GPS导航gRPC接口（gps_navigation.proto）

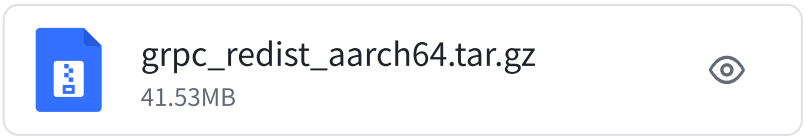
一、gRPC安装指南

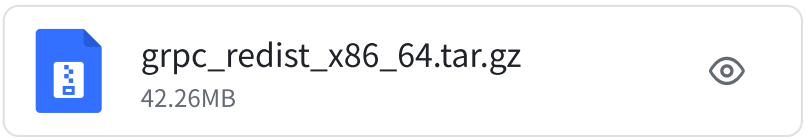
1.1 系统要求

- Ubuntu 20.04 LTS (推荐)
- ROS Noetic
- CMake 3.5+
- Git

1.2 依赖安装

方法一（经验证可行）：





代码块

```
#解压对应架构下的安装包并进入包内
sudo mv ./bin /usr/local/bin
sudo mv ./lib /usr/local/lib
sudo mv ./include /usr/local/include
```

方法二：

代码块

```
# 安装基础编译工具
sudo apt update
sudo apt install -y build-essential autoconf libtool pkg-config

# 安装gRPC依赖
```

```
sudo apt install -y libssl-dev zlib1g-dev

# 安装gRPC和Protobuf
sudo apt install -y libgrpc-dev libgrpc++-dev protobuf-compiler protobuf-compiler-grpc libprotobuf-dev
```

1.3 验证安装

代码块

```
# 验证Protobuf
protoc --version  # 应显示libprotoc 3.12.4或更高版本

# 验证gRPC
pkg-config --modversion grpc++  # 应显示1.38.0或更高版本
```

1.4 可选：从源码安装（最新版本）

代码块

```
# 安装依赖
sudo apt install -y cmake git

# 克隆gRPC仓库
git clone --recurse-submodules -b v1.46.3 https://github.com/grpc/grpc
cd grpc
mkdir -p cmake/build
cd cmake/build

# 编译安装
cmake -DgRPC_INSTALL=ON \
      -DgRPC_BUILD_TESTS=OFF \
      -DCMAKE_INSTALL_PREFIX=/usr/local \
      ../..
make -j$(nproc)
sudo make install

# 更新动态链接库缓存
sudo ldconfig
```

二、Proto文件生成头文件

2.1 文件结构准备

代码块

```
gps_navigation_grpc/
├── CMakeLists.txt
├── package.xml
├── proto/
│   └── gps_navigation.proto
└── src/
    ├── gps_server.cpp
    └── gps_client.cpp
```

2.2 Proto文件内容（`proto/gps_navigation.proto`）

代码块

```
syntax = "proto3";

package gps_navigation;

import "google/protobuf/empty.proto";

message Descartes {
    float x = 1;    // 经度
    float y = 2;    // 纬度
    float z = 3;
}

message Euler {
    float roll = 1;
    float pitch = 2;
    float yaw = 3;    // 航向角（弧度）
}

message Pose {
    Descartes position = 1;
    Euler attitude = 2;
```

```
}

message NaviState {
    Descartes position = 1;
    Descartes velocity = 2;
    Euler attitude = 3;
    bool navigating = 4;    // 是否在导航中
    int32 code = 5;         // GPS 状态码
}

message Response {
    bool succeeded = 1;
    string msg = 2;
}

message NaviResponse {
    bool succeeded = 1;
    string msg = 2;
    bool arrived = 3;       // 是否到达
    NaviState state = 4;
}

service GPSNaviController {
    rpc setDestination(Pose) returns (Response);
    rpc startNavi(google.protobuf.Empty) returns (stream NaviResponse);
    rpc stopNavi(google.protobuf.Empty) returns (Response);
    rpc getState(google.protobuf.Empty) returns (NaviState);
}
```

2.3 执行生成命令

代码块

```
# 设置脚本可执行权限
protoc --cpp_out=. --grpc_out=. --plugin=protoc-gen-grpc=/usr/local/bin/grpc_cpp_plugin gps_navigation.proto
```

2.5 验证生成文件

生成的头文件和源文件应位于 `include/gps_navigation_grpc/` 目录：

```
include/gps_navigation_grpc/
├── gps_navigation.grpc.pb.cpp
├── gps_navigation.grpc.pb.h
├── gps_navigation.pb.cpp
└── gps_navigation.pb.h
```

三、编译与测试

src_gps_grpc_06191056.zip
259.96KB

3.1 CMakeLists.txt配置

代码块

```
cmake_minimum_required(VERSION 3.5)
project(gps_navigation_grpc)

find_package(catkin REQUIRED COMPONENTS
  roscpp
  std_msgs
)

# 查找gRPC和Protobuf
find_package(Protobuf REQUIRED)
#find_package(gRPC REQUIRED)

catkin_package()

# 生成的消息头文件目录
include_directories(
  include
  ${catkin_INCLUDE_DIRS}
  ${Protobuf_INCLUDE_DIRS}
  ${gRPC_INCLUDE_DIRS}
)

# 添加生成的协议文件
file(GLOB PROTO_SRCS "include/gps_navigation_grpc/*.pb.cc")
file(GLOB PROTO_HDRS "include/gps_navigation_grpc/*.pb.h")

# 编译服务端
add_executable(gps_server src/gps_server.cpp ${PROTO_SRCS})
```

```
target_link_libraries(gps_server
  ${catkin_LIBRARIES}
  ${Protobuf_LIBRARIES}
  grpc++ grpc++_reflection grpc gpr protobuf absl_log_internal_check_op absl_log_internal_message absl_log_internal_nullguard absl_cord absl_cordz_info absl_synchronization absl_cordz_functions
)

# 编译客户端
add_executable(gps_client src/gps_client.cpp ${PROTO_SRCS})
target_link_libraries(gps_client
  ${catkin_LIBRARIES}
  ${Protobuf_LIBRARIES}
  grpc++ grpc++_reflection grpc gpr protobuf absl_log_internal_check_op absl_log_internal_message absl_log_internal_nullguard absl_cord absl_cordz_info absl_synchronization absl_cordz_functions
)

# 安装规则（可选）
install(TARGETS gps_server gps_client
  RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)

```

3.2 package.xml配置

代码块

```
<?xml version="1.0"?>
<package format="2">
  <name>gps_navigation_grpc</name>
  <version>0.1.0</version>
  <description>gRPC integration for GPS navigation in ROS</description>

  <maintainer email="you@example.com">Your Name</maintainer>
  <license>Apache 2.0</license>

  <buildtool_depend>catkin</buildtool_depend>

  <depend>roscpp</depend>
  <depend>std_msgs</depend>
  <depend>geometry_msgs</depend>

  <build_depend>libgrpc-dev</build_depend>
```

```
  <build_depend>libgrpc++-dev</build_depend>
  <build_depend>protobuf-compiler</build_depend>
  <build_depend>protobuf-compiler-grpc</build_depend>

  <exec_depend>libgrpc++</exec_depend>
  <exec_depend>libprotobuf</exec_depend>
</package>

```

3.3 编译工作空间

代码块

```
# 在工作空间根目录
cd ~/gps_navigation_ws

# 清理之前的构建
rm -rf build devel

# 构建工作空间
catkin_make

# 配置环境
source devel/setup.bash
```

3.4 测试流程

步骤1: 启动服务端

代码块

```
rosrun gps_navigation_grpc gps_navi_server
```

预期输出：

代码块

```
[ INFO] [时间戳]: GPS Navigation Server listening on 0.0.0.0:50049
```

步骤2: 启动客户端

```
代码块
1   rosrun gps_navigation_grpc gps_navi_client
```

预期输出：

```
代码块
1   [ INFO] [时间戳]: Waiting for server to start...
2   [ INFO] [时间戳]: Current State:
3   [ INFO] [时间戳]:   Position: [0.00, 0.00, 0.00]
4   [ INFO] [时间戳]:   Velocity: [0.00, 0.00, 0.00]
5   [ INFO] [时间戳]:   Attitude: R:0.0° P:0.0° Y:0.0°
6   [ INFO] [时间戳]: Destination set: Destination set successfully
7   [ INFO] [时间戳]: Starting navigation...
8   [ INFO] [时间戳]: Nav Update: Pos[0.10, 0.06] | Speed: 0.11 m/s | Yaw: 31.0° |
    Arrived: NO | Status: Navigating to destination
9   ...
10  [ INFO] [时间戳]: Destination reached!
11  [ INFO] [时间戳]: Current State:
12  [ INFO] [时间戳]:   Position: [5.00, 3.00, 0.00]
13  [ INFO] [时间戳]:   Velocity: [0.00, 0.00, 0.00]
14  [ INFO] [时间戳]:   Attitude: R:0.0° P:0.0° Y:31.0°
```

3.5 功能测试用例

测试用例	方法	预期结果
设置目的地	`setDestination(5.0, 3.0, 0.0)`	返回成功消息
开始导航	`startNavi()`	接收流式位置更新
中途停止	`stopNavi()`	导航立即停止
获取状态	`getState()`	返回当前位置和姿态
目的地到达	等待导航完成	收到到达通知

无效目的地	`setDestination(0,0,0)`	返回错误消息
无目的启动	`startNavi()` 未设目的地	返回错误消息

3.6 性能测试

代码块

```
# 启动服务端
rosrun gps_navigation_grpc gps_navi_server

# 测试RPC延迟
grpc_cli call localhost:50049 getState ""

# 测试流式响应性能
grpc_cli call localhost:50049 startNavi ""
```

故障排除

常见问题解决方案

1. **Protobuf版本冲突**

代码块

```
sudo apt remove libprotobuf-dev protobuf-compiler
sudo apt install libprotobuf-dev=3.12.4-1ubuntu7 protobuf-compiler=3.12.4-1ubuntu7
```

2. **gRPC链接错误**

在CMakeLists.txt中添加：

代码块

```
target_link_libraries(gps_navi_server
  ...
```

```
    grpc++
    grpc
    protobuf
    gpr
    address_sorting
    upb
    absl_bad_any_cast_impl
    absl_bad_optional_access
)
```

3. **缺少abseil库**

```
sudo apt install libabsl-dev
```

4. **生成文件未更新**

```
rm include/gps_navigation_grpc/*
./scripts/generate_grpc_code.sh
```

5. **ROS包未找到**

```
cd ~/gps_navigation_ws
catkin_make
source devel/setup.bash
```