

# 自动驾驶C++后端技术栈

## 自动驾驶C++后端技术栈

### 适宜建议

- 对自动驾驶和机器人相关领域感兴趣，想从事该领域工作
- 本科或研究生基于ros框架，研究过算法方向，但是发布顶刊论文比较吃力，学历不占优势
- 目前只学过C++通用的技术栈，想钻研一个细分领域，提高自己校招的竞争优势
- 以上总结三种建议，后续学习需要花大量时间

### C++通用技术栈

#### Linux学习

##### Ubuntu系统（x86/ARM）

熟悉Ubuntu操作系统的基本知识和常用命令。

在x86和ARM不同体系结构上进行开发或运行，了解不同体系结构的特点和差异。

##### Linux操作系统知识

Linux的基本概念和原理，包括进程管理、内存管理、文件系统、设备驱动等。

深入了解Linux的进程间通信机制，如管道、套接字和信号等。

##### Linux系统函数知识

熟悉Linux系统提供的常用函数，比如文件操作、进程管理、网络编程等。(大家常做的webserver项目)

##### Linux相关性能分析和优化知识

了解Linux性能分析工具和优化技术，识别和解决系统性能瓶颈

工具如top、iotop、strace、perf等进行性能分析和排查问题

#### C++

##### C++基础语法

包括变量、数据类型、运算符、条件语句、循环语句等

##### C++特性

面向对象编程（OOP）、类和对象、继承、多态性

智能指针，线程库，C++常用特性

##### 模板编程

了解模板的基本语法、函数模板、类模板、模板特化等

##### 常见标准库

boost库，STL库

##### 计算机通识技术栈

数据结构，设计模式，计算机网络，计算机操作系统

# 自驾领域技术栈

## 项目构建工具

cmake、bazel

<https://github.com/Kitware/CMake>

<https://github.com/search?q=bazel&type=repositories>

## ros1

ros教程: <https://wiki.ros.org/>

### ros基本原理

- Nodes: ROS系统由多个节点组成, 每个节点代表一个进程, 可以是传感器、控制算法等。节点之间通过Messages进行通信。
- Topics: 主题是节点之间进行异步通信的方式, 一个节点可以Publish消息到一个主题, 其他节点可以通过Subscribe该主题来接收相关消息。
- Services: 服务是节点之间进行同步通信的方式, 一个节点可以提供一个服务, 其他节点可以请求该服务来获取特定的信息或执行特定的操作。
- Parameter Server: 参数服务器是一个分布式的键值存储系统, 用于存储和共享配置参数和运行时参数。节点可以从参数服务器中读取和写入参数。
- Messages: ROS使用基于消息的通信模型, 消息是以特定格式定义的数据结构, 用于在节点之间传递信息。消息可以是标准类型(如整数、字符串等), 也可以是自定义的复杂数据类型。

发布订阅模型, 消息序列化协议 (json,protobuf,xml)

### ros相关源码

roscpp <https://wiki.ros.org/roscpp>

roscpp\_serialization

## apollo

github: <https://github.com/ApolloAuto/apollo>

cyber: <https://github.com/ApolloAuto/apollo/tree/master/cyber>

通信模块: intra, shm, fastdds

cyber doc: [https://github.com/ApolloAuto/apollo/blob/master/docs/04\\_CyberRT/](https://github.com/ApolloAuto/apollo/blob/master/docs/04_CyberRT/)

序列化协议: protobuf

## DDS

ros2 ,apollo

常见: fastdds : <https://github.com/eProsima/Fast-DDS>