王志辰-111161-20161001059

3-1

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）为什么数据流风格中构件设计具有良好的隐蔽性和高内聚、低耦合的特点？

（2）为什么说数据流风格系统支持良好的软件复用特性？

（3）为什么说数据流风格系统可维护性较好？

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。

3. 基于 AWS SQS（消息队列）实现一个简单的进程间通信程序，并结合程序分析基于消息 队列的进程通信方式的特点

1.（1）每个构件都有一组输入和输出，构件读输入的数据流，经过内部处理，然后产生输出数据流。构件之间看不见其他的内部操作，所以有很好的隐蔽性，构件之间的交流只依靠数据流，所以构件之间的耦合程度很低，但通过数据流之间的频繁交流，所以内聚程度很高。

（2）过滤器是独立的实体，它不会与其它的过滤器共享数据，当一个构件中需要的功能恰好是另一个构件的功能时，便可以在构件里调用另一个构件，就不用再去编写代码来实现相同的功能。

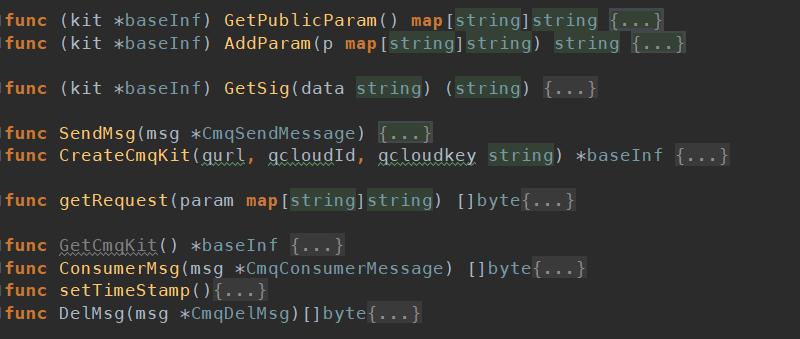
（3）新的过滤器可以很方便地添加到现有系统中来，替换掉旧的过滤器，每个构件都相对独立的存在，输入数据流和输出数据流是固定的，所以对一个构件进行维护只需要在这个构件的内部进行修改，不用在其他调用这个构件的部分进行修改。

2. 信号 ( sinal ) ： 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。

消息队列( message queue ) ： 消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

共享内存( shared memory ) ：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 方式，它是针对其他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，如信号两，配合使用，来实现进程间的同步和通信。

3. 消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。



使用单例模式，实现了cmq的go语言版SDK。