

## 第二讲：源码构建 MySQL 调试环境

知春路遇上八里桥

<2024-05-08 Wed>



① 前置准备

② 编译配置

③ 启动调试



1

## 前置准备



# 系统库和前置依赖

MySQL 8.0.x 代码官网包含安装前置依赖<sup>1</sup>的说明

```
sudo apt install -y build-essential git libaio1 \  
    cmake \  
    libncurses-dev \  
    bison
```

- cmake 编译系统
- ncurses TUI 文本用户界面库
- bison 语法解析器

---

<sup>1</sup><https://dev.mysql.com/doc/refman/8.0/en/source-installation-prerequisites.html>



# 构建工具版本

MySQL 8.0.x 编译需要包含如下依赖及对应版本

```
# GCC 10
gcc --version
# GNU make 3.75 or later.
make --version
# cmake 3.x
cmake --version
# bison 2.1 or later
bison --version
```



## 三方库: boost

- ① boost<sup>2</sup> 是 C++ 常用的第三方库, 它实现了标准库以外的一些常见功能
- ② MySQL 8.0.37 对应 boost 1.77.0 版本<sup>3</sup>
  - ▶ 查看 cmake/boost.cmake 配置文件
  - ▶ [https://boostorg.jfrog.io/artifactory/main/release/1.77.0/source/boost\\_1\\_77\\_0.tar.bz2](https://boostorg.jfrog.io/artifactory/main/release/1.77.0/source/boost_1_77_0.tar.bz2)
- ③ 下载完成后解压到 /opt 文件夹中

# 解压

```
tar xvf boost_1_77_0.tar.bz2
```

# 移动到 /opt

```
sudo mv boost_1_77_0 /opt
```

---

<sup>2</sup><https://www.boost.org/>

<sup>3</sup><https://sourceforge.net/projects/boost/files/boost/1.77.0/>



2

## 编译配置



# 下载源代码

## ❶ 代码目录：存放 MySQL 的源代码的 mysql-server 文件夹

# 源代码存放目录

```
cd /opt/src
```

# 下载源代码

```
git clone https://github.com/mysql/mysql-server.git
```

# 基于 *mysql-8.0.37* 构建, *rtc: Read the Code*

```
git checkout -b rtc mysql-8.0.37
```

## ❷ 安装目录：这里将安装目录放在 /opt 下

- ▶ 代码编译最后安装在 /opt/mysql
- ▶ 初始化数据放在 /opt/mysql/data

## ❸ 代码也可从官网直接下载代码包<sup>4</sup>

- ▶ MySQL Community Server / Version 8.0.37 / Source Code

<sup>4</sup>源码下载 <https://dev.mysql.com/downloads/>





# 准备配置文件

准备一个 `my.cnf` 配置文件放在 `/opt/mysql` 路径下

```
[mysqld]
basedir      = /opt/mysql
datadir      = /opt/mysql/data
log-error    = /opt/mysql/mysqld-error.log
pid-file     = /opt/mysql/mysqld.pid
socket       = /opt/mysql/mysql.sock

[client]
user         = root
socket       = /opt/mysql/mysql.sock
```



# 配置 cmake 及编译安装

- ① 进入源码目录后配置好 cmake, 注意配置选项 <sup>5</sup>

```
cd mysql-server
mkdir build && cd build
cmake -DCMAKE_INSTALL_PREFIX=/opt/mysql \
      -DWITH_BOOST=/opt/boost_1_77_0 \
      -DWITH_DEBUG=1 \
      -DWITH_UNIT_TESTS=1 \
      ..
```

- ② 在 build 目录下通过 make 系统编译

```
make -j$(nproc)
```

- ③ 安装编译产物

```
make install
```

<sup>5</sup><https://dev.mysql.com/doc/refman/8.0/en/source-configuration-options.html>



# 初始化数据库

## ❶ 配置环境变量<sup>6</sup>, 添加 /opt/mysql/bin 目录

```
export MYSQL_HOME=/opt/mysql
export PATH=$MYSQL_HOME/bin:$PATH
```

## ❷ 初始化数据目录: /opt/mysql/data

```
/opt/mysql/bin/mysqld \
--defaults-file=/opt/mysql/my.cnf --initialize-insecure
```

- ▶ --initialize-insecure 允许不安全的初始化
- ▶ --defaults-file 制定默认配置文件

- ❶ Unix 系统默认配置文件为 /etc/my.cnf
- ❷ 设置 MYSQL\_HOME 后默认配置文件为 \$MYSQL\_HOME/my.cnf

<sup>6</sup><https://dev.mysql.com/doc/refman/8.0/en/environment-variables.html>



# 导入示例数据库

## ❶ 员工示例数据库 <sup>7</sup>

```
git clone https://github.com/datacharmer/test_db.git
```

## ❷ 导入测试样例数据

```
mysql -t < employees.sql
```

## ❸ 表：员工、部门以及它们之间的关联关系

- ▶ 员工 employees salaries titles
- ▶ 部门 departments dept\_emp dept\_manager
- ▶ 部门员工关联表 dept\_emp

## ❹ 视图：方便查询数据关系

- ▶ 当前部门员工 current\_dept\_emp
- ▶ 最新日期部门员工 dept\_emp\_latest\_date

<sup>7</sup><https://dev.mysql.com/doc/employee/en/employees-installation.html>



3

## 启动调试



# 启动数据库引擎

直接启动 mysqld

```
/opt/mysql/bin/mysqld \  
--defaults-file=/opt/mysql/my.cnf \  
--gdb
```

- mysqld 可以直接制定配置文件后启动
- --gdb 为 gdb 的 Ctrl-C 安装 SIGINT 信号<sup>8</sup>

启动后查看 pid 文件，通过配置文件 pid-file 设置

```
cat /opt/mysql/mysqld.pid
```

---

<sup>8</sup><https://dev.mysql.com/doc/refman/8.0/en/using-gdb-on-mysqld.html>



# gdb 调试

方法一：直接命令行通过 pid 连接

```
gdb -p 12345
```

```
gdb -p $(cat /opt/mysql/mysqld.pid)
```

方法二：进入 gdb 后，通过 pid 连接

```
$ gdb
```

```
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
```

```
...
```

```
Type "apropos word" to search for commands related to "word".
```

```
(gdb) attach 12345
```



# 结束

