

第3章 软件体系结构风格

1. 上机调试管道-过滤器风格单词排序程序，并回答以下问题：

(1) 为什么数据流风格构件设计具有良好的隐蔽性和高内聚、低耦合的特点？

隐蔽性：

用户在使用该单词排序系统的时候只能看到最开始的构件接口，即将实现每个功能的代码隐蔽或风转载一个单一的模块中，更深层的连接件和接口是不会被用户显示调用的，用户也不会知道内部的调用是什么构件。并且多个接口的封装也隐藏了功能实现的具体细节。

高内聚：

数据流风格中的模块

- ① 只对外暴露最小限度的接口，形成最低的依赖关系
- ② 只要接口的输入输出格式不变，模块内部的修改，就不会影响其他模块。
- ③ 每个模块只专注于一种业务，模块的功能分化单一，使得该模块被其他模块调用的机会减少。

低耦合：

过滤器和其他构件以一种有限的方式进行交互，链接的紧密程度低，即使不需要该系统中的某特定操作，也能正常使用过滤器的功能。

(2) 为什么说数据流风格系统支持良好的软件复用特性？

软件复用是将已有软件用于建立新的软件，以缩减软件开发和维护的花费。在数据流风格中，由于各个构件的高内聚低耦合特点，过滤器之间没有强逻辑关系，使得构件的独立性大大增强，对原来系统的依赖性减弱，因此在别的项目中，可以拿来直接投入工程使用。

(3) 为什么说数据流风格系统可维护性较好？

由于数据流风格的高内聚低耦合，使得对每个构件修改时的代价就不会那么高，不会出现“牵一发而动全身”的情况，只要构件的输入输出格式不变，单纯对模块进行修改，不会对整个系统的正常运行造成影响，就算影响也只会影响少量模块，提高了系统的可维护性。

(4) 数据流风格适用于哪些应用?

各种需要组件之间相互调用的系统, 如翻译软件, ATM 存取系统, 压缩程序, 地图路线查询, 音频处理软件等等。他们之间都是用户通过简单接口调用, 输入一些数据, 然后软件内部将数据通过各种管道和构件进行处理, 最终将结果展现出来。

2. 常见的进程间通信方式有哪些? 列举至少三种通信方式并分析其优缺点。

① 管道通信

允许没有直接关系的进程之间通信, 分为对指定进程之间单项通信的无名管道和对任意进程

优点: 结构简单, 方便易用

缺点: 由于其易于使用, 很容易由于使用不当导致出错。

② 共享内存

共享内存就是映射一段能被其他进程所访问的内存, 这段共享内存由一个进程创建, 但多个进程都可以访问. 共享内存是最快的 IPC(进程间通信)方式, 它是针对其它进程间通信方式运行效率低而专门设计的. 它往往与其他通信机制, 如信号量, 配合使用, 来实现进程间的同步与通信.

优点: 针对消息缓冲的缺点改而利用内存缓冲区直接交换信息, 无须复制, 快捷、信息量大是其优点

缺点: 共享内存的通信方式是通过将共享的内存缓冲区直接附加到进程的虚拟地址空间中来实现的. 因此, 这些进程之间的读写操作的同步问题操作系统无法实现. 必须由各进程利用其他同步工具解决. 另外, 由于内存实体存在于计算机系统中. 所以只能由处于同一个计算机系统中的诸进程共享, 不方便网络通信。

③ 信号通信

信号是一种比较复杂的通信方式, 用于通知接收进程某个事件已经发生.

优点: 当用户需要传递的信号少, 或者需要通过信号来出发某些行为, 可以考虑使用信号通信

缺点: 不适合处理传递的信息量比较大或者进程间存在交换数据的通信需求。