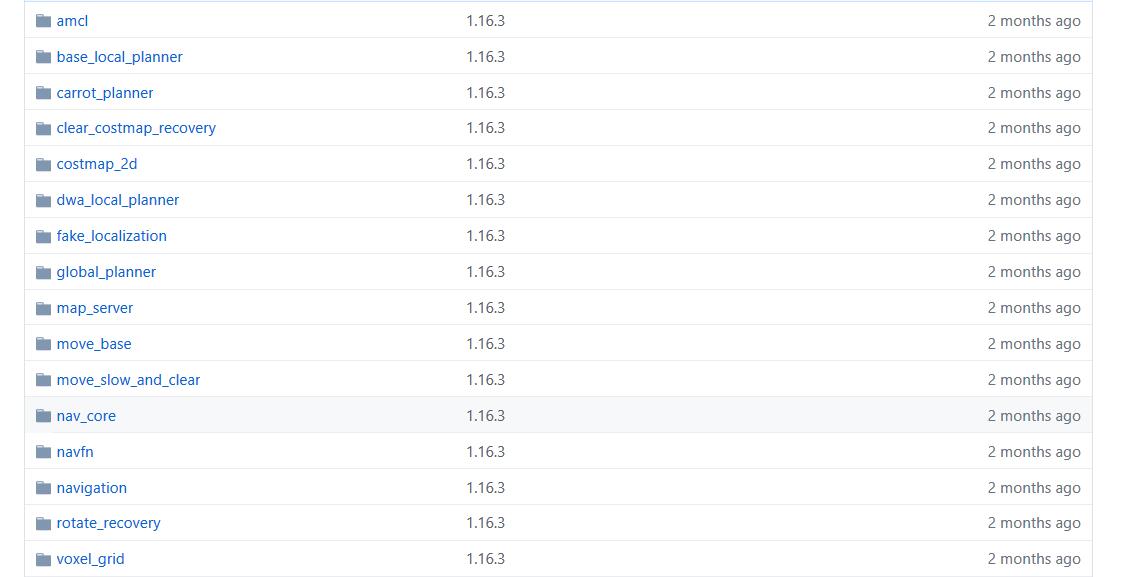


这个是wiki上的movebase框架图



我们对照github上的线程标注一下每个线程的作用

1.amcl是一种机器人在2D中移动的概率定位系统。 它实现了自适应（或KLD采样）蒙特卡罗定位方法，该方法使用粒子滤波器来针对已知地图跟踪机器人的位姿。

2.base\_local\_planner使用Trajectory Rollout and Dynamic Window approaches来做平面上运动的机器人局部导航，控制器基于给定的路径规划和costmap生成速度命令后发送给机器人，**我们应用movebase算法大部分需要修改的参数都在这里**。

3.carrot\_planner试图找到适当的位置来使机器人跟随，完成了两个任务：

（1）规划器获取用户指定的目标位置，检查用户指定的位置是否在障碍区中。

（2）如用户指定的位置在障碍区中，规划器就会在机器人与指定位置的已规划路径中寻找一个在障碍区外的可行的目标位置，然后再将此目标位置发送给局部规划器或者控制器

4.clear\_costmap\_recovery为导航堆栈提供了一种恢复行为，其尝试通过将导航堆栈使用的代价地图恢复到给定区域外的静态地图来清除空间。（这部分还不太理解）

5.costmap\_2d提供了一种2D代价地图（关于代价地图部分的知识还需补充）的实现方案，该方案从实际环境中获取传感器数据，构建数据的2D或3D占用栅格（取决于是否使用基于体素的实现），以及基于占用栅格和用户定义膨胀半径的2D代价地图的膨胀代价。

（1）costmap\_2d-staticmap静态地图主要包含来自外部源的不变数据。

（2）costmap\_2d-inflation为代价地图的代价值定义**5个与机器人有关的标记**：致命的（"Lethal" cost）、内切（"Inscribed" cost）、可能外切（"Possibly circumscribed" cost ）、自由空间（"Freespace" ）、未知（"Unknown"） 。

6.dwa\_local\_planner使用DWA（Dynamic Window Approach）算法实现了平面上移动机器人局部导航功能。（和base\_local\_planner什么关系？）

7.fake\_localization用来替代定位系统，并且提供了amcl定位算法ROS API的子集。

8.global\_planner为导航实现了一种快速，内插值的全局路径规划器， 继承了nav\_core包中nav\_core::BaseGlobalPlanner接口，该实现相比navfn使用更加灵活。

9.map\_server提供了一个map\_server ROS 节点, 该节点通过ROS 服务器方式提供地图数据。提供了map\_saver命令行utility, 使用该工具可将动态创建的地图保存成文件。

10.move\_base将全局路径和局部路径规划程序链接在一起，以完成其全局导航任务；维护两个costmaps，一个用于全局路径规划器，一个用于局部路径规划器，ove\_base\_msgs 能够保留move\_base包的操作描述和相关消息。

11.move\_slow\_and\_clear是一种简单的修复机制，用来清除代价地图中信息并且限制机器人移动速度。（是一种简单的修复机制，用来清除代价地图中信息并且限制机器人移动速度。）

12.nav\_core包含了导航功能包的关键接口。主要是：全局路径规划器、局部路径规划器、修复机制接口

13.navfn快速内插值的导航功能，用于为移动基座创建路径规划，有两种：

（1）机器人为圆形并利用代价地图来进行操作，以从栅格的起点到终点找到代价最小的路径规划；

（2）Dijkstra算法；

（3）在move\_base中被作为全局路径规划器插件使用。

14.navigation——定义cmakelist

15.rotate\_recovery给导航功能包提供了rotate\_recovery::RotateRecovery修复机制，它尝试让机器人执行360度旋转来完成清理导航功能包里的代价地图的空间。

16.voxel\_grid实现里高效的3D voxel grid