

# RMDB使用指南

---

## RMDB使用指南

环境配置

项目下载

编译

运行 (S/C)

单元测试

flex & bison文件的修改

## 环境配置

---

RMDB需要以下依赖环境库配置：

- gcc 7.1及以上版本（要求完全支持C++17）
- cmake 3.16及以上版本
- flex
- bison
- readline

可以通过命令完成环境配置(以Debian/Ubuntu-apt为例)

```
sudo apt-get install build-essential # build-essential packages, including gcc, g++,  
make and so on  
sudo apt-get install cmake          # cmake package  
sudo apt-get install flex bison      # flex & bison packages  
sudo apt-get install libreadline-dev # readline package
```

可以通过 `cmake --version` 命令来查看cmake版本，如果低于3.16，需要在官网下载3.16以上的版本并解压，手动进行安装。

注意,在CentOS下,编译时可能存在头文件冲突的问题,我们不建议你使用Ubuntu以外的操作系统。

## 项目下载

---

你可以通过以下方式来完成RMDB的下载

```
git clone https://gitlab.eduxiji.net/sxy/rmdb.git
```

## 编译

---

整个系统分为服务端和客户端，你可以使用以下命令来进行服务端的编译：

```
mkdir build
cd build
cmake .. [-DCMAKE_BUILD_TYPE=Debug] | [-DCMAKE_BUILD_TYPE=Release] # []中为可选项
make rmdb <-j4>|<-j8> # 在未完成代码之前，无法编译成功
```

可以使用以下命令来进行客户端的编译：

```
cd rmdb_client
mkdir build
cd build
cmake .. [-DCMAKE_BUILD_TYPE=Debug] | [-DCMAKE_BUILD_TYPE=Release]
make rmdb_client <-j4>|<-j8> # 选择4 or 8线程编译
```

## 运行 (S/C)

首先运行服务端：

```
cd build
./bin/rmdb <database_name> # 如果存在该数据库,直接加载;若不存在该数据库,自动创建
```

然后开启客户端，用户可以同时开启多个客户端：

```
cd rmdb_client/build
./rmdb_client
```

用户可以通过在客户端界面使用exit命令来进行客户端的关闭：

```
RMDB> exit;
```

服务端的关闭需要在服务端运行界面使用ctrl+c来进行关闭，关闭服务端时，系统会把数据页刷新到磁盘中。

- 如果需要删除数据库，则需要在build文件夹下删除与数据库同名的目录
- 如果需要删除某个数据库中的表文件，则需要在build文件夹下找到数据库同名目录，进入该目录，然后删除表文件

## 单元测试

单元测试使用GoogleTest框架，在项目的src/文件夹下包含测试示例文件unit\_test.cpp文件，参赛队伍可以运行unit\_test单元测试来了解单元测试流程。

以unit\_test为例，可以通过以下命令进行测试：

```
cd build
make unit_test
./bin/unit_test
```

## flex & bison文件的修改

---

在parser子文件夹下涉及flex和bison文件的修改，开发者修改lex.l和yacc.y文件之后，需要通过以下命令重新生成对应文件：

```
flex --header-file=lex.yy.hpp -o lex.yy.cpp lex.l
bison --defines=yacc.tab.hpp -o yacc.tab.cpp yacc.y
```