

LOCAL_PRELINK_MODULE

Prelink 利用事先链接代替运行时链接的方法来加速共享库的加载，它不仅可以加快启动速度，还可以减少部分内存开销，是各种 Linux 架构上用于减少程序加载时间、缩短系统启动时间和加快应用程序启动的很受欢迎的一个工具。程序运行时的动态链接尤其是重定位(relocation)的开销对于大型系统来说是很大的。

动态链接和加载的过程开销很大，并且在大多数的系统上，函数库并不会常常被更动，每次程序被执行时所进行的链接动作都是完全相同的，对于嵌入式系统来说尤其如此。因此，这一过程可以改在运行时之前就可以预先处理好，即花一些时间利用 Prelink 工具对动态共享库和可执行文件进行处理，修改这些二进制文件并加入相应的重定位等信息，节约了本来在程序启动时的比较耗时的查询函数地址等工作，这样可以减少程序启动的时间，同时也减少了内存的耗用。

Prelink 的这种做法当然也有代价：每次更新动态共享库时，相关的可执行文件都需要重新执行一遍 Prelink 才能保证有效，因为新的共享库中的符号信息、地址等很可能与原来的已经不同了，这就是为什么 android framework 代码一改动，这时候就会导致相关的应用程序重新被编译。

这种代价对于嵌入式系统的开发者来说可能稍微带来一些复杂度，不过好在对用户来说几乎是可以忽略的。

变量设置为 false 那么将不做 prelink 操作

```
LOCAL_PRELINK_MODULE := false
```

默认是需要 prelink 的，同时需要在 build/core/prelink-linux-arm.map 中加入

```
libhellod.so      0x96000000
```

这个 map 文件是制定动态库的地址的，在前面注释上面有一些地址范围的信息，注意库与库之间的间隔数，如果指定不好的话编译的时候会提示说地址空间冲突的问题。另外，注意排序，这里要把数大的放到前面去，按照大小降序排序。

解析 LOCAL_PRELINK_MODULE 变量

```
build/core/dynamic_binary.mk:94:ifeq ($(LOCAL_PRELINK_MODULE),true)
ifeq ($(LOCAL_PRELINK_MODULE),true)
$(prelink_output): $(prelink_input) $(TARGET_PRELINKER_MAP) $(APRIORI)
    $(transform-to-prelinked)
transform-to-prelinked定义:
./build/core/definitions.mk:1002:define transform-to-prelinked
define transform-to-prelinked
@mkdir -p $(dir $@)
@echo "target Prelink: $(PRIVATE_MODULE) ($@)"
$(hide) $(APRIORI) /
        --prelinkmap $(TARGET_PRELINKER_MAP) /
        --locals-only /
        --quiet /
        $/build/tools/apriori"
```

参考文档：

动态库优化——Prelink（预连接）技术

<http://www.eefocus.com/article/09-04/71629s.ht>