Software Architecture

Lecture 4-2. Dataflow Systems Pipes and Filters Architecture

管道过滤器架构

Professor: Yushan (Michael) Sun Fall 2020

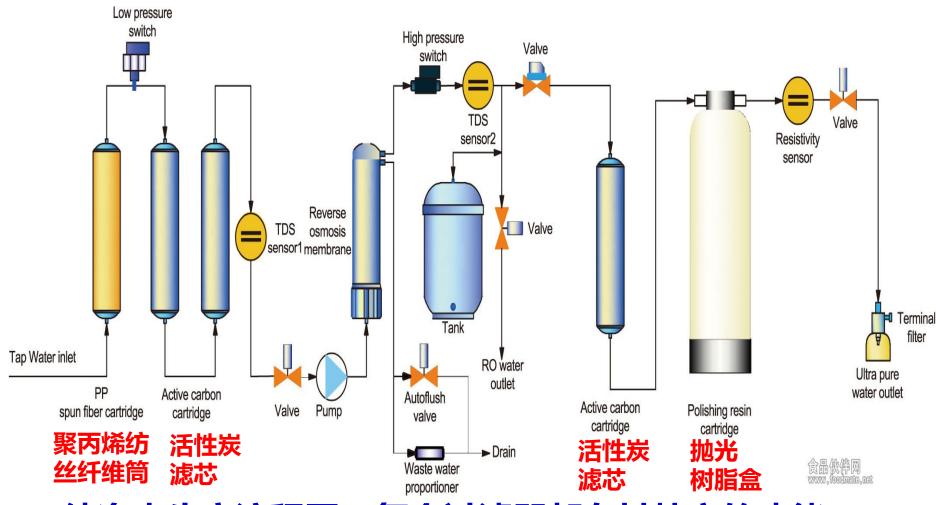
本讲内容

- 1. Concept of Pipes-and-filters architecture
- 2. Example programs designed in pipes and filters architecture
- 3. <u>Unix support for pipes and filters</u> architecture
- 4. Comparison of batch sequential architecture and pipes and filters architecture

Concept of Pipes and Filters architecture

管道-过滤器架构

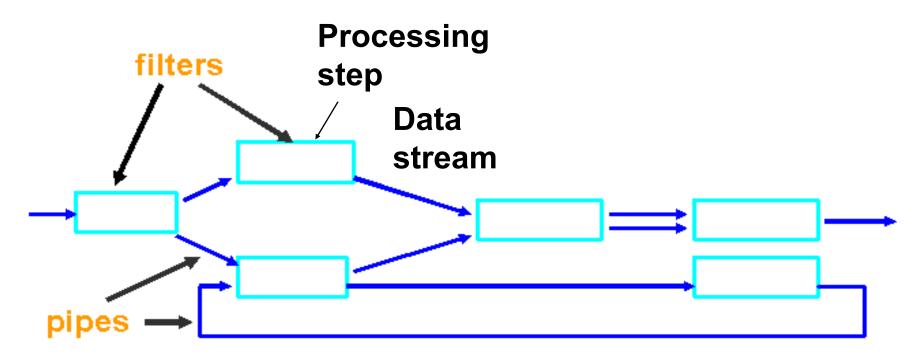
管道-过滤器软件体系结构



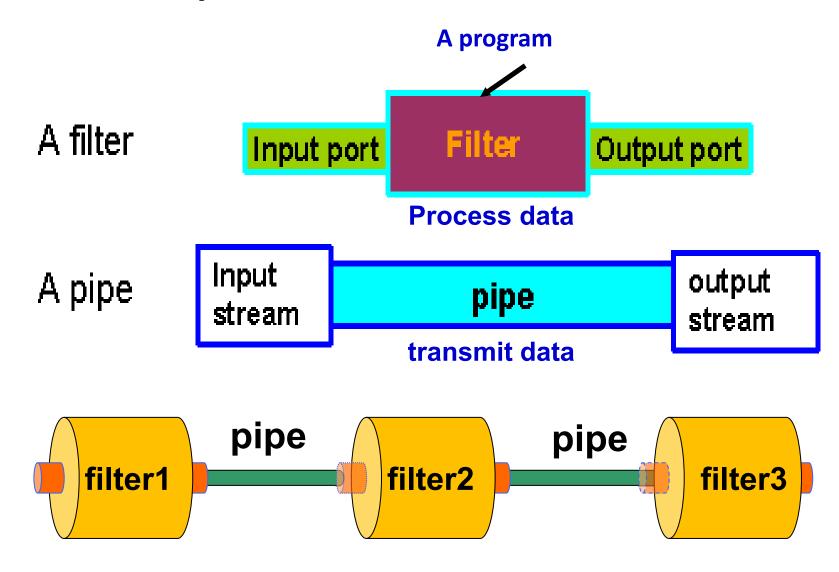
纯净水生产流程图:每个过滤器都有其特定的功能

管道-过滤器软件体系结构的定义:

Pipes and filters architecture is composed of filters to do data processing and pipes that carry data from one filter to the next filter.



Pipes and filters architecture



filter structure and pipe structure

过滤器的功能(Functionalities of Filters):

Incrementally transform some of the source data into sink data

- a) 添加信息. Enrich data by computing and adding information
- b) 变换数据. Transform data by changing representation
- c) 流对流变换. Stream to stream transformation
- d) 不保留状态. Preserve no state between instantiations (forget what has happened)

管道的功能(Functionalities of a Pipe): move data from a filter's output to a filter's input

- 单向流. One way flow
- 数据传输图. Pipes form data transmission graph

整体运行机制(Overall operations)

- Run pipes and filters until no more computations are possible
- Action is mediated by data delivery

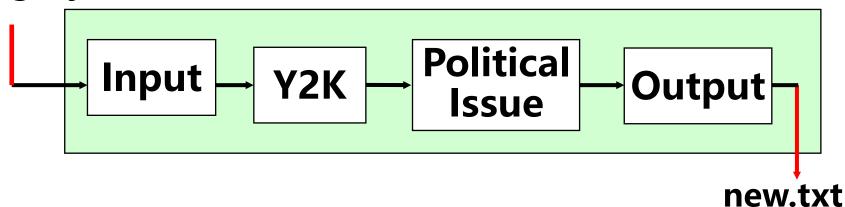


Example programs designed in pipes and filters architecture

例1. 干年虫与政治问题 (Year 2000 problem and political issues)

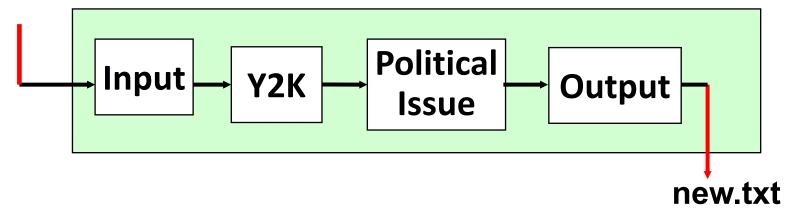
In a text file, change all year expressions "xx" into "19xx", e.g. "89" into "1989", and change all expressions "Republic of China" into "Taiwan"

legacy.txt



Legacy file update system designed using Pipes and filters architecture

legacy.txt

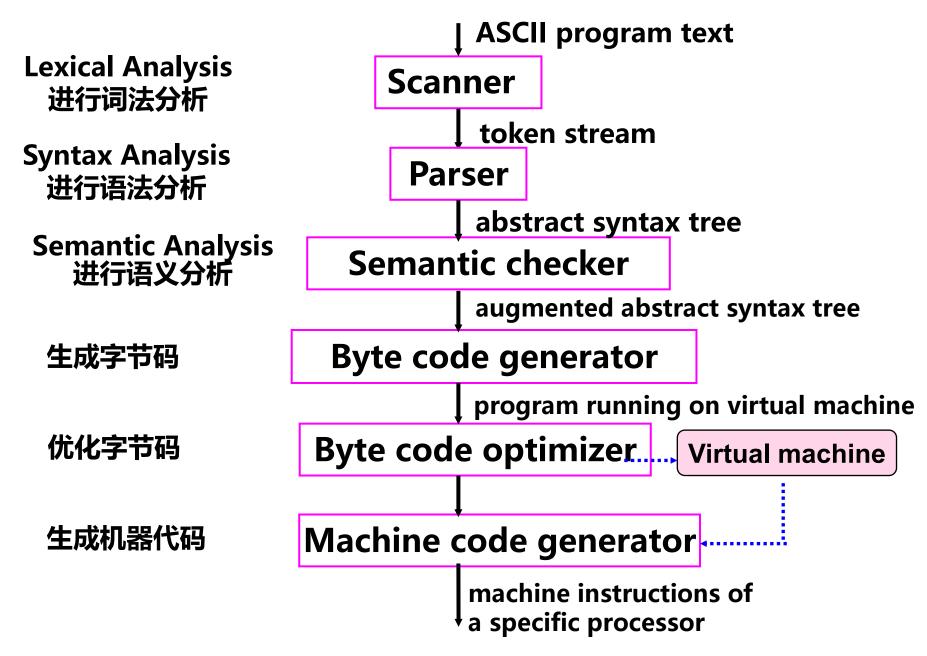


After Input: Television \$3800 Republic of China 95-12-11 After Y2K: Television \$3800 Republic of China 1995-12-11 After Political Issue: Television \$3800 Taiwan 1995-12-11 After Output: Television \$3800 China 1995-12-11

数据以数据流的方式从左端流入,流经各个过滤器,对数据进行处理;对于处理完的数据,立即写入到下游管道.

例2: 编译器(Compiler example)

A typical architecture of a compiler example is pipes and filters architecture describing in the next page.



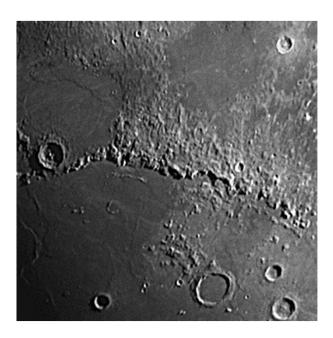
Functionalities of each filter

- Scanner: Lexical Analysis (词法分析)
 - Misspellings: such as "String" into "Sting"
- Parser: Syntax Analysis (语法分析)
 - Omission, wrong order of tokens
- · Semantic checker: Semantic Analysis (语义分析)
 - Incompatible types: e.g., "float", "double"
- Byte code generator (字节代码生成器): byte code generation:
 - C/C++ compiler: from source code to assembly code
 - Java compiler: from Java code to bytecode.
 - Java跨平台其实就是基于相同的bytecode规范做不同平台的虚拟机,我们的Java程序编译成bytecode后就可以在不同平台跑了。
- Byte code optimizer: optimizing byte code
- Machine code generator: generating machine code

例3:关于图像处理问题

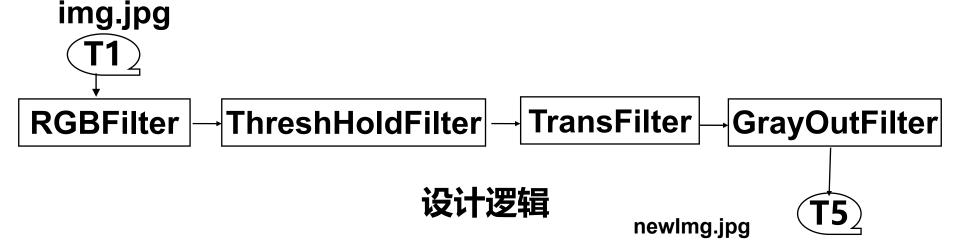
- Image processing problem
- Pipes and Filters architecture can be used to do some image processing to digital pictures (images)

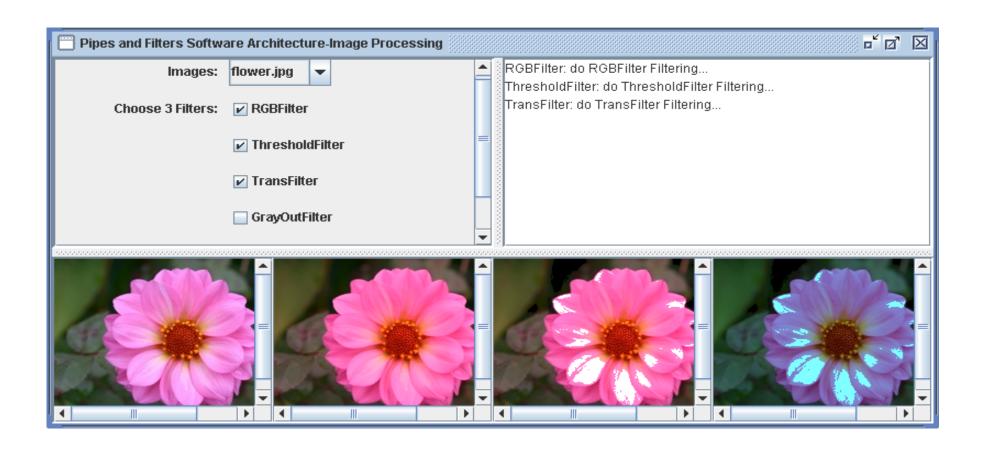




Picture before and after image sharpening

- · 使用管道-过滤器体系结构设计的数字图像处理系统。分别实现对彩色图像的
 - RGBFilter (三色素滤波, Red, Green, Blue; 值0: 255),
 - ThreshHoldFilter (阈值滤波;例如超过186,全过滤掉),
 - TransFilter (变换滤波;用数学公式改变颜色的强度)
 - GrayOutFilter(变灰滤波; 图片颜色变得暗淡)





利用管道-过滤器架构设计的数字图像处理系统用户界面图



Unix support for pipes and filters architecture

例4. 在Unix环境下怎样搭建、实现管道-过滤器架构 Example 4. 如下的Unix命令组成一个管道/过滤器程 序,将in.txt文件的包含"Heroin"的行进行排序, 然后将结果输出到out.txt文件中。 命令行:

cat in.txt | grep "Heroin" | sort > out.txt

Cat:将指定

的文件按照 Grep: 在本文档

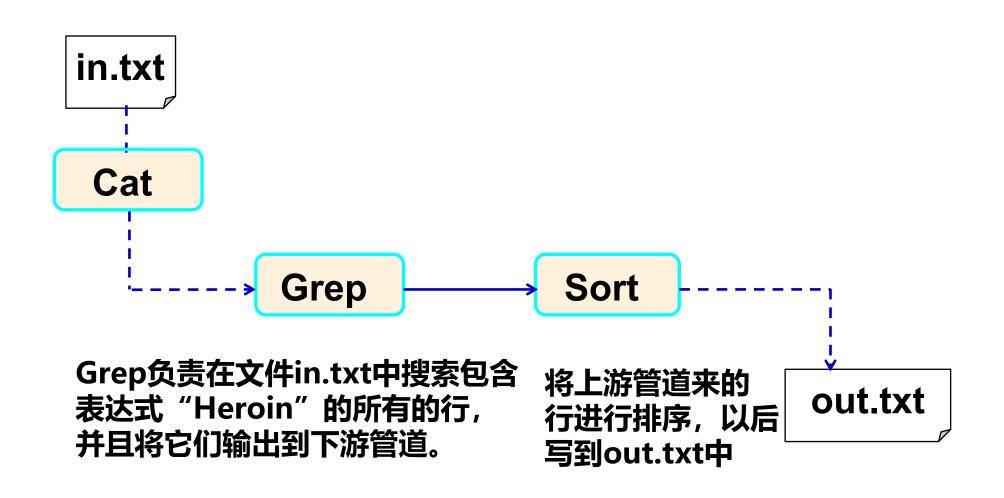
标准输出到 中查找字符串

Sort:

将文档内容排序

显示器

以上命令行的管道/过滤器结构图如下。



- · 带有分支的管道-过滤器架构
- The pipes and filters pattern allows filters with multiple input and/or multiple output pipes to be connected in any directed graph (有向图) structure.
- In the UNIX system, the tee filter provides a mechanism to split a stream into two streams

例5. 在Unix下搭建分支的管道-过滤器架构

Example 3: Tee & Join in Unix

Task: Print a sorted list of words that occur more than

once

mknod pipeA p

mknod pipeB p

sort pipeA > pipeB &

cat file1 | tee pipeA | sort -u | comm -13 - pipeB > file2

Tee: 读取标准输入的

数据,并将其内容输

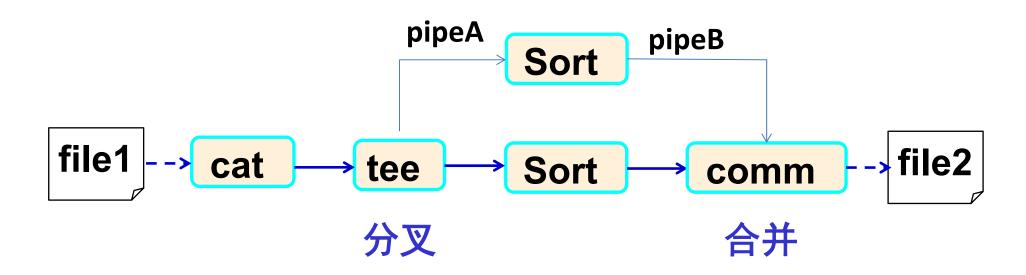
出成几个文件。

comm命令可以用于两个

文件之间的比较。前提:

两个文件必须有序。

以上命令行的管道/过滤器结构图如下。

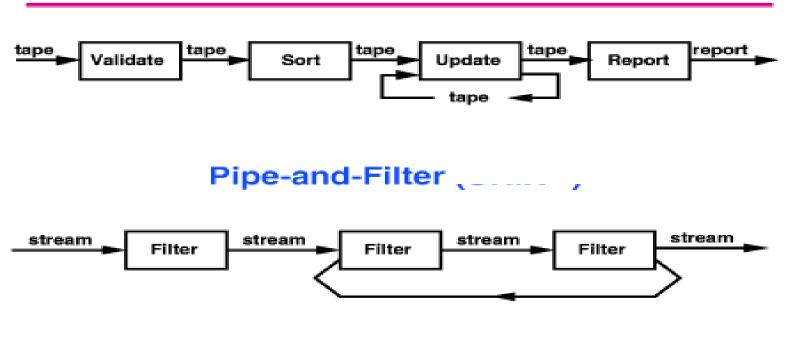


最后将在file1中出现过2次或者更多次的文字打印出来



Comparison of batch sequential architecture and pipes and filters architecture

Batch Sequential





Comparison between batch sequential architecture and pipes-and-filters architecture

- · 相似之处: 处理过程之间互不调用 (independent processing modules)
- In both Batch sequential and pipes and filters architectures, the processing steps are independent of each other, that is
 - >one filter in pipes and filters architecture doesn' t call any other filters in the system
 - ➤one processing step in batch sequential architecture doesn't call any other processing step in the system

区别1: 数据处理方式不同 (the way for data processing is different):

- 1. In batch sequencial Architecture (数据以块状形式传输), each step runs to completion before the next step starts, that is, data is transmitted as a whole between steps
- 2. In pipes and filters architecture (数据以流的 形式传输), the filters Incrementally transform some of the source data into sink data, that is, the filters do stream to stream transformation

区别 2: 数据量不同 (data amount is different):

- 1. In batch sequential Architecture, the amount of data is limited
- 2. In pipes and filters architecture, the amount of data can be unlimited

