2021年7月15日星期四

1、先把函数的流程读完，然后理解此函数的功能，然后再实现该函数中各个功能模块；

2、写一个类时，同时完整的写上构造函数和析构函数；

3、解析配置文件前的一些打印工作；

2021年7月16日星期五

1、抓住主要功能，抓住我想要的，比如书写一个类时，把无关的成员先屏蔽；

2、c++的一些语法概念，如函数模板，函数调用运算符()，结构体和类的区别，等；

2021年7月19日星期一

1、函数名 ——> 函数参数 ——> 函数体；

2、debug进去，看构造函数，析构函数调用顺序情况；

2021年7月22日星期四

1、先把声明，定义的架子写出来，而不要把内容全部填满，如类或结构体的定义，里面有一些成员变量和成员函数是暂时用不到的，可不必全部定义；

2、函数实现：一层一层的实现，在一层当中，先大概理解每一个函数的功能即可，不必深入，到下一层实现时再深入研究它的功能；

3、变量的定义可以先不关注，重点关注函数功能的实现；

2021年7月23日星期五

1、实现scanArgv函数，完成命令行参数的读取与赋值，赋值给m\_bitstreamFileName和m\_reconFileName；

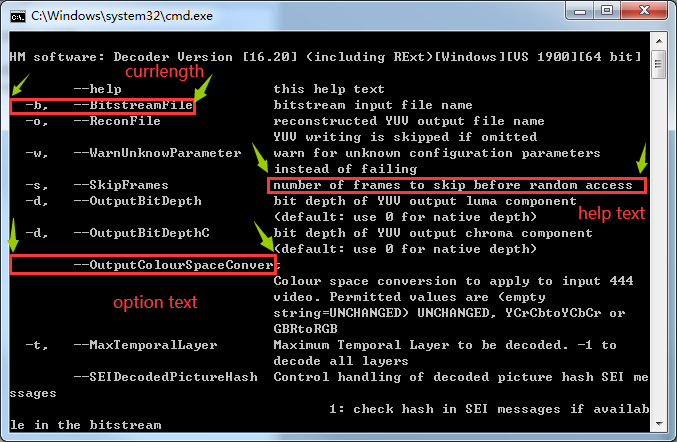
2、对parseGNU函数进行了改写，使其更简洁，意思更明了：decoder端可以处理的命令行参数格式为：-b str.bin或—BitstreamFile=str.bin或--help；

2021年7月26日星期一

1、help内容打印格式：

option text（opt\_width（占33字符）） help text（desc\_width（占47字符））

currlength ——> 短option和长option长度之和



2021年7月27日星期二

1、经验：实现顺序上，实现一个完整的功能，而不仅仅是实现一个函数；

2、完成对help内容打印的实现；

2021年9月18日星期六

1. 注释全部改为英文；
2. 命名习惯更加适合自己；

2021年9月27日星期一

1、从码流文件中提取一个NALU，并去除NALU中的leading\_zero\_8bits（如有），zero\_byte，起始码（start\_code\_prefix\_one\_3bytes），trailing\_zero\_8bits（如有）；

2、阅读spec Annex B Byte stream format；