# MMKV介绍

MMKV 分别代表的是 Memory Mapping Key Value（内存映射键值），是基于 mmap 内存映射的 key-value 组件，底层序列化/反序列化使用 protobuf 实现，性能高，稳定性强，能够很好的取代SP存储。

mmap(一种内存映射文件的方法) 一个文件或者其它对象映射进内存，文件被映射到多个页上，如果文件的大小不是所有页的大小之和，最后一个页不被使用的空间将会清零。

将磁盘上文件的一部分或整个文件或者其它对象映射到进程(应用程序)的地址空间，实现文件磁盘地址和进程虚拟地址空间中一段虚拟地址的一一对应关系，从而应用程序可以用访问内存的方式访问磁盘文件；

protobuf(Google Protocol Buffers)是Google提供一个具有高效的协议数据交换格式工具库(类似Json)，但相比于Json，Protobuf有更高的转化效率，protobuf的空间效率是JSON的2-5倍，时间效率要高，对于数据大小敏感，传输效率高的模块可以采用protobuf库。

Protobuf和Xml、Json序列化的方式不同，采用了二进制字节的序列化方式，用字段索引和字段类型通过算法计算得到字段之前的关系映射，从而达到更高的时间效率和空间效率，特别适合对数据大小和传输速率比较敏感的场合使用。

# 优势：

1. 采用protobuf数据存储，mmap内存映射，更高效。
2. 支持多进程并发。
3. 数据加密。

(1)数据格式及更新范围优化：SharedPreferences 采用 xml 数据存储，每次读写操作都会全局更新；mmkv采用 protobuf 数据存储,对数值进行编码/解码，更紧密，支持局部更新.

(2)文件耗时操作优化：mmkv使用mmap与文件保持内存同步，采用 MMap 内存映射的方式取代 I/O 操作，使用 0拷贝技术提高更新速度，实现最佳性能。

(3)多进程并发:MMKV支持进程之间的并发读写访问。

(4)易于使用的:你可以随时使用mmkv。所有的更改都会立即保存，不需要同步，也不需要apply调用。

(5)小。少数几个文件:MMKV包含进程锁、编码/解码帮助程序和mmap逻辑等等。很整洁。大约60K的二进制大小:MMKV在每个架构上增加了大约60K的应用程序大小，而压缩(apk)时增加的就少多了。

(6)跨进程状态同步：SharedPreferences 为了线程安全不支持跨进程状态同步；MMKV 通过 CRC 校验 和文件锁 flock 实现跨进程状态更新；

（7）相比于替代SP的优点与需求：数据加密，多进程共享，匿名内存，效率更高。

（8）MMKV 通过 flock 文件锁和 CRC 校验支持多进程的读写操作；