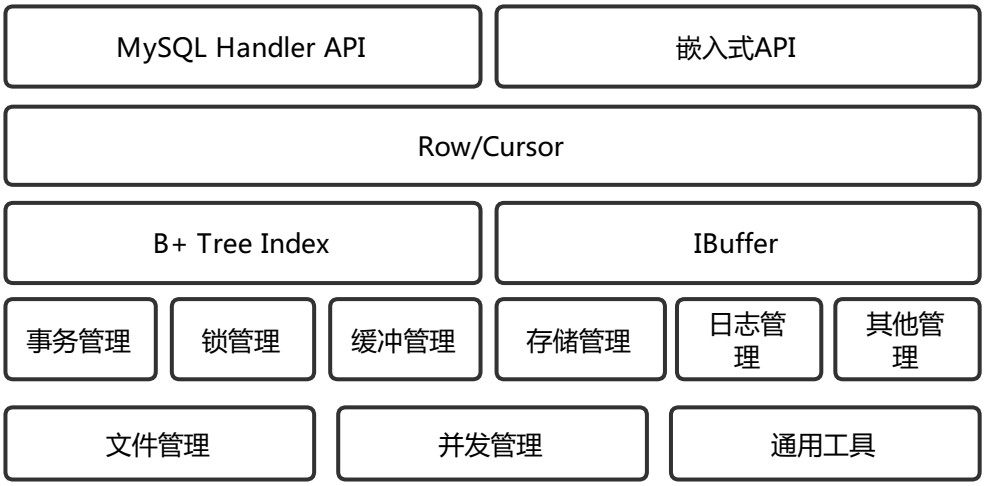
InnoDB 源码目录剖析

InnoDB 除了与 MySQL 层交互的部分为 C++实现外，其余的基本都是以 C 语言实现。还有很少量的汇编语言。源码结构非常清晰。每个模块一个单独的文件夹，文件夹下为.c 或者.cc 文件。而.h 文件统一放在 include 目录下。文件的命令规则为：模块名 0 子模块名。例如，关于 B+树索引模块的文件命名为 btr/btr0xxx.cc。文件名中，模块名部分就是文件夹的名字。有的源代码是将.h 和.c 放在一起，InnoDB 这一点与其他的不同。.h 文件在 include 目录下的文件名与 c 源程序文件夹中的.c 或.cc 的文件名对应。include 中除了.h 文件外还有.ic 文件。这类文件为每个模块定义了内联函数。打开任意一个.ic 文件，你会发现其中的函数都被 UNIV_INLINE 修饰。UNIV_INLINE 在 include/univ.i 中定义：

```
#define UNIV_INLINE static inline
```

InnoDB 为了达到工业级的稳定性要求，做了很多努力，性能上也做了很多优化，这也伴随着大量的代码，如果不先从宏观上把握其结构而一头扎进源码中，就会有盲人摸象的感觉。因此先把 InnoDB 的各个模块画出来：



最底层的是最基本的模块，文件管理封装了 InnoDB 对文件的各类操作。并发管理封装了各类 mutex 和 latch，通用工具实现了一些基本的数据结构和算法，例如：链表、哈希表等。

途中至底向上看，第二和三层是 InnoDB 的核心，也是一个存储引擎都包含的各种模块。理解了它们基本就理解了 InnoDB 的运行机理和一个存储引擎的实现方法。

图中最上边的两层为接口层，通过这些接口与存储引擎内部进行通信。InnoDB 可以不依赖于 MySQL 数据库，而作为一个嵌入式的数据库存在，因此还提供了嵌入式 API 的接口（这一部分不是我们学习的范畴）。

了解了 InnoDB 的整体结构，接下来看看 InnoDB 的源码目录。

对于 MySQL 5.7.18，InnoDB 的源码目录结构如下：

1、api：InnoDB 可以作为 SQL 语义的存储引擎，同时也可以用于嵌入式，而不是通过 SQL 接口访问，该目录中就是这些 API 的实现。我们不做学习。

2、btr：B+树以及 B+树索引等的实现。重点学习。

3、buf：任何一个存储引擎对内存的高效管理都是必不可少的，因为通过缓冲磁盘的数据减少磁盘 IO，从而提高性能。该目录就是缓冲区的管理（缓冲区很大一部分都是用来缓冲磁盘的数据的）。重点学习。

4、data：影响行的次要实用程序集合。

5、dict：数据字典相关的程序。重点学习。

6、eval（EVALUATING）：SQL 语句执行代价的评估。执行 SQL 存储过程是 InnoDB 的一个功能，但 MySQL 以其自己的方式处理存储过程，因此该目录中的程序不是非常重要。

7、fil：file 的简写，数据库文件的读取和写入的实现，与底层文件 IO 例程协作完成文件的读写。底层的文件 IO 例程在 os 文件夹中的 fil0fil.c 中。重点学习。

8、fsp：file space 的简写，作用与 fil 类似，也是用于文件的读写。重点学习。

9、fts：full text search 的简写，全文检索相关的实现。

10、fut：file utility 的简写，一些文件操作的使用例程。主要的文件操作和管理还是在 fil、fsp、fts 中。重点学习。

11、gis：R-tree 相关的实现。

12、ha：哈希表的实现。重点学习。

13、handler：与 MySQL 层通信的接口。重点学习。

14、ibuf：之前的插入缓冲区，现在的 change buffer 的实现。重点学习。

15、include：包含了 InnoDB 实现需要的各种头文件。重点学习。

16、lock：事务相关的锁的操作实现。重点学习。

17、log：InnoDB 所有日志相关的实现。例如重做日志。重点学习。

18、mach：mach0data.c 中有两个小程序用于读取压缩的 ulints（无符号长整数）

19、mem：通用内存池，即不在缓冲池（buf 目录中的实现）和日志池（log 目录中的实现）中的内存空间。重点学习。

20、mtr：*MINI-TRANSACTION* 的简写。被大多数其他程序组调用的小型事务例程。可以理解成低级实用程序集。

21、os：这是一组其他模块在需要使用操作系统资源时可以调用的实用程序。其中的实现将不同的操作系统的函数进行了封装，为其他模块提供了同一的语义（因为，例如 Linux 和 windows 中的打开文件的函数就不一样，该目录中的代码会屏蔽这些不同）。重点学习。

22、page：InnoDB 中的操作以 page（InnoDB 的页，不要和操作系统的页混淆）为单位，其中为页相关的实现。学习和研究的重点。重点学习。

23、pars：*PARSING* 的简写，该目录实现的功能是：输入一个包含 SQL 语句的字符串，并输出一个内存中的解析树。EVALUATING（目录 eval）程序将使用树进行 SQL 执行的代价评估。

通常的做法是使用 Bison 和 Flex 工具。pars0grm.c 是由 Bison 解析器从原始文件 pars0grm.y 生成的，而且 lexyy.c 是 Flex 生成的。

由于 InnoDB 本身是一个 DBMS，因为它支持嵌入式的 API。所以解析 SQL 语句很自然。但是在 MySQL / InnoDB 组合中，MySQL 处理大部分的解析。因此这些文件并不重要。

24、que：程序 que0que.c 表面上是关于包含 commit /rollback 语句的存储过程的执行。我认为这对于一般的 MySQL 用户来说并不重要。

25、read：事务实现时需要使用的一组例程。

26、rem：record manager，记录管理。学习的重点。

27、row：InnoDB 行的相关程序的实现，例如：InnoDB 行与 MySQL 行的转换，与行有关的维护活动，例如插入、更新、删除等等。学习的重点

28、srv：这是服务器读取初始配置文件，依据配置拆分线程并进行访问的实现。也是学习和研究的重点

29、sync：现代操作系统和 C 语言函数库都会实现互斥原语，但是有时候不高效，自己定制实现反而高效，该目录中就是同步与互斥相关的实现，以及死锁等同步问题的处理。学习的重点。

30、trx：InnoDB 事务的实现。学习和研究的重点

31、usr：一个用户可以有多个会话（会话是连接和断开连接之间发生的所有事情）。这是 InnoDB 用于跟踪会话 ID 和服务器/客户端消息传递的位置。这通常是 MySQL 的工作。因此，对于学习 InnoDB 也不是很重要。

32、ut：一些试用程序，不重要。

作者：许富博

限于本人水平有限，欢迎大家随时指正，联系方式：

xufubobo@gmail.com

xufubobo@163.com

1332841493@qq.com