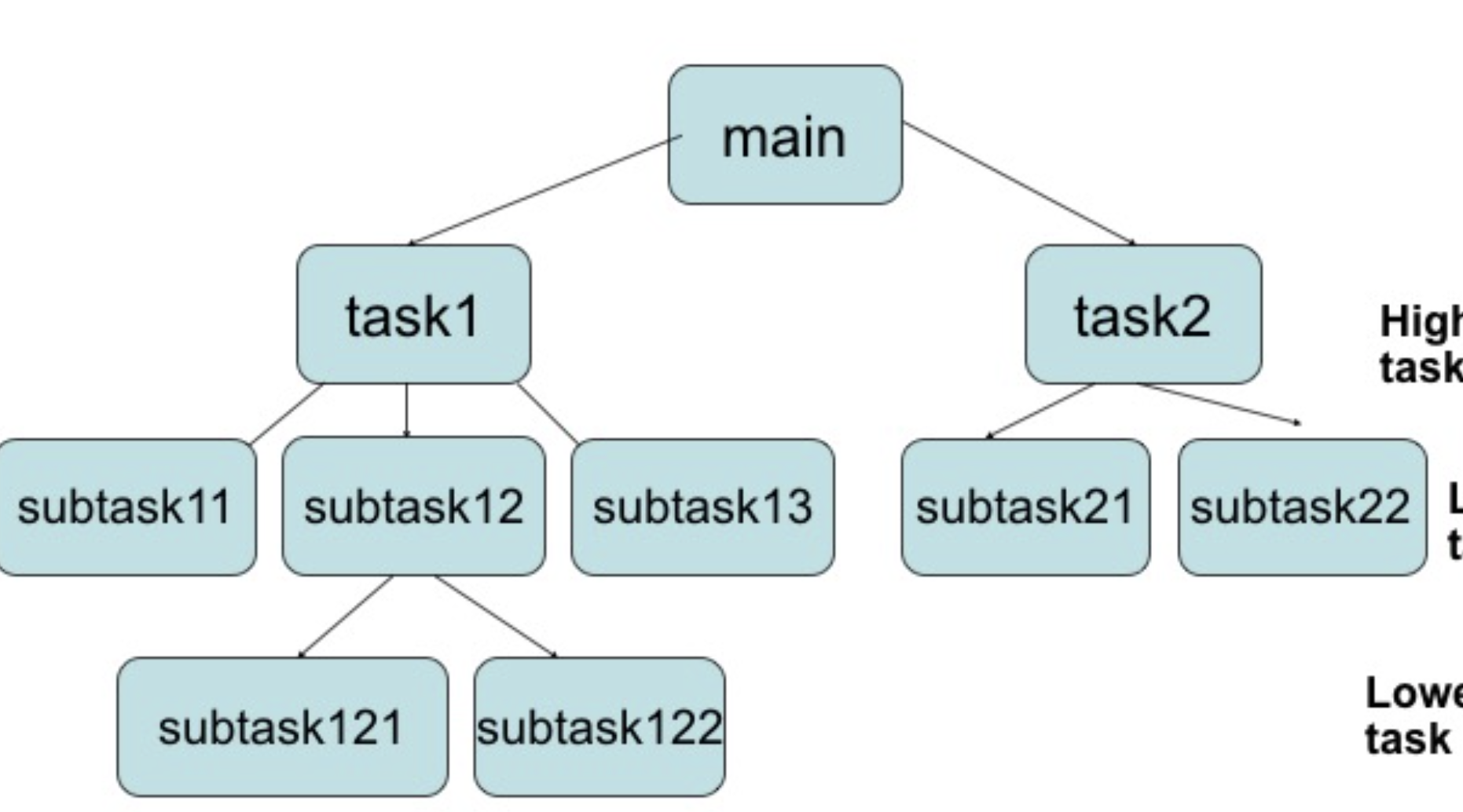
software architecture style

主程序-子程序风格是结构化程序设计的一种典型风格，面向功能设计系统，逐步分解功能，在这种风格里，component是主程序、子程序。连接器是调用-返回机制。

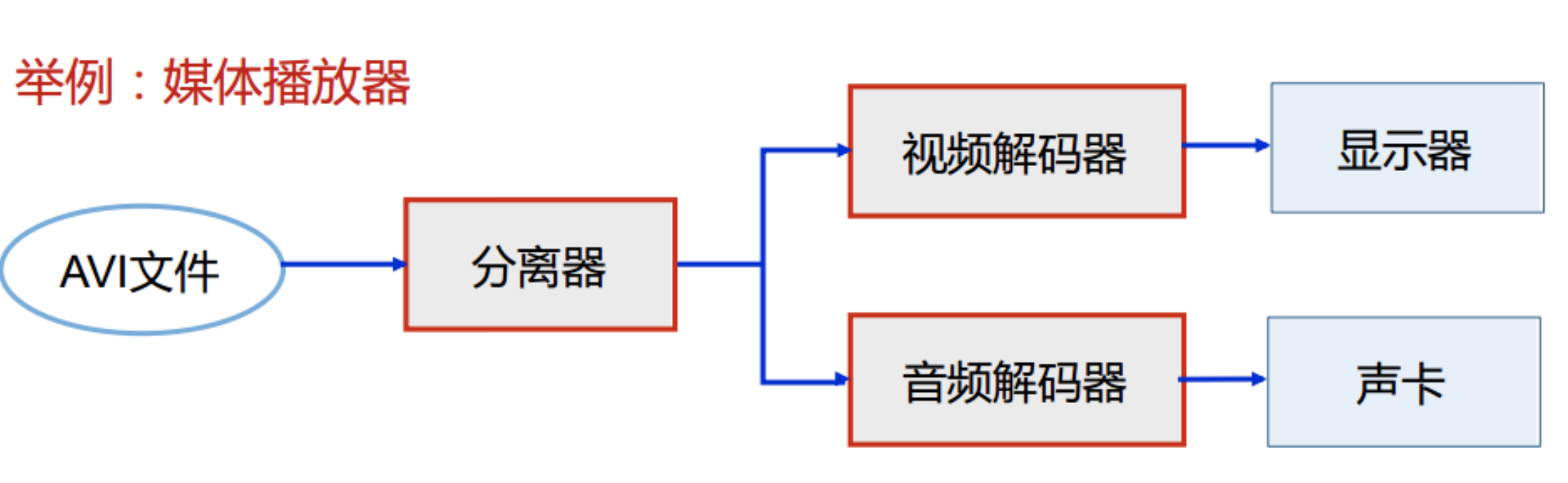
在主程序中调用多个子程序，也就是我们常用的main函数。其中子程序还可以调用它自己的子程序，来实现我们的功能。



管道过滤器风格是面向数据流的软件体系结构，在这个体系结构中，**每个构件都有一组输入和输出，构件读输入的数据流，经过过滤器处理，然后产生输出数据流**

**最主要的特点是，一个步骤的输出是下一个步骤的输入**。

在这种风格里，component是filter，连接器是pipe

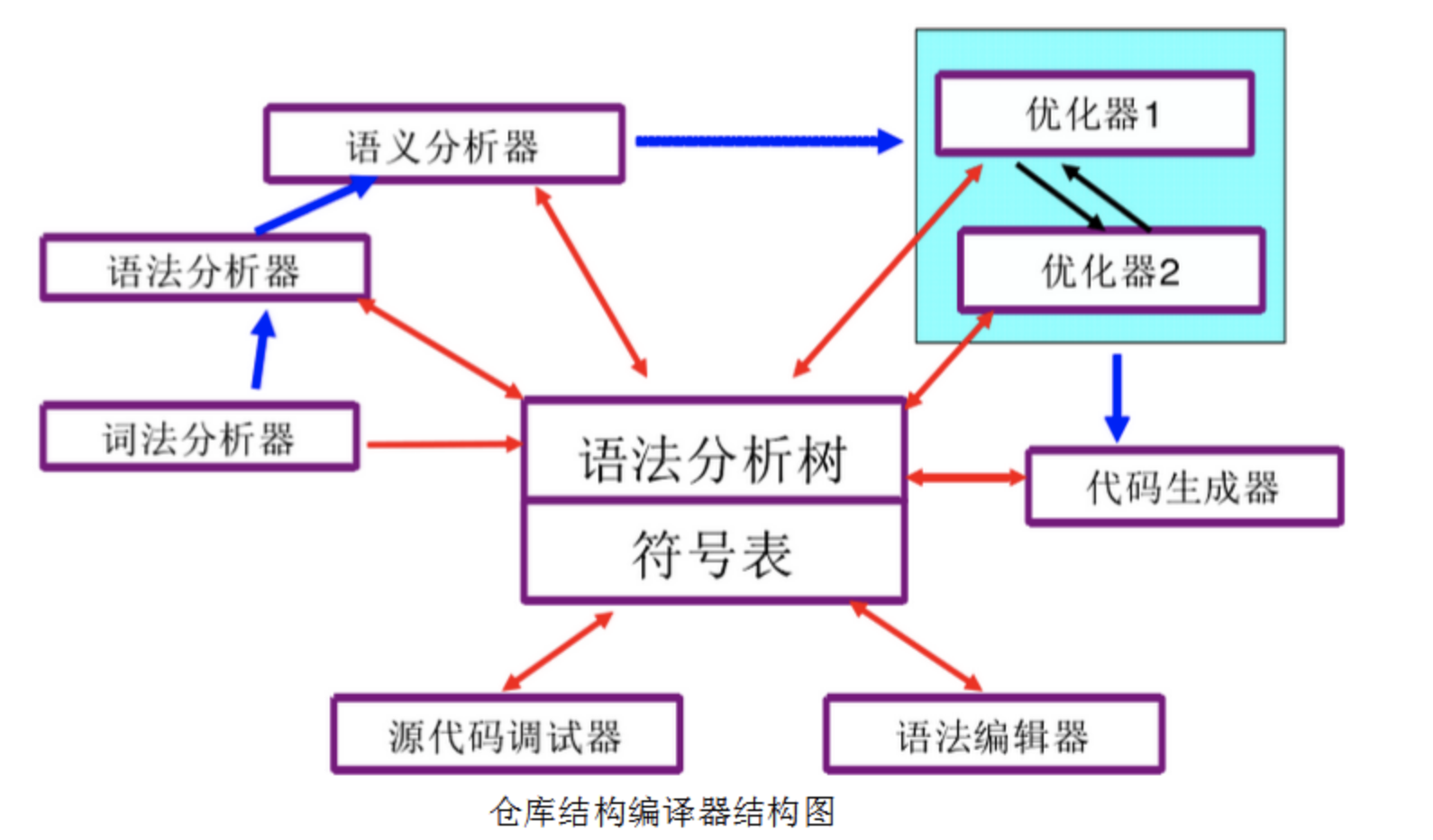


举个例子：这是一个mp4 player，数据流的开始是一个mp4文件，然后经过pipe到达第一个filter分离器，分离出视频流和音频流，然后通过pipe到达视频解码器和音频解码器filter，然后通过pipe到达声卡和显示器

第二个例子，在linux中，也有管道过滤器，我们可以使用管道符号，例如我们在使用grep命令时，we could use the symbol “|” and a filter to process this data,for example : grep ‘david’name.txt | grep ‘Lee’，先筛选出含david的名字，再从结果中筛选出带lee的名字

仓库风格**Repository Architecture** style

是一种以数据为中心的体系结构，适合于数据由一个模块产生而由其他模块使用的情况。

例如编译器这样的软件，就会用到仓库风格

仓库存储了固定不变的语法分析树和符号表，其他所有的component（例如 语法分析器，词法分析器，代码生成器）都需要使用到这些数据都需要到中央仓库来获取