

同济大学课程考核试卷 (A 卷)  
2022 — 2023 学年第 一 学期

审核教师签名:

考试考查：考试

此卷选为：期中考试( )、期终考试( ☒ )、重考( )试卷

年级\_专业\_学号\_姓名\_得分\_\_\_\_\_

注：所有的回答请用英文，填写在题目对应 Answer 处。

## General Format Questions (1-12)

**1.** The following table is from Appendix C of the textbook for the following questions. (8 points)

Op- code	Operand	Description
1	RXY	LOAD the register R with the bit pattern found in the memory cell whose address is XY.
2	RXY	LOAD the register R with the bit pattern XY.
3	RXY	STORE the bit pattern found in register R in the memory cell whose address is XY.
4	0RS	COPY the bit pattern found in register R to register S.

Encode each of the following commands in terms of the machine language described in the language description table.

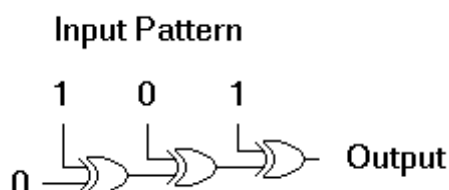
- A. 27A5      LOAD register 7 with the value A5.  
B. 17A5      LOAD register 7 with the contents of the memory cell at address A5.  
C.    2A00      places 00000000 in register A

Decode each of the following instructions that were encoded using the language description table.

- D. 4034 COPY the bit pattern found in register 3 to register 4.

**ANSWERS:**

2.a. What is the output of the circuit below? (1 point)



**ANSWER: 0**

- b. Translate each of the following binary representations into its equivalent base ten representation.

(2points)

1) 1100 \_\_\_ 12 \_\_\_      2) 10.011 \_\_\_ 2.375 \_\_\_

ANSWER:

c. Rewrite each of the following values (represented in base ten notation) in binary notation. (2points)

3) 23 \_\_\_ 10111 \_\_\_      4)  $2\frac{1}{4}$  \_\_\_ 10.01 \_\_\_

ANSWER:

d. Based on the 8-bit representation, write the two's complement representation of 5 and -5. (2points)

ANSWER:

- 5 的补码表示是 00000101。
- -5 的补码表示是 11111011。

### 3. What conditions are necessary for deadlock to occur? (9 points)

ANSWER:

1. 资源是按需分配的

2. 一旦分配，资源不能被强制回收

3. 存在对不可共享资源的竞争

### 4. Fill in the blanks below with the part on the operating system (file manager, memory manager, device drivers, window manager, scheduler, dispatcher) that performs the activity described. (10 points)

- A. \_\_\_ dispatcher 分派器 \_\_\_ Executes each time a time slice terminates  
\_\_\_ dispatcher \_\_\_ Performs the switching from one process to another
- B. \_\_\_ memory manager \_\_\_ Creates virtual memory
- C. \_\_\_ scheduler 调度器 \_\_\_ Places new entries in the process table  
\_\_\_ scheduler \_\_\_ Removes entries from the process table
- D. \_\_\_ device drivers \_\_\_ Performs the actual communication with I/O units
- E. \_\_\_ file manager \_\_\_ Protects files from unauthorized access  
\_\_\_ file manager \_\_\_ Maintains a record of available mass storage space  
(维护可用大容量存储空间的记录)

F. \_\_\_\_\_ **Window manager** \_\_\_\_\_ Maintains a record of what is displayed on the **computer's screen**

**ANSWERS:**

**A. Dispatcher (分派器)**

- 分派器负责在**时间片结束时**将 CPU 从一个进程切换到另一个进程。

**B. Memory Manager (内存管理器)**

- 内存管理器负责创建虚拟内存以支持多任务和大内存需求。

**C. Scheduler (调度器)**

- 调度器负责将新进程的信息放入**进程表中**以准备调度。

**D. Device Drivers (设备驱动程序)**

- 设备驱动程序负责与 I/O 设备进行实际的低级通信。

**E. File Manager (文件管理器)**

- 文件管理器负责控制对文件的访问，防止未经授权的访问或修改。

**5. Describe and explain each step of a machine cycle for an instruction execution.**

Which layer of the four-layer(application,transport,network,link) network model actually transmits a message? (8 points)

**ANSWER:**

**1. 题目翻译**

描述并解释**指令执行中****机器周期**的每一步。

**四层网络模型**中，哪一层实际传输消息？

---

**2. 答案**

机器周期的步骤：

**Fetch an instruction and increment the program counter,**

**decode the instruction,**

**execute the instruction.**

取指令并**递增程序计数器**，**解码指令**，并**执行指令**。

**四层网络模型中实际传输消息的层：**

在**四层网络模型**（通常指 **TCP/IP 模型**）中，**(link Layer)** 负责实际传输消息。

**6. Describe the respective functions of repeater, bridge, hub, switch and router. (10 points)**

**ANSWER:**

• **题目翻译**

描述以下设备的各自功能：**中继器 (repeater)**、**网桥 (bridge)**、**集线器 (hub)**、**交换机 (switch)** 和 **路由器 (router)**。

---

• **答案**

**中继器和网桥**都用于连接两条总线，

但**中继器**传输所有消息，而**网桥**只转发那些目标是另一侧的消息

集线器只是一条中央（短）总线，计算机连接到该总线以形成总线网络。

交换机连接多个总线网络以形成更大的网络。

路由器连接两个网络，形成一个 **Internet**，其中原来的两个网络继续作为独立的网络运行。

**7.** What are two protocols for implementing the transport level in the four-layer network model (6 points)

A. \_\_\_\_\_ B. \_\_\_\_\_

Which one is a reliable protocol in that the origin and destination work together to confirm that the entire message was successfully transferred.

C. \_\_\_\_\_

- 题目翻译

在四层网络模型中，实施传输层的两种协议是什么？（6分）

A. \_\_\_\_\_

B. \_\_\_\_\_

哪一个协议是可靠的协议，即源和目标之间相互配合，确认整个消息已成功传输？

C. \_\_\_\_\_

---

- 答案

**A. TCP（传输控制协议）**

- 解释：TCP 是一种面向连接的可靠传输协议，确保数据的可靠传输，并能处理丢包、错误检查、数据顺序、流量控制等问题。

**B. UDP（用户数据报协议）**

- 解释：UDP 是一种无连接、不可靠的传输协议，速度较快，但不保证数据的完整性和顺序。适用于对时延要求高、能容忍丢包的应用场景。

**C. TCP**

- 解释：**TCP 是可靠的协议**，因为它通过三次握手建立连接、通过确认应答机制确保每个数据包被成功接收，并在必要时重发丢失的数据包，从而确保整个消息的成功传输。

**8.** Please describe the difference between **a process, an algorithm, a program, a programming language and a pseudocode.** (10 points)

**ANSWER:**

- 题目翻译

请描述**进程、算法、程序、编程语言和伪代码**之间的区别。（10分）

---

- 答案

**1. 进程（Process）**

- 定义：进程是**计算机中正在执行的程序**的实例。它包括程序代码、程序计数器、堆

栈、数据段等资源，并且拥有独立的执行环境。

- **特点：**进程是程序在运行时的表现，每个进程有自己的内存空间和系统资源，通常由操作系统管理和调度。

## 2. 算法 (Algorithm)

- **定义：**算法是**解决问题的步骤与规则的集合**，通常是以清晰、有限的步骤来描述的，可以通过计算机程序实现。
- **特点：**算法具有明确的输入、输出、步骤及结束条件，解决问题的过程是确定的，并且应该是有限的、有效的和无歧义的。

## 3. 程序 (Program)

- **定义：**程序是**由一系列指令组成的，能够在计算机上执行某些特定任务的软件**。程序是代码的具体实现，可以用编程语言编写的。
- **特点：**程序通过编写特定的代码，按照设定的规则控制计算机硬件来完成特定的任务。程序需要经过编译（对于编译型语言）或解释执行（对于解释型语言）。

## 4. 编程语言 (Programming Language)

- **定义：**编程语言是**用于编写程序的语言**，提供了一组语法规则和语义定义，允许开发者以某种结构化和抽象的方式描述计算过程。
- **特点：**编程语言如 C++、Python、Java 等，具有不同的特性和用途，通常包括变量、控制结构、数据类型等，帮助开发人员实现算法并开发软件应用。

## 5. 伪代码 (Pseudocode)

- **定义：**伪代码是一种非正式的、**接近自然语言的编程描述方式**，用于描述算法的逻辑步骤，而不关注具体编程语言的语法细节。
- **特点：**伪代码通常简单易懂，能够清楚地展示算法的思路，既可以被程序员理解，也能为实际的程序开发提供指导。它不依赖于任何编程语言的语法规则，主要用于设计和交流算法。

---

### • 总结

**进程：**正在执行的程序实例，具有独立资源和环境。

**算法：**解决问题的步骤或规则集合，描述解决问题的具体方法。

**程序：**由指令组成的可执行代码，完成特定任务。

**编程语言：**用于编写程序的语言，具备语法规则和语义定义。

**伪代码：**用自然语言接近的方式描述算法，不依赖具体编程语言的语法。

## 9. Explain the distinction between the imperative (命令式), declarative (声明式) and functional programming paradigms (范式).

Why a high-level programming language is machine independent? (8 points)

ANSWER:

### • 题目翻译

解释命令式编程、声明式编程和函数式编程范式之间的区别。

命令式范式要求程序员描述解决手头问题的算法。声明式范例要求程序员描述问题。

为什么高级编程语言是与机器无关的？（8分）

---

- 答案

### 命令式编程 (Imperative Programming)

- 定义: 命令式编程强调如何通过一系列明确的指令来改变程序的状态。这种编程范式通过顺序执行一组语句来完成任务。
- 特点: 命令式编程侧重于计算的控制流, 程序员需要详细描述每个步骤和状态的变化。常见的命令式编程语言包括 C、Java 和 Python。
- 例子: 使用循环、条件语句等来改变程序状态的过程, 例如:  $x = x + 1$ 。

### 声明式编程 (Declarative Programming)

- 定义: 声明式编程关注做什么而不是如何做。程序员描述的是想要达成的结果, 而非如何一步步实现。
- 特点: 声明式编程避免了具体的控制流和状态变化的描述, 程序员只需指定问题的定义和要求。SQL 和 HTML 是典型的声明式编程语言。
- 例子: 在 SQL 中, `SELECT * FROM users WHERE age > 30`, 这条命令描述了我们想要的结果, 但没有指定如何获取数据。

### 函数式编程 (Functional Programming)

- 定义: 函数式编程是一种特殊类型的声明式编程, 强调使用数学函数来进行计算。它的核心概念是函数的“无副作用”和“不可变性”。
- 特点: 函数式编程注重函数的组合与递归, 而非改变程序的状态。变量在函数式编程中通常是不可变的, 不会在执行过程中发生改变。常见的函数式编程语言有 Haskell 和 Lisp。
- 例子: `map`、`filter` 和 `reduce` 函数在函数式编程中非常常见, 它们操作数据并返回新结果, 而不会改变原数据。

---

### 为什么高级编程语言是与机器无关的?

这意味着用该语言编写的程序不引用特定计算机属性, 因此与任何计算机兼容。

**10.** What are formal and actual parameters? When a function passes parameters, what is the difference between the call-by-reference and call-by-value modes? (8 points)

ANSWER:

- 题目翻译

什么是形式参数 (formal parameters) 和实际参数 (actual parameters)? 当一个函数传递参数时, 按引用传递 (call-by-reference) 和按值传递 (call-by-value) 模式有什么区别? (8 分)

---

- 答案

形式参数 (Formal Parameters)

- **定义：**形式参数是函数声明时定义的参数，它们只是占位符，表示函数在调用时需要传入的值。形式参数在函数内部使用，并作为函数的输入。
- **特点：**形式参数的名称是由开发者定义的，函数体内的变量将通过这些形式参数来接收传递进来的实际值。
- **例子：**
  - `void func(int x, int y) { // x, y 是形式参数`
  - `// 使用 x 和 y 进行操作`
  - `}`

### 实际参数 (Actual Parameters)

- **定义：**实际参数是函数调用时传递给形式参数的具体值或表达式。它们可以是常量、变量、甚至是表达式的结果。
  - **特点：**实际参数提供给函数的输入值，在函数调用时由调用方提供。
  - **例子：**
    - `int a = 5, b = 10;`
    - `func(a, b); // a 和 b 是实际参数`
- 

### 按值传递 (Call-by-Value)

- **定义：**在按值传递模式下，函数接收到的是实际参数的副本，而不是原始值。函数内部对参数的修改不会影响到调用函数中的变量。
- **特点：**参数传递时，实际参数的值被复制到形式参数，函数内部对形式参数的更改不会影响实际参数。适用于不需要修改原始数据的情况。
- **例子：**
  - `void func(int x) {`
  - `x = 10; // 修改的是形式参数`
  - `}`
  - `int a = 5;`
  - `func(a);`
  - `// 这里 a 仍然是 5`

### 按引用传递 (Call-by-Reference)

- **定义：**在按引用传递模式下，函数接收到的是实际参数的引用（地址），即对实际参数的直接访问。函数内部对参数的修改会影响到调用函数中的变量。
- **特点：**参数传递时，传递的是变量的内存地址，函数内部对参数的更改会直接影响到实际参数。适用于需要修改原始数据或传递大型数据结构的情况。
- **例子：**
  - `void func(int &x) {`

- `x = 10; // 修改的是实际参数`
  - `}`
  - 
  - `int a = 5;`
  - `func(a);`
  - `// 这里 a 的值变为 10`
- 

## • 总结

**形式参数：**函数声明时定义的占位符，用于接收实际参数的值。

**实际参数：**函数调用时传递的实际值，可以是常量、变量或表达式。

**按值传递：**函数接收的是实际参数的副本，函数内部修改不会影响实际参数。

**按引用传递：**函数接收的是实际参数的引用，函数内部修改会直接影响实际参数。

**11.** Explain the distinction between translating a program (in a high-level language) and interpreting **解释** the program.

What are the main 3 activities in the process of translating a program? Please describe their roles briefly. (8 points)

**ANSWER:**

翻译程序与解释程序的区别；

**翻译一个程序就是**在不执行它的情况下将其转换为另一种（通常是低级的）语言。

**解释程序就是**直接从其高级语言形式执行它。

---

翻译程序的三项主要活动及其角色

1. **词法分析 (Lexical Analysis):**

- **角色：**词法分析的主要任务是**将源代码中的字符序列转换为一系列的记号**



(tokens)。每个记号代表源代码中的一个语法元素，例如关键字、变量、操作符等。

- **过程:** 该阶段会扫描整个源代码并将其分割成有意义的词法单元，例如将 `int x = 10;` 转换为 `int, x, =, 10, ;` 等记号。

## 2. 语法分析 (Syntax Analysis):

- **角色:** 语法分析的任务是根据源代码的语法规则，检查源代码是否符合语言的语法结构。通过构建语法树，验证程序中各个语句的结构是否正确。
- **过程:** 在这一阶段，编译器将词法分析得到的记号按语法规则进行排列和组合，确保程序没有语法错误。例如，`int x = 10` 这样的语法就应该符合“类型+变量+赋值”的结构。

## 3. 语义分析 (Semantic Analysis):

- **角色:** 语义分析检查程序中的语义错误，确保程序的含义符合语言的规则，处理程序中变量、类型等的正确性。
- **过程:** 在这一阶段，编译器检查符号表、数据类型等方面的错误，例如，确保变量在使用前已经声明，或者变量类型在赋值时是否匹配。

---

### 总结

- **翻译程序:** 将源代码一次性转换为机器语言或中间代码，生成可执行文件，程序执行较快。
- **解释程序:** 逐行解释执行源代码，不生成独立的可执行文件，执行速度较慢。
- **翻译程序的两项活动:**
  1. **词法分析:** 将源代码转换为记号 (tokens)
  2. **语法分析:** 检查代码的语法结构，构建语法树
  3. **语义分析:** 检查代码中的语义错误，确保类型和变量等的正确性

**12.** What are the differences between list, stack, queue and tree? (8 points)

### ANSWER:

#### 1. 题目翻译

列出链表、栈、队列和树之间的区别。(8分)

---

#### 2. 答案

列表、堆栈、队列和树都是常见的数据结构，它们各自有不同的特点和用途。下面是它们之间的主要区别：

- **1. 列表 (List)**

**定义:** 列表是一种线性数据结构，用于存储一组元素，元素在内存中可以按顺序排列。列表中的元素可以是不同类型的，可以通过索引访问。

**访问方式:** 支持随机访问，可以通过索引访问任何元素。

**插入与删除:** 可以在任意位置插入或删除元素。

**特点:**

- 元素的顺序是有意义的。
- 大多数列表是动态大小的，意味着它们的大小可以变化。
- 可支持重复元素。

应用：用于存储和管理一组元素，常见于实现其他数据结构（如堆栈、队列）时。

• 2. 堆栈 (Stack)

定义：堆栈是一种后进先出（**LIFO**, Last In, First Out）的数据结构。堆栈只能在一端进行操作，这一端叫做栈顶。

访问方式：只能访问栈顶的元素。

插入与删除：只能在栈顶插入元素或从栈顶删除元素（即推入和弹出操作）。

特点：

- 数据是有序的，但只能通过栈顶访问和操作。
- 遵循“后进先出”原则。

应用：常用于函数调用的管理（递归调用栈）、浏览器历史记录的回退、撤销操作等。

• 3. 队列 (Queue)

定义：队列是一种先进先出（**FIFO**, First In, First Out）的数据结构。队列支持从一端（队尾）插入元素，从另一端（队头）删除元素。

访问方式：只能访问队头的元素。

插入与删除：元素从队尾插入，从队头删除。

特点：

- 数据的顺序也很重要，但遵循“先进先出”原则。
- 插入和删除的操作分别发生在队列的两端。

应用：用于任务调度、缓冲区管理（如操作系统的任务队列）、打印队列等。

• 4. 树 (Tree)

1. 定义：树是一种层次结构的数据结构，由节点和边组成。一个树由一个根节点开始，根节点通过边连接其他节点，形成一个层次结构。

2. 访问方式：通常不支持像列表那样的随机访问，访问元素时要通过遍历树的节点（如深度优先遍历、广度优先遍历）。

3. 插入与删除：节点可以根据树的类型（如二叉树、平衡树、红黑树等）进行插入和删除操作。

4. 特点：

- 树是层次性结构，每个节点可以有多个子节点。
- 常见的树类型有二叉树、AVL 树、红黑树、堆等。

5. 应用：常用于表示层次结构（如文件系统目录树），以及在算法中用来表示决策过程（如二叉搜索树用于查找、堆用于优先队列等）。

总结：

数据结构	访问方式	插入与删除操作	存储结构特点	典型应用
列表	随机访问，通过索引	在任意位置插入或删除元素	有序，元素可以是不同类型	存储和管理元素，通常用于实现其他数据结构
堆栈	只能访问栈顶元素	只能在栈顶插入 (push) 或删除 (pop) 元素	遵循“后进先出”原则	函数调用栈、撤销操作、表达式求值
队列	只能访问队头元素	只能从队头删除，队尾插入元素	遵循“先进先出”原则	任务调度、缓冲区管理、打印队列等
树	通过遍历访问	插入和删除根据树的结构进行 (如二叉树的左右子节点)	层次结构，节点可有多个子节点	层次结构表示、二叉搜索树、文件系统、决策树等

通过这些区别，可以更清楚地了解每种数据结构的特点和使用场景。

