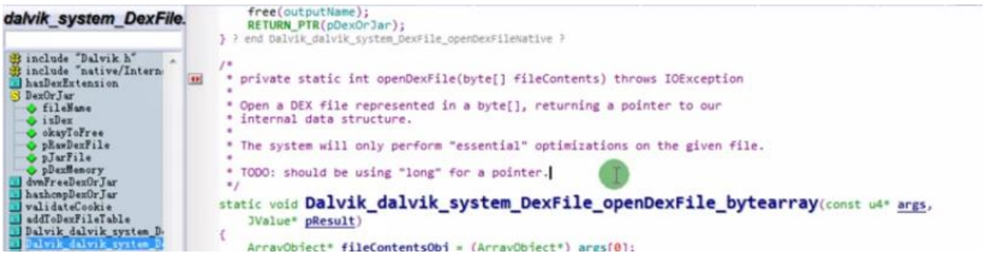
1. **Dex 加载流程**

   Vm->native->dalvik_system_DexFile->openDexFile

   openDexFile，读取内存中的 Dex 文件数据并加载

   Dalvik_dalvik_system_DexFile_openDexFile_bytearray



```
    free(outputName);
    RETURN_PTR(pDexOrJar);
} ? end Dalvik_dalvik_system_DexFile_openDexFileNative ?

/*
 * private static int openDexFile(byte[] fileContents) throws IOException
 *
 * Open a DEX file represented in a byte[], returning a pointer to our
 * internal data structure.
 *
 * The system will only perform "essential" optimizations on the given file.
 *
 * TODO: should be using "long" for a pointer.
 */
static void Dalvik_dalvik_system_DexFile_openDexFile_bytearray(const u4* args,
    JValue* pResult)
{
    ArrayObject* fileContentsObj = (ArrayObject*) args[0];
```

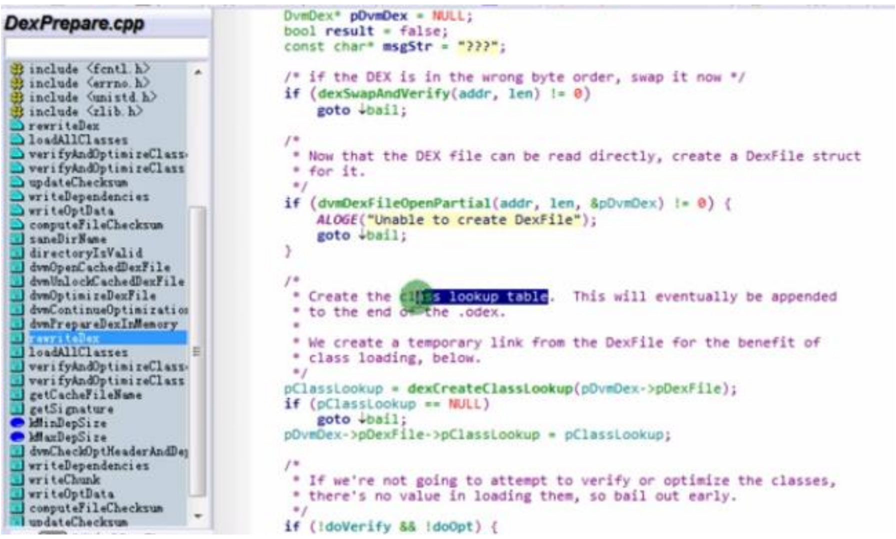
   1) 转换存储的 dex 格式为执行的 dex 格式

   dvmRawDexFileOpenArray(pBytes, length, &pRawDexFile) //将 byte 数据转换成 Android 可以加载的数据

   2) 添加到 gDvm 中

   addToDexFileTable(pDexOrJar);

   脱壳点之一的函数：dvmDexFileOpenPartial



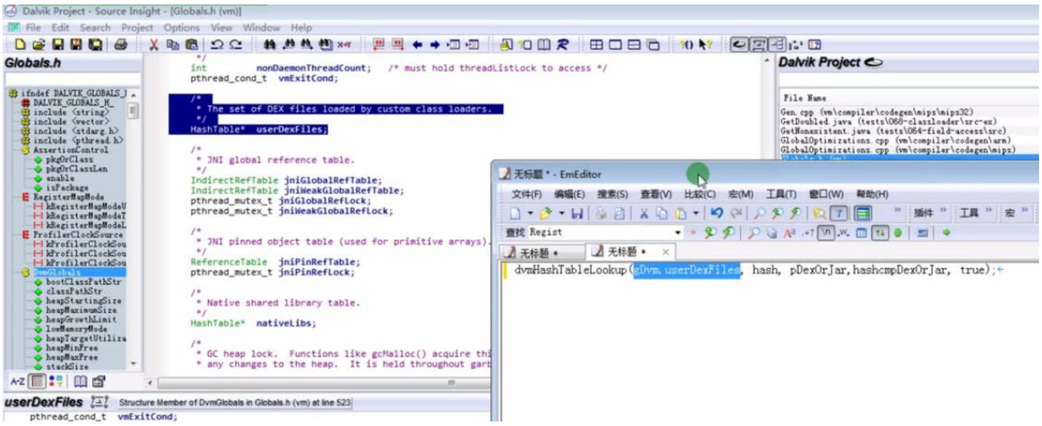
```
    DvmDex* pDvmDex = NULL;
    bool result = false;
    const char* msgStr = "???";

    /* if the DEX is in the wrong byte order, swap it now */
    if (dexSwapAndVerify(addr, len) != 0)
        goto bail;

    /*
     * Now that the DEX file can be read directly, create a DexFile struct
     * for it.
     */
    if (dvmDexFileOpenPartial(addr, len, &pDvmDex) != 0) {
        ALOGE("Unable to create DexFile");
        goto bail;
    }

    /*
     * Create the class lookup table.  This will eventually be appended
     * to the end of the .odex.
     *
     * We create a temporary link from the DexFile for the benefit of
     * class loading, below.
     */
    pClassLookup = dexCreateClassLookup(pDvmDex->pDexFile);
    if (pClassLookup == NULL)
        goto bail;
    pDvmDex->pDexFile->pClassLookup = pClassLookup;

    /*
     * If we're not going to attempt to verify or optimize the classes,
     * there's no value in loading them, so bail out early.
     */
    if (!doVerify && !doOpt) {
```

2. 壳实现加载流程
   1) 内存中解密 dex 函数
   2) 将 dex 存储结构转换为执行结构
   3) 添加到 gDvm 中

      部分壳是自己实现了该功能，部分壳是调用系统的函数，一般这里可以作为一个脱壳点
   4) 抹去 dex 存储结构中的有效数据
3. 脱壳思路

   dvmHashTableLookup(gDvm userDexFiles, hash, pDexOrJar, hashcmpDexOrJar, true);

   加载后的 Dex 数据会添加到 userDexFiles 哈希表中，通过遍历 userDexFiles 获取到当前所有已经加载的 Dex 文件数据


Dalvik Project - Source Insight - [Globals.h (vm)]
File Edit Search Project Options View Window Help
Globals.h
Dalvik Project
File Name
userDexFiles
Structure Member of DvmGlobals in Globals.h (vm) at line 523
无标题 * - EmEditor
文件(F) 编辑(E) 搜索(S) 查看(V) 比较(C) 宏(M) 工具(T) 窗口(W) 帮助(H)
插件 工具 宏
查找 Regist
无标题 •
无标题 •

```
ifndef DALVIK_GLOBALS_H
DALVIK_GLOBALS_H_
include <string>
include <vector>
include <stdarg.h>
include <pthread.h>
AssertionControl
    pkgOrClass
    pkgOrClassLen
    enable
    isPackage
RegisterMapMode
    kRegisterMapModeU
    kRegisterMapModeT
    kRegisterMapModeL
ProfilerClockSource
    kProfilerClockSou
    kProfilerClockSou
    kProfilerClockSou
DvmGlobals
    bootClassPathStr
    classPathStr
    heapStartingSize
    heapMaximumSize
    heapGrowthLimit
    lowMemoryMode
    heapTargetUtiliza
    heapMinFree
    heapMaxFree
    stackSize
```

```
     */
    int             nonDaemonThreadCount;   /* must hold threadListLock to access */
    pthread_cond_t  vmExitCond;

    /*
     * The set of DEX files loaded by custom class loaders.
     */
    HashTable*  userDexFiles;

    /*
     * JNI global reference table.
     */
    IndirectRefTable jniGlobalRefTable;
    IndirectRefTable jniWeakGlobalRefTable;
    pthread_mutex_t jniGlobalRefLock;
    pthread_mutex_t jniWeakGlobalRefLock;

    /*
     * JNI pinned object table (used for primitive arrays).
     */
    ReferenceTable  jniPinRefTable;
    pthread_mutex_t jniPinRefLock;

    /*
     * Native shared library table.
     */
    HashTable*  nativeLibs;

    /*
     * GC heap lock.  Functions like gcMalloc() acquire thi
     * any changes to the heap.  It is held throughout garb
```

```
pthread_cond_t  vmExitCond;
```

```
Gen.cpp (vm\compiler\codegen\mips\mips32)
GetDoubled.java (tests\068-classloader\src-ex)
GetNonexistent.java (tests\064-field-access\src)
GlobalOptimizations.cpp (vm\compiler\codegen\arm)
GlobalOptimizations.cpp (vm\compiler\codegen\mips)
```

```
dvmHashTableLookup(gDvm.userDexFiles, hash, pDexOrJar,hashcmpDexOrJar, true);
```