程序实现

## 读取conf文件

主要函数

int **loadopts**(const char \*file, opt\_t \*opts);

File为conf文件

opts在实际工作中 传入的是一个已经预先定义好的结构体数组

extern opt\_t sysopts[]; /\* system options table \*/

该数组内已经提前定义且对应conf的各项参数设置

数组内同时关联了各个全局变量

/\* system options buffer -----------------------------------------------------\*/

static prcopt\_t prcopt\_;

static solopt\_t solopt\_;

static filopt\_t filopt\_;

static int antpostype\_[2];

static double elmask\_,elmaskar\_,elmaskhold\_;

static double antpos\_[2][3];

static char exsats\_[1024];

static char snrmask\_[NFREQ][1024];

函数内部实现匹配各个参数项 并利用下面的形式

void \*var; (void \*)&全局变量.成员

直接赋值给各个全局变量

然后通过函数

void **getsysopts**(prcopt\_t \*popt, solopt\_t \*sopt, filopt\_t \*fopt);

将系统变量赋值给创建的临时变量，及完成了一次conf文件的读取

即conf文件->读取到全局变量->赋值给需要的结构体

全局变量的设立只是为了读取 如果读取多个文件，为了防止上一次读取的变量还存在于全局变量中，需要使用函数

void **resetsysopts**(void);

对全局变量进行一次重设将全局变量的各个项都重置为default值 。

（正常来说 全局变量建立后并没有赋值，在读第一个文件前 应该也需要对全局变量使用一次该函数，为全局变量赋予初值，以避免读取的conf文件缺失的项太多，导致程序运行出现未知的错误）

全局变量的初值是在源代码中赋予了一份默认值，面对不同的处理场景，默认值可能会需要设置不同的值，可以在主程序外添加一份默认的conf文件，该conf文件的意义是可以重设系统的默认值，这样在实际操纵中先load默认conf到系统，再读取的的conf文件可以缺省大部分的值，只需要输入在默认conf的基础上需要修改的项即可。