**ROS中slam\_gmapping、map\_server源码分析**

这两个包是在建图时是用得最多的。

1、slam\_gmapping

slam\_gmapping包也是依赖开源openslam\_gmapping库。

在slma\_gmapping包中编译后会生成好几个节点，不过真正需要调用的建图节点的主要实现在slam\_gmapping.cpp中，通过roslaunch slam\_gmapping slam\_gmapping.launch或者rosrun gmapping slam\_gmapping scan:=scan启动，开始扫图。

首先是初始化，载入或者配置参数。

然后调用startLiveSlam函数开启建图模式，在这个函数中会发布三个话题，/enropy(机器人姿态分布熵的估计)，/map(建图过程中的数据，用占有网格数据描述，其值在0~255之间)，/map\_metadta(地图的描述消息)；会发布一个服务，/dynamic\_map，也是为了获取地图数据；会订阅激光数据和tf坐标转换后的数据，还用到了message\_filters方法，也是为了加快数据传输效率。此外，还开启了一个线程，用于不断广播map和odom之间的坐标变换。

其次的关键就是对获取的点云scan数据进行处理，这个在回调函数laserCallback中实现。一旦第一次获取到scan数据，会调用initMapper函数形成一个初始的地图形状，这个函数里面也是进行了大量tf变换，然后通过getOdomPose函数获取机器人初始位置，正常情况下该函数只会执行一次，再往后获取到的scan数据则由addScan函数处理了。addScan函数中会先得到里程计数据，然后处理scan数据。最后更新地图数据，由updateMap函数完成，并调用computePoseEntropy函数计算机器人位置分布熵，同时发布map和map\_metadata消息。

slam\_gmapping包的主要工作就这些，就是将数据处理为了在ros通讯机制下的，其核心算法还是在openslam\_gmapping里面实现的。

2. map\_server

map\_server的功能就是地图[服务器](https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)。编译map\_server后会生成map\_server\_image\_loader动态库，这个用于处理pgm格式的图片，其依赖了SDL库；还会生成map\_server和map\_saver节点。

map\_server节点的源代码很容易看懂，主要就是为了加载pgm格式的地图，用yaml文件描述的，同时发布map\_metadata和map话题，以及static\_map服务，其目的都是为了方便其他节点获取到地图数据。指令为rosrun map\_server map\_server mymap.yaml。

map\_saver的作用就是为了将slam\_gmapping建图数据保存下来，主要就是订阅map消息。rosrun map\_server map\_saver -f mymap。