

# 软件体系结构作业（三）

---

## KWIC还可以采用哪些风格？分析各自的架构特点和运行特点。

除了课程中已经讲过的主程序/子程序风格，我认为KWIC可以采用如下六种架构风格实现：

### 批处理风格

#### 架构特点：

将KWIC系统划分为离散的处理批次。每个批次包含一组文本，可能来自不同的来源，需要经过一系列的处理步骤。每个批次都会经历相同的处理流程，包括分词、预处理和索引生成。每个处理步骤可以被视为一个批次处理器，负责特定的任务。

#### 运行特点：

批处理风格适合用于离线处理或大规模文本处理，因为它允许系统在不需要实时响应的情况下高效地处理大量文本数据。这种风格在文本处理的各个阶段之间建立清晰的分隔，使得每个阶段可以并行执行。优点在于高吞吐量、数据一致性；局限性在于不适合实时处理且可能需要大量存储。

### 管道/过滤器风格

#### 架构特点：

管道/过滤器风格适合将处理流程分解为一系列连续的过滤器或处理步骤。在KWIC系统中，每个过滤器可以代表不同的文本处理任务，如分词、预处理和索引生成。

#### 运行特点：

管道/过滤器风格可实现高度可组合性，使得添加、删除或更改过滤器变得相对容易。每个过滤器以流水线方式处理数据，这可以提高效率。优点在于模块化和可组合性、易于调试和测试等；局限性在于限制了处理顺序且数据转换开销可能较大。

### 层次结构风格

#### 架构特点：

KWIC系统的组件按照功能或责任领域分为不同的层次。每个层次在系统中扮演特定的抽象角色，它们通过明确定义的接口进行交互。较高层次的抽象通常建立在较低层次的抽象之上。各个层次之间的通信通常是通过标准接口进行的，这降低了各层次之间的依赖性，使系统更容易维护和扩展。

### 运行特点：

层次结构的一个主要优点是它将不同关注点分离开来。数据存储层负责数据的存储和检索，业务逻辑层负责处理文本处理和索引生成，用户界面层负责呈现结果和与用户交互。因此整个系统具有较高的可维护性和可扩展性。但每个层次之间的数据传递和通信可能会导致性能开销，且大量的接口依赖可能需要较多精力进行接口管理。

## 仓库风格

### 架构特点：

仓库风格将KWIC系统的文本数据集中存储在一个中心化的数据仓库中，允许用户查询并获取KWIC索引。

### 运行特点：

这种架构有助于数据的集中管理和维护。用户可以使用查询系统来检索索引，这种架构适用于需要大量存储和检索的情况。仓库风格的优点在于数据被集中管理且支持复杂查询；缺点在于如果数据仓库出现故障，整个系统可能会受到影响，因此需要实施高可用性和容错机制。

## 进程通讯风格

### 架构特点：

进程通讯风格适用于将KWIC系统拆分为多个独立的进程或组件，每个进程负责不同的任务，如文本处理、索引生成和查询处理。

### 运行特点：

进程通讯风格架构提高了系统的可扩展性和模块化。不同进程之间可以使用通信协议进行数据交换。进程通讯风格的主要优点在于支持分布式处理，并行与并发且实现了松耦合。但这也带来了问题：进程通讯需要在不同进程之间传递数据和消息，这可能会引入通信开销和延迟；同时系统的复杂性显著提高。

## 事件系统风格

### 架构特点：

事件系统风格允许不同模块以松耦合的方式协作，模块可以发布和订阅事件。在KWIC系统中，各个模块可以响应事件以执行其任务。

### 运行特点：

这种风格可以提高系统的灵活性，模块可以相对独立地响应事件，无需直接依赖其他模块的内部实现。事件系统主要也实现了松耦合与模块化，但由于事件系统的事件流通常是异步的，因此当事件流变得复杂时，可能使系统的流程难以追踪和调试。