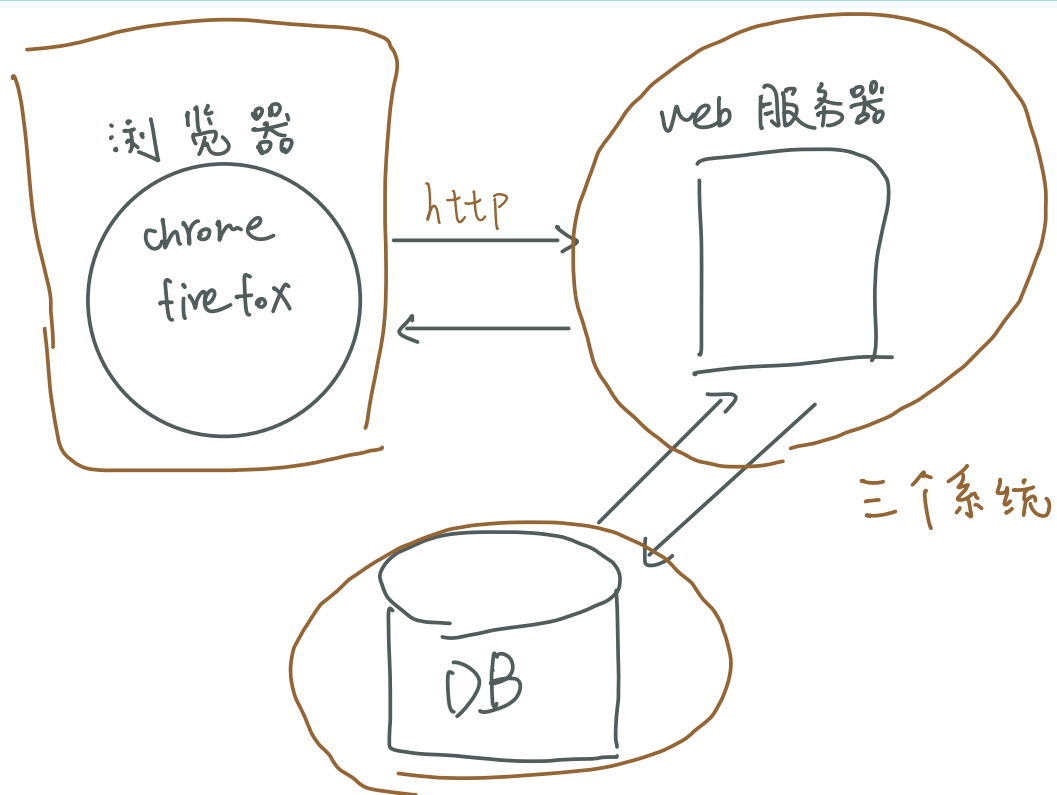


1.1 什么是系统？是组成软件的一个独立的模块（可以独立运行自成一体）



1.2 系统的特性

- 1、通用性（http, 浏览器, DB 等（区别于你自己写的只有你会用））
- 2、隔离性（更简单更容易维护）API 接口
- 3、精巧性（性能 尽善尽美 效率）

什么是数据管理系统? 1.3

一个模块实现数据管理的功能，就是数据管理系统。

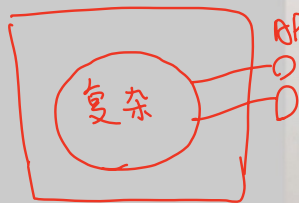
1. 提供数据存放的功能 (不丢失, 稳妥)
2. 组织数据的功能, 当我需要数据的时候我能很快找到我要的数据
3. 确保数据准确无误, 保证正确性
4. 提供数据处理的平台, 统计某个用户余额够不够, 买了多少东西

(在DB中处理, 把result给web服务器, 而不是在Web上处理)

1.5 程序的模块化设计

隔离性 / 封装性 (少依赖别的模块)

☆ 信息隐藏原则



KWIC 索引问题

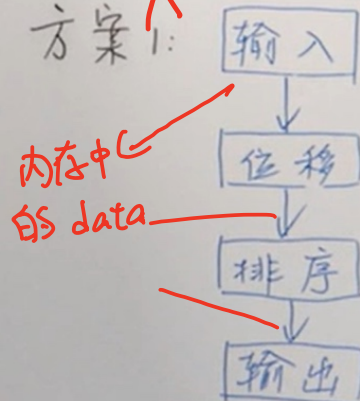
David. L. Parnas
1972

输入: 一张表 每一行为一个单词序列

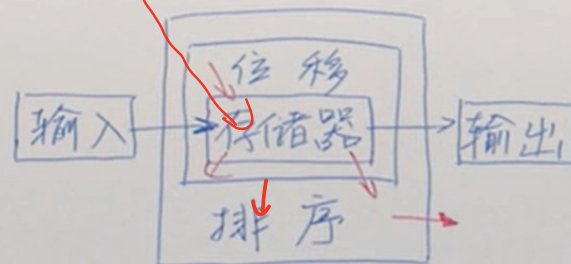
输出: 对每一行实施“位移”,
并将字母顺序输出所有“位移”

位移: ^{第一个单词移到最后}
"A good day" → "good day A"
"day A good"

方案1: ✗



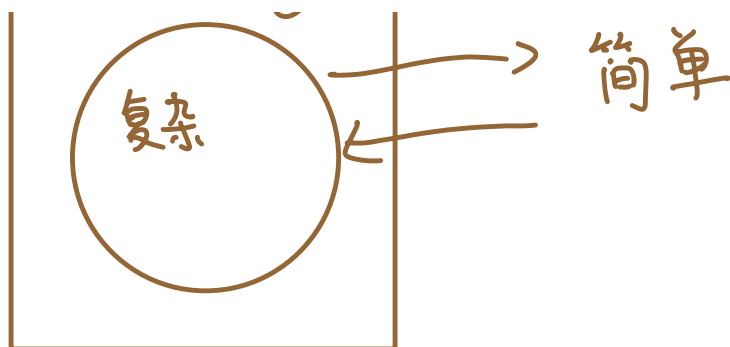
方案2: ✓



RDBMS 关系数据库 & NOSQL 之一的文档数据库

api





姓名: 杨西雅
学号: 10215501435
专业: 数据科学与大数据技术
ip:

监控已禁用

选择题

1 2 3

下一题

查看结果 重新练习 返回

练习模式

单选题

第 1 题(本题2分): 如课程中提到, 由于大部分应用程序都需要数据管理功能, 因此人们将数据管理功能抽象出来, 形成可独立运行的系统。然而并非所有的功能抽象都有必要成为独立的系统。比如“排序”这一功能, 只需要构造成一个库函数, 供各种程序调用即可。那么, 为什么数据管理功能需要被构造造成独立运行的系统? 以下那种说法不成立?

☒ A: 数据管理功能很复杂, 复杂的功能需要成为独立的系统。✓

☐ B: 数据管理需要消耗大量资源 (包括CPU、内存、硬盘), 独立系统可实现独立的资源调配, 有利于资源的高效利用。✗

☐ C: 同一份数据有时候需要被多个应用程序共享, 独立的系统有利于共享。✗

☐ D: 作为独立系统的一个好处是: 当应用程序出错时, 不容易牵连数据管理系统, 造成数据损坏。✗

题千字体大小

小 中 大

当要改动内存中的数据的结构时, 方案一的每个模块都要改动, 但是方案二只需要改动存储



数据管理功能之所以需要构建成独立运行的系统，以数据库管理系统 (DBMS) 为例，有以下几个关键原因：

1. **数据共享和一致性：** 数据库管理系统可以支持多个应用程序同时访问和共享数据，而且可以确保数据的一致性。这是因为数据库系统通过事务管理来确保数据的完整性，避免了数据被多个应用程序同时修改而导致的问题。
2. **数据安全性：** 数据库管理系统提供了对数据的安全性管理机制，包括用户认证、授权、加密等，以确保只有授权的用户可以访问和修改数据。这对于敏感信息和业务数据非常重要。
3. **性能优化：** 数据库系统经过优化，可以处理大量数据并提供高效的查询和检索功能。这种性能优化包括索引、查询优化、并发控制等功能，可以提高数据访问速度。
4. **数据备份和恢复：** 数据库系统可以自动进行数据备份和恢复操作，以防止数据丢失或损坏。这对于数据的长期保存和业务连续性非常关键。
5. **数据模型和查询语言：** 数据库系统提供了丰富的数据模型和查询语言，使用户能够以更抽象的方式管理和查询数据，而不需要深入了解数据存储细节。
6. **数据一致性维护：** 数据库系统支持事务处理，确保数据的一致性和完整性。这是在多个应用程序同时访问和修改数据时非常重要的功能。

关于“数据管理功能很复杂，复杂的功能需要成为独立的系统”这种说法为什么是错的，实际上并非所有复杂的功能都需要成为独立的系统。决定是否将某个功能构建成独立的系统通常取决于以下因素：

1. **复杂性和需求：** 复杂性是一个考虑因素，但不是唯一的决定因素。如果某个功能非常复杂且需要在多个应用程序之间共享，那么构建成独立的系统可能是合理的。然而，如果功能相对简单且只在一个应用程序内部使用，那么将其构建成库函数或模块可能更合适。
2. **重用性：** 如果某个功能有潜在的重用价值，即其他应用程序可能也会需要类似的功能，那么将其构建成库函数或独立系统可以提高代码重用性，减少开发工作量。
3. **维护和扩展：** 考虑将功能构建成独立系统还需要考虑维护和扩展的难易程度。如果功能需要频繁更新和扩展，将其构建成独立系统可以更容易地进行维护和升级。

综上所述，决定将数据管理功能构建成独立系统的关键在于需求和复杂性，而不是简单地基于复杂性而做出的决策。不同的功能可以采用不同的方法来实现，以满足特定的业务需求。



信息隐藏

思考题：请为以下程序做模块设计

初始输入：若干文件，每个文件是一篇文章，由若干英文单词组成。例如：file1.txt, file2.txt, file5.txt, ...

调用输入：初始输入中的一个文件的文件名, 例如：file2.txt;

调用输出：从初始输入的若干文件中，找到和调用输入文件最相似的文件（除了它自己），并输出这个文件的文件名，例如：file5.txt。

注意：程序未来可能需要调整，比如，相似度的计算方法可能会调整，又如，调用输入可能允许输入一个新文件，再如，调用输出可能允许输出多个相似的文件，等等。



为了设计这个程序的模块，你可以考虑将它划分为以下几个模块，每个模块负责不同的功能。这样可以使程序更加模块化、易于维护和扩展：

1. **文件读取模块：**

- 负责读取初始输入的文件内容。
- 可以将读取的文本数据进行预处理，例如分词、去除停用词、转换为小写等操作，以便后续的相似度计算。

2. **相似度计算模块：**

- 用于计算文件之间的相似度。
- 可以选择合适的相似度度量方法，如余弦相似度、Jaccard相似度、编辑距离等，具体根据需求进行选择。
- 可以考虑将相似度计算模块设计为可扩展的，以便未来可以更改相似度计算方法。

3. **查询处理模块：**

- 负责接收调用输入，即指定要查询相似文件的文件名。
- 调用相似度计算模块来计算每个文件与查询文件的相似度。
- 根据相似度计算结果，选择最相似的文件或根据需要返回多个相似文件。

4. **输出模块：**

- 用于输出查询结果，即找到的最相似的文件名或多个相似文件名。
- 可以根据程序未来可能的调整，灵活处理输出需求。

5. **主程序模块：**

- 将以上模块组合在一起，实现整体的程序逻辑。
- 负责管理文件的读取、相似度计算、查询处理和输出。

6. **配置模块：**

- 用于管理程序的配置，例如相似度计算方法、文件路径、停用词列表等配置项。
- 可以使程序更加灵活，方便根据需要进行调整。

7. **扩展性模块：**

- 设计程序以便轻松扩展功能。例如，允许输入新文件、允许输出多个相似文件等。
- 考虑将功能扩展与核心逻辑解耦，以便未来扩展不会影响程序的稳定性和可维护性。

以上模块划分是一个初步的设计，具体实现可以根据编程语言和框架的选择来进行调整和细化。此外，确保在程序设计中考考虑到异常处理、性能优化和用户友好的界面设计等因素，以实现一个功能强大且易于使用的数据管理系统。