

ROS(机器人操作系统)调研报告

李易 PB2311726

1.技术背景

1.1 什么是ROS

ROS(Robot Operating System)是一种开源的机器人操作系统，它提供了一系列的工具和库，用于开发机器人应用。ROS的主要目标是为机器人开发者提供一个统一的框架，使得机器人应用的开发更加简单、高效。[ROS官方文档](#) [ROS2官方文档](#)

1.2 ROS的优势

对于机器人开发者来说，ROS 可以让你专注于算法和应用的开发，而无需过多关注底层硬件和通信细节：

- 提供硬件抽象层，支持多种传感器和执行器。
- 提供丰富的工具和库，如导航、SLAM（同步定位与地图构建）、运动规划等。
- 支持分布式计算，允许在多台机器上运行不同的模块。
- 提供强大的可视化工具，如 RViz 和 Gazebo，用于调试和仿真。

2.核心功能

ROS 的设计与操作系统的核心功能紧密相关，主要体现在以下几个方面：

1. 模块化与进程管理 ROS 将系统功能划分为多个独立节点（Node），每个节点对应操作系统中的一个进程。节点之间通过发布-订阅（Topic）和请求-响应（Service）机制通信，类似于操作系统的进程间通信（IPC）。
2. 资源管理与调度 ROS Master 负责协调节点间的通信，类似于操作系统的调度器管理进程资源。支持多线程任务，可通过优先级调度优化实时性。
3. 文件与数据管理 使用参数服务器（Parameter Server）存储全局配置，类似于操作系统的共享内存或配置文件管理。提供日志记录和调试工具（如 rosbag），支持数据持久化存储。
4. 硬件抽象与设备驱动 提供硬件抽象层（HAL），支持传感器和执行器的统一接口，类似于操作系统对硬件的驱动管理。

3.核心框架与操作系统

- 节点 每个 ROS 节点对应操作系统中的一个独立进程，通过进程调度分配 CPU 资源。

- 发布-订阅 基于 **TCP/IP** 协议实现异步通信，类似操作系统的管道（**Pipe**）或消息队列（**Message Queue**）。
- 请求-响应 基于 **TCP/IP** 协议实现同步通信。
- 内存管理
 - 参数服务器 存储全局配置参数，类似操作系统的共享内存或环境变量。可优化方向：结合操作系统的虚拟内存机制，避免参数频繁读写导致的性能瓶颈。
 - 消息序列化 **ROS** 消息（**Message**）通过序列化传输。

4.可选择方向

节点调度实验

编写两个 **ROS** 节点（如数据采集与处理），通过调整优先级观察执行顺序变化，验证操作系统的调度策略。

与内存相结合

尝试对内存泄漏进行检测，利用上操作系统的内存管理机制

与文件相结合

可以对**ROS**文件读写的性能进行优化

5.总结

往年与**ROS**相关的选题：[ForWWWard](#)

相关资料

[ROS简介-从零开始讲解ROS（适合超零基础阅读）-CSDN博客](#)

[机器人操作系统（ROS）浅析.pdf](#)

[ApolloAuto/apollo: An open autonomous driving platform](#)

[古月居 - ROS机器人知识分享社区](#)