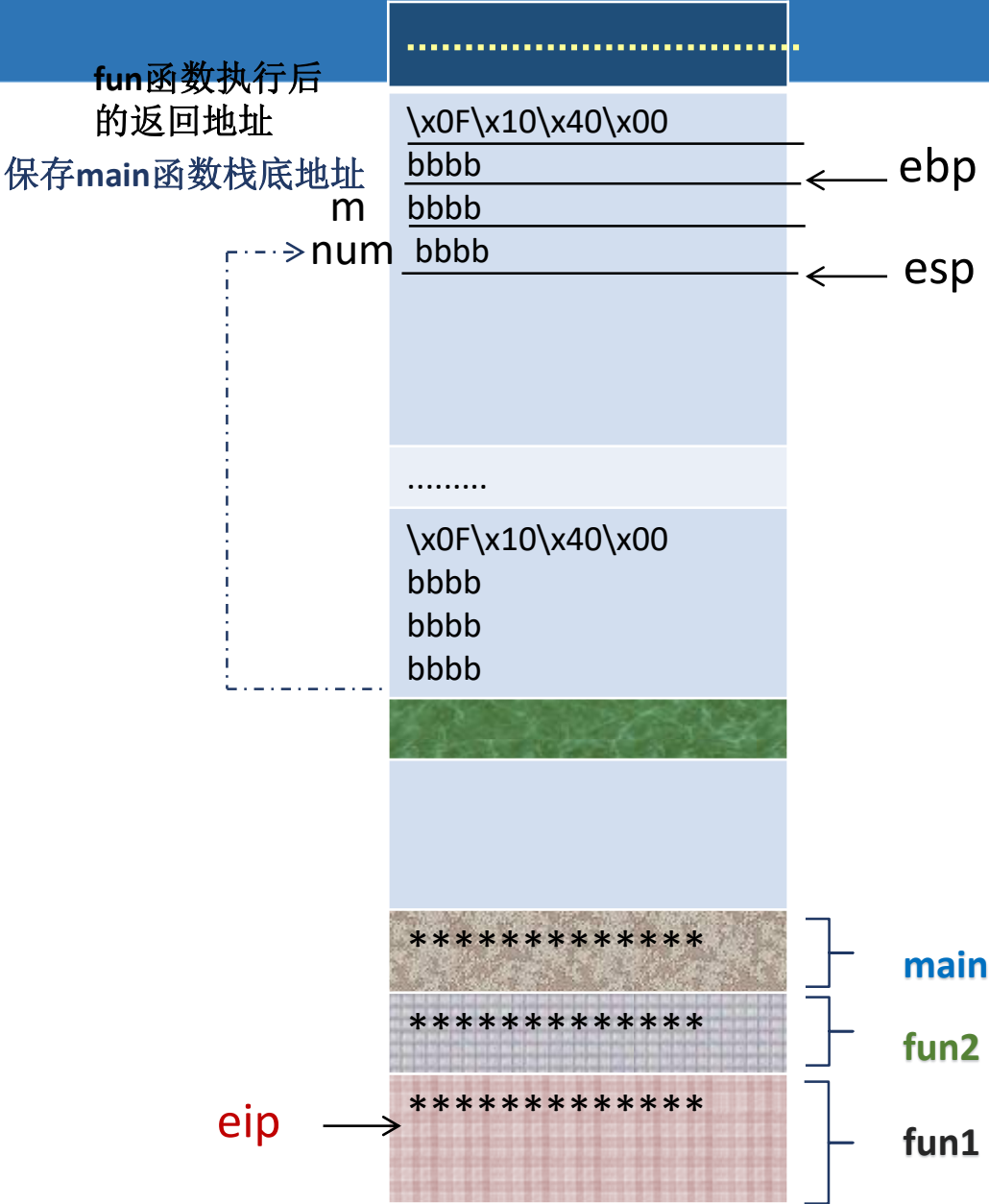
```

• #include <stdio.h>
• #include < string.h >
void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbbbbbbbb\x0F\x10\x40\x00")
;
}
void fun2( )
{
    printf ("You were attacked!!!\n");
}
int main()
{
    fun1();
    return 0;
}
    
```



```

• #include <stdio.h>
• #include < string.h >
void fun1()
{
    int m=10;
    char num[4];
    strcpy(num,"bbbbbbbbbb\x0F\x10\x40\x00")
;
}
void fun2( )
{
    printf ("You were attacked!!!\n");
}
int main()
{
    fun1();
    return 0;
}
    
```



- #include <stdio.h>
- #include < string.h >

```

void fun1()
{
  int m=10;
  char num[4];
  strcpy(num,"bbbbbbbbbbbb\x0F\x10\x40\x00")
;
}

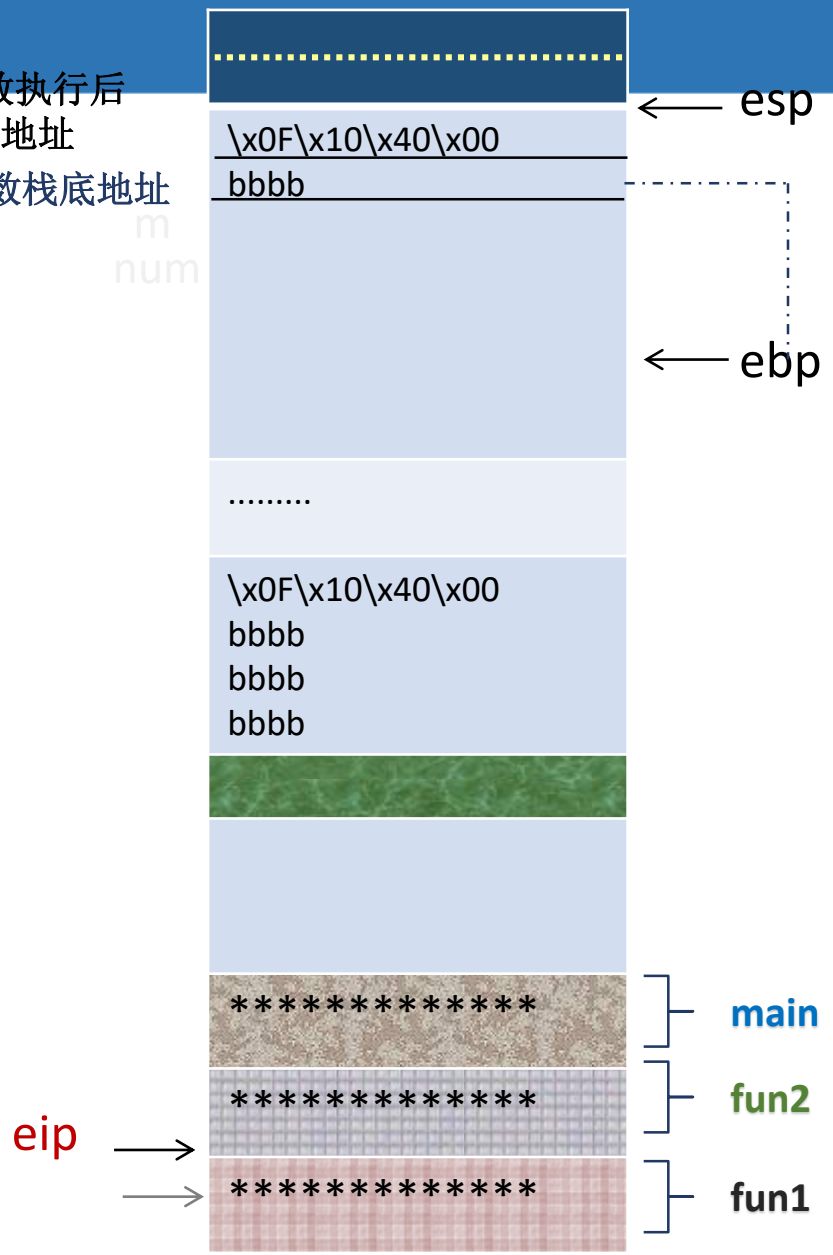
void fun2( )
{
  printf ("You were attacked!!!\n");
}

int main()
{
  fun1();
  return 0;
}

```

fun1结束，返回main函数

fun函数执行后的
返回地址
保存main函数栈底地址
m
num



- #include <stdio.h>
- #include <string.h >

```

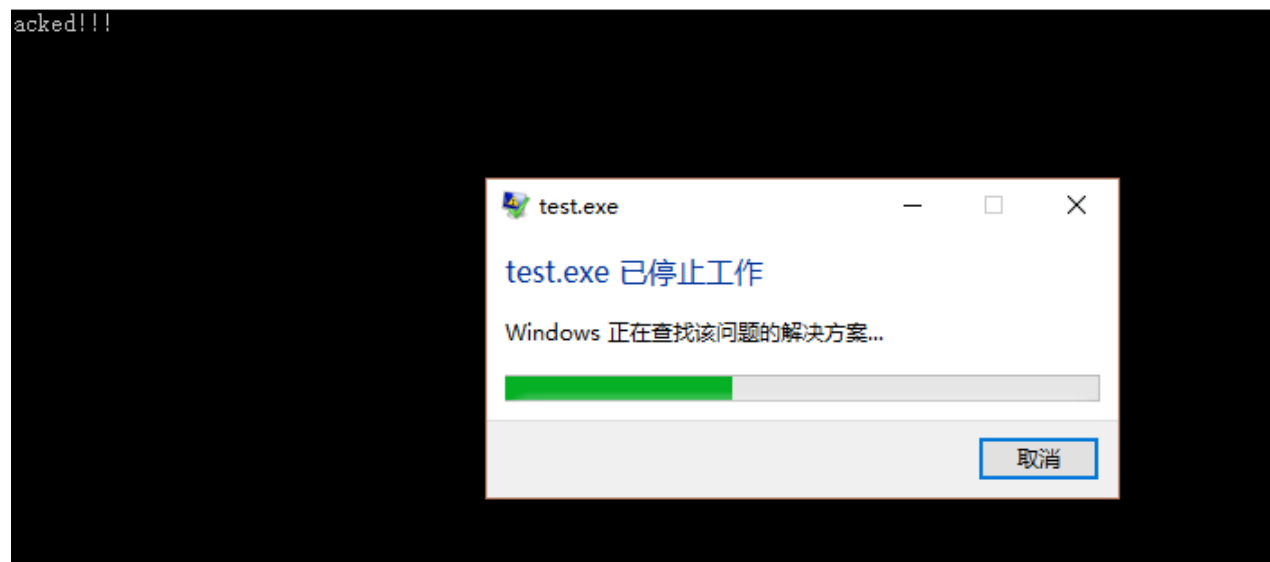
void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbb");
}

void fun2( )
{
    printf ("You were attacked!!!\n");
}

int main()
{
    fun1();
    return 0;
}
    
```

- **在C语言中，栈的方向是从高地址向低地址延伸，而数组中数据在栈中的存储方向正好相反。**
- **字符串拷贝等数组操作是不对数据长度做审核的。如果实际的数据长度超过了栈中预留的空间，就会将栈中其他数据覆盖。---**
“栈溢出”。
 - **可能导致不可预测的错误**
 - **也可能导致一个精心策划的执行流程发生改变**
- **因此，是否能够对自己所写程序的运行时状态做到心中有数，是能否写出高质量、安全代码的前提保证。进一步再考虑尽量不要给黑客留漏洞。**

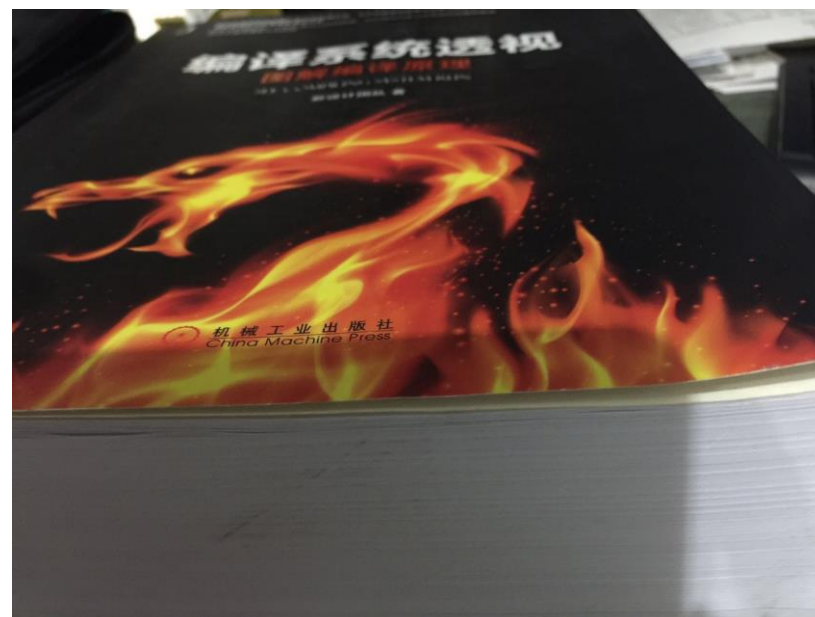
- 可惜，我运行了一下，fun2并没有被调用。
- 试了几个编译器，都没有
- 如果用32位的gcc运行的话 直接会提示这个错误



- #include <stdio.h>
- #include <string.h>

```
void fun1( )
{
    int m=10;
    char num[4];
    strcpy(num,"bbbbbbbbbb\x0F\x10\x40\x00")
;
    printf(m,%d);
}
void fun2( )
{
    printf ("You were attacked!!!\n");
}
int main()
{
    fun1();
    return 0;
}
```

改为Java语言， m的值没有被修改



1041页

谁错了？

你了解你用的编译器吗？

其他课程案例

- C语言的存储管理案例
- Java语言的存储管理案例

- C语言的符号表管理
- Java语言的符号表管理

- 典型的代码生成

项目实践的案例

- 目前，北航是实践环节，是实现一个小型编译器
- 为了防止抄袭，同时保证学生可以完整设计和实现一个编译器
- 目前：
 - 案例编译器，C0文法，Pascal文法，学生读懂，教师检测
 - 提供分级、不同难度系数的文法，学生选难度系统，系统随机分配文法
 - 尽可能支持自动化评判
 - 为此需要大量的文法案例—测试案例，自动化测试支撑。
 - 优化部分，针对不同文法的优化比赛（机器自动排名）

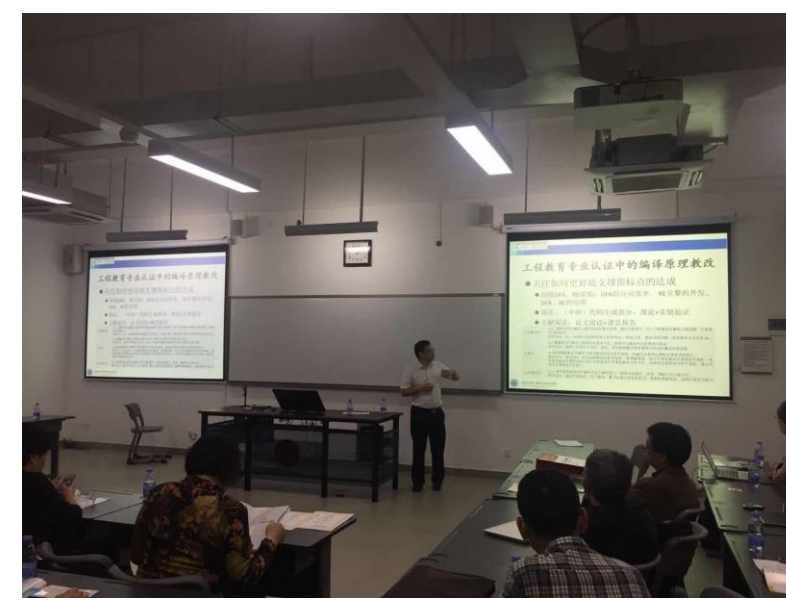
2018年的案例比赛情况

- 2018年正值“软件工程”正式提出50周年之际，在教育部推动新工科教学改革的背景下，CCF软件工程专委决定启动建立中国软件工程教学案例库。
- 同时首次举办案例比赛，旨在征集和推广软件工程领域的优秀教学案例。
- 本次比赛吸引众多985、211高校及企业参加。
- 资深专家评审团从教学案例专业性、完整性、新颖性、有用性、可操作性等维度评分。
- 在49个参赛案例中，17个案例通过初评。

优秀教学案例交流和评奖比赛

- 2018年11月23日下午，软件工程优秀教学案例交流和评奖比赛在深圳大学顺利举行。
- 本次大赛邀请了来自北京大学、复旦大学、北京航空航天大学、武汉大学、四川大学、云南大学的7位专家担任现场评委。
- 17个案例团队介绍软件工程教学案例内容及其使用方法，回答专家的问题。

优秀教学案例交流和评奖比赛 (续)



案例比赛评奖

- 专家们听取各参赛案例的汇报后，最终评选3个一等奖、4个二等奖和多个优秀奖。



成绩和遗憾

➤ 首先遗憾：非常优秀的老师，落选了。Why?

- 案例教学与教学案例
- 案例的可重用性

➤ 第一次，大家也一直在希望我们提供模板，但是我们没有：

- 第一，我们不敢给
- 第二，不愿意限制大家的发挥空间

CCF-SE软件工程案例征集及比赛 获奖名单

一等奖案例名称	获奖人员	单位
多旋翼飞行器设计与应用	廖勇、陈逸奇、孙玉婷、桑楠、杨茂林、葛旭阳、张晋川、杨坤山、施荣圳、吴金添	电子科技大学
基于小米便签的开源软件阅读维护	尹良泽、毛新军	国防科技大学
契约式软件设计与测试的闯关式案例	孙青、吴际	北京航空航天大学

二等奖案例名称	获奖人员	单位
通用剧院票务管理系统（C语言版）	舒新峰、梁琛、张丽丽、黄茹、王小银	西安邮电大学
如何开展站立晨会	刘玉媛、胡娟	腾讯
WOWO空间分时共享平台	赵娜、王剑、王炜、赵明雄、谢诚、郁湧	云南大学
UML建模之旅：旅游业务申请系统分析与设计建模	谭火彬、林广艳	北京航空航天大学

多旋翼飞行器设计与应用

奖项	一等奖
作者信息	<p>廖勇 (电子科技大学)</p> <p>陈逸奇 (大疆深圳市大疆创新科技有限公司)</p> <p>孙玉婷 (大疆深圳市大疆创新科技有限公司)</p> <p>桑楠 (电子科技大学)</p> <p>杨茂林 (电子科技大学)</p> <p>葛旭阳 (阿里巴巴网络技术有限公司)</p> <p>张晋川 (百度 (北京))</p> <p>杨坤山 (电子科技大学)</p> <p>施荣圳 (电子科技大学)</p> <p>吴金添 (电子科技大学)</p>
案例类型	项目式教学案例
适用学生	软件工程、计算机、自动化、机电专业等专业研究生
推荐使用课程	《嵌入式系统设计》、《高级计算机结构》、《软件架构与设计》、其它专业相关课程
覆盖知识点	多旋翼飞行器原理及基本知识、计算机控制技术、嵌入式操作系统、计算机组成原理、汇编语言与接口技术、C 程序设计、数据结构、嵌入式系统设计、高级计算机结构、软件架构与设计等课程核心知识点。
教学目标与用途	<p>本案例以学生为主体，以多旋翼飞行器的设计与实现为主线，探讨系统工程能力及创新思维能力的培养。</p> <p>通过该案例及课堂教学环节辅助学生建立起计算机系统能力，为毕业后成为高层次软件系统工程师提供支持。</p>
摘要	<p>本案例与深圳市大疆创新科技有限公司 (DJI) 联合建设，首先构建了一个简单多旋翼飞行器原型系统，再循序渐进，基于大疆公司 MATRICE 100 四轴翼飞行器、GUIDANCE 图像采集模块、异构多核嵌入式平台 Tegra K1，设计了一个基于异构多核嵌入式平台的四轴翼飞行器，并对软件系统进行并行优化，从而实现基于双目立体视觉的 3D 地图重建、路径规划及躲避障碍等功能。</p> <p>本案例建设做到了前沿理论与实际产业的紧密结合，落实了产学紧密渗透，相关技术综合性强，有利于学生交叉复合和交叉融合能力与素质的形成。</p>



附件清单

案例相关的附件文件清单：

- 1) 案例教学视频
- 2) 案例描述
- 3) 案例教学与研讨幻灯片
 - (1) 多旋翼飞行器基础
 - (2) 搭建飞行器嵌入式开发环境
 - (3) 嵌入式实时软件系统设计方法
 - (4) 三维重建
 - (5) 嵌入式操作系统
- 4) 课前预习资料清单（见使用说明 3.2 节）
- 5) 四轴飞行器演示系统一套（有条件的班级）

基于小米便签的开源软件阅读维护


奖项	一等奖
作者信息	尹良泽 (国防科技大学) 毛新军 (国防科技大学)
案例类型	项目式教学案例
适用学生	高年级本科生
推荐使用课程	《软件工程》、《软件体系结构与设计》等
覆盖知识点	功能分析、体系结构设计、详细设计、软件代码质量。
教学目标与用途	通过让学生阅读维护优秀软件工程师所开发的优秀软件项目让学生建立对较大规模高质量软件的感性认识，为软件工程方法和技术的理论教学和后期的软件项目实践打下基础。同时培养学生对较大规模复杂软件的分析、设计、和开发能力，以及基于 Git 的团队协作开发和复用开源软件实现软件功能的能力。
摘要	针对软件工程课程实践教学中学生缺乏大规模和高质量软件开发经验以及高质量软件设计开发能力不足的问题，本案例通过在软件工程课程实践中让学生阅读和维护优秀开源软件来增强学生的工程意识，快速建立学生对大规模和高质量软件感性认识，从而提高学生设计开发大规模高质量软件的综合能力。


[CCF-SE教学案例提交表.pdf \(89 KB\)](#)

[案例使用说明书.pdf \(348 KB\)](#)

契约式软件设计与测试的闯关式案例

奖项	一等奖
作者信息	孙青（北京航空航天大学） 吴际（北京航空航天大学）
案例类型	知识点教学案例
适用学生	本案例适用于计算机相关专业二年级及以上本科生
推荐使用课程	面向对象程序设计类课程
覆盖知识点	面向对象编程、规格设计、测试用例设计。
教学目标与用途	在面向对象相关课程教学中，通过案例的训练，使得学生能够理解面向对象程序设计规格的重要性，通过数据抽象和方法抽象的具体实现，掌握面向对象规格设计的方法；初步掌握如何通过程序规格设计测试用例。
摘要	<p>在面向对象编程中，通过规格来开展程序设计，可以在类实现之前就确定类及其关系，以及方法的行为要求，从而有效保证软件整体行为的逻辑清晰性和功能稳定性。本案例训练 Java 程序规格设计，并基于程序规格开展单元测试。</p> <p>案例选择了单电梯调度系统为实现背景，给出一份完整的软件需求说明以及不完整的实现代码。采用代码及设计规格交叉补充的方式，训练学生对需求和规格设计的理解，以及补全代码来实现完整需求的能力，加深学生对于规格设计的理解和体会。在此基础上，要求学生依据设计规格设计测试用例，进一步理解通过规格设计开展契约式编程的重要意义。训练为限时完成，要求学生在规定的训练时间内，分析需求，并从数据抽象、方法抽象、测试用例设计几个方面完成具体的训练任务。</p>

 CCF-SE教学案例提交表.pdf (99 KB)

 案例使用说明书.pdf (336 KB)

CCF-SE软件工程案例征集及比赛 优秀奖名单

案例名称	获奖人员	单位
面向服务的软件系统开发实验课	涂志莹、王忠杰、黄虎杰	哈尔滨工业大学
《软件工程》实训课程 & 教学案例	于海雯	南昌大学
基于Web2.0的合作式学习平台	边耐政	湖南大学
代码圈复杂度的案例	廖力、李必信	东南大学
腾讯TAPD团队如何做敏捷研发	胡娟、刘玉媛	腾讯
基于项目的C++ 语言实践教学案例	郭勇、杨大易	哈尔滨工业大学
面对区块链的软件工程教学实践	欧阳建权、史佩昌、王怀民	湘潭大学、国防科技大学
毕设选导系统原型设计	张栋、林为伟、朱丹红	福州大学
软件工程化流程及软件过程的交付成果	韩万江、杨金翠、张笑燕、陆天波、孙艺	北京邮电大学
软件需求分析与设计实例-网上购物系统	姜瑛、丁家满	昆明理工大学

➤ 案例上线

➤ https://www.educoder.net/moop_cases

➤ 第十八届全国软件与应用学术会议(NASAC 2019)

- 会议时间：2019年11月22日至24日 会议地点：杭州
- 网址：<http://nasac2019.zju.edu.cn/>

首页
执行委员会
大会主席
程序委员会
组织委员会
大会征文 HOT
教育论坛
特邀报告人
顶会论坛
青年学者论坛
优秀博士生论坛
其它论坛/并行会议
软件工程教学案例
+ 原型系统竞赛 NEW
录用公告
会议日程
个人自荐书 NEW

CFF-SE软件工程教学案例征集通知

一. 活动背景

软件工程学科是一门实践性很强的学科，理论联系实践是教学的重要特点。基于案例的教学有助于深入理解理论知识，培养解决复杂软件工程问题的能力。然而，目前软件工程领域缺少成熟的教学案例库，特别是优秀的教学案例更是缺乏。

为了汇集和推广软件工程领域的优秀教学案例，2018年Nasac会议举办软件工程优秀教学案例交流和评奖比赛。本次比赛众多985、211高校及企业参加，从49个参赛案例评选出17个案例参加现场交流和评比，最终评选出一等奖、二等奖和优秀奖。

鉴于第一次比赛成功举办，我们于今年举办第二次软件工程教学案例征集及比赛活动，征集本科生和研究生的教学案例，并在征集的案例中推荐参加NASAC2019举办的优秀教学案例比赛。

二. 案例征集

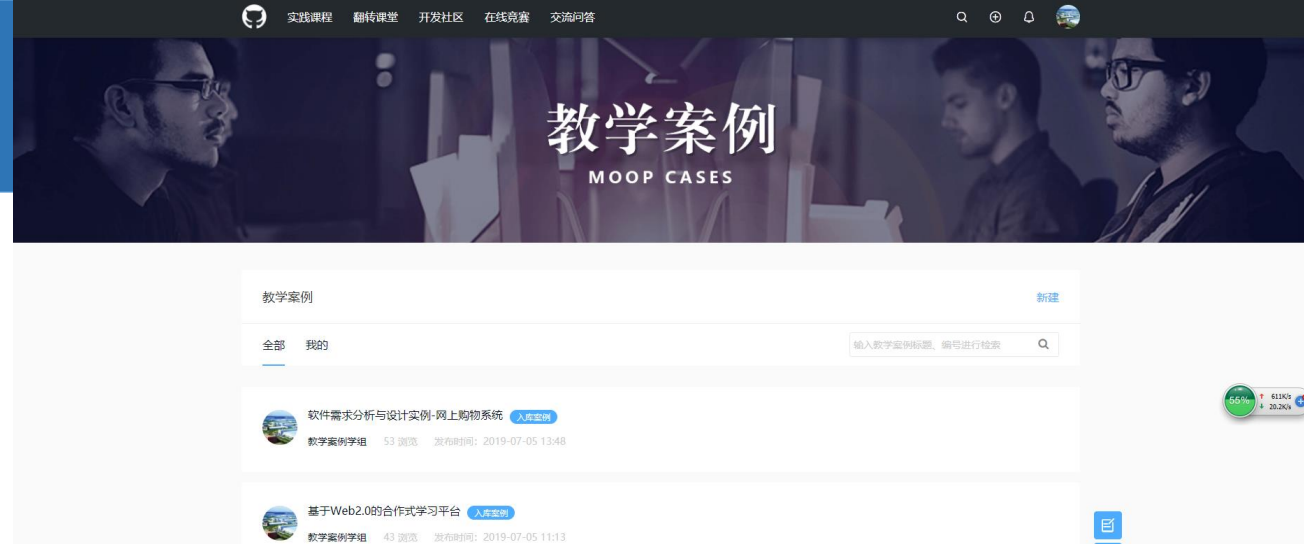
案例征集面向全国高校，征集可用于本科生和研究生的各类软件工程教学案例。提交案例要符合基本的入库要求，入库标准参见附录一。提交时还需包括案例使用说明书，使用说明见附录二。此外，参与者应确保提交的教学案例不涉及任何知识产权纠纷。

征集的教学案例内容和形式包括但不限于以下方面：

- 1.完整项目式教学案例：一般用于综合实践课程教学，案例内容覆盖某个工程实践项目的全部流程或主要阶段，系统地分析和探讨如何开展和实施软件工程实践活动。
- 2.知识点教学案例：一般用于辅助理论课程的某个或某些知识点的教学，以实际需求、现实问题切入，形象地讲解和分析软件工程相关理论知识的技术，启发学生思考。
- 3.检验型教学案例：一般用于测试或验证学生对理论知识和实践能力的掌握情况，可服务于课程测验、考试和竞赛等。
- 4.案例大小不限；案例形式可以文字为主、视频为主或者是两者相结合的综合性案例。

未来的考虑

- 案例库更加开放
- 除了推荐的案例库外，可以上传案例。
- 拟参考百度文库的做法，免费浏览，有偿使用（积分？）
 - 获奖案例奖励积分
 - 根据贡献度，奖励积分
- 以后的案例比赛，除了专家评，还可以根据访问量等进行推荐；也可以考虑网络打分相结合的方式



谢谢，