## 结构体kb\_option\_t

typedef struct \_kb\_options {

uint32\_t version; //!< Should be set to #kKbootApiVersion.

uint8\_t \*buffer; //!< Caller-provided buffer used by Kboot.

uint32\_t bufferLength;

kb\_operation\_t op;

union {

//! Settings for #kKbootAuthenticate operation.

kb\_authenticate\_t authSB;

//! Settings for #kKbootAuthenticate operation.

//! Settings for #kKbootLoadSB operation.

kb\_load\_sb\_t loadSB;

};

} kb\_options\_t;

typedef enum \_kb\_operation {

kRomAuthenticateImage = 1, //!< Authenticate a signed image.

kRomLoadImage = 2, //!< Load SB file.

kRomOperation = 3,

} kb\_operation\_t;

## 结构体kb\_authenticate\_t

typedef struct \_kb\_authenticate {

uint32\_t profile;

uint32\_t minBuildNumber;

uint32\_t maxImageLength;

uint8\_t \*userRHK;

} kb\_authenticate\_t;

## 结构体kb\_load\_sb\_t

typedef struct \_kb\_load\_sb {

uint32\_t profile;

uint32\_t minBuildNumber;

uint32\_t overrideSBBootSectionID;

uint8\_t \*userSBKEK;

uint32\_t regionCount;

const kb\_region\_t \*regions;

} kb\_load\_sb\_t;

## 从内存查看memoryMap

1.从0x0880001c查看gbootloaderTree的地址

2.gbootloaderTree地址数4个4byte长度，得到runtimeContext的地址，ROM code里面

.runtimeContext = &g\_bootloaderContext,

3.从2得到的地址，数2个4byte长度，得到.memoryMap = g\_memoryMap,

4. g\_memoryMap 是一个结构体数组，从g\_memoryMap的地址可以看到第一个结构体数组的值。也就是M4 flash的startaddress，endaddress

