

Steps to Clean and Validate Order Files



Sharjeel Khan
PhD Student at Georgia Institute of Technology
smkhan@gatech.edu

Special Thanks: Aditya Kumar and Android LLVM Team

Order File Overview

- Order files are text files containing symbols representing functions names.
- Linker (lld) uses order files to layout functions in a specific order.
- Ordered binaries reduce page faults and improve a program's launch time (>5% improvement in launch time [1])
- Different methods to collect order files:
 - GProf [2]
 - Sanitizer Coverage [3]
 - LLVM Order File Flags [4]
 - LLVM IRPGO [6]

```
main
_Z9quickSortPiii
_Z5mergePiiii
_Z9mergeSortPiii
```

Order File Example

Order File Overview

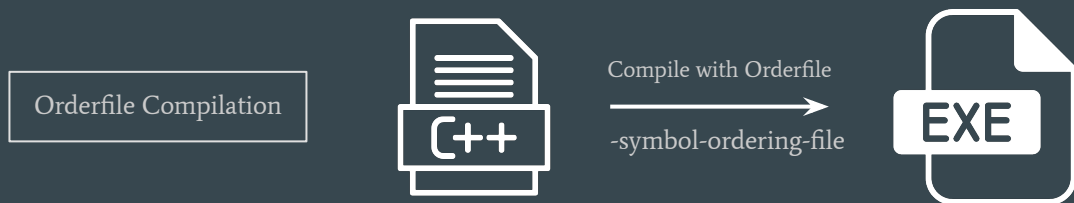
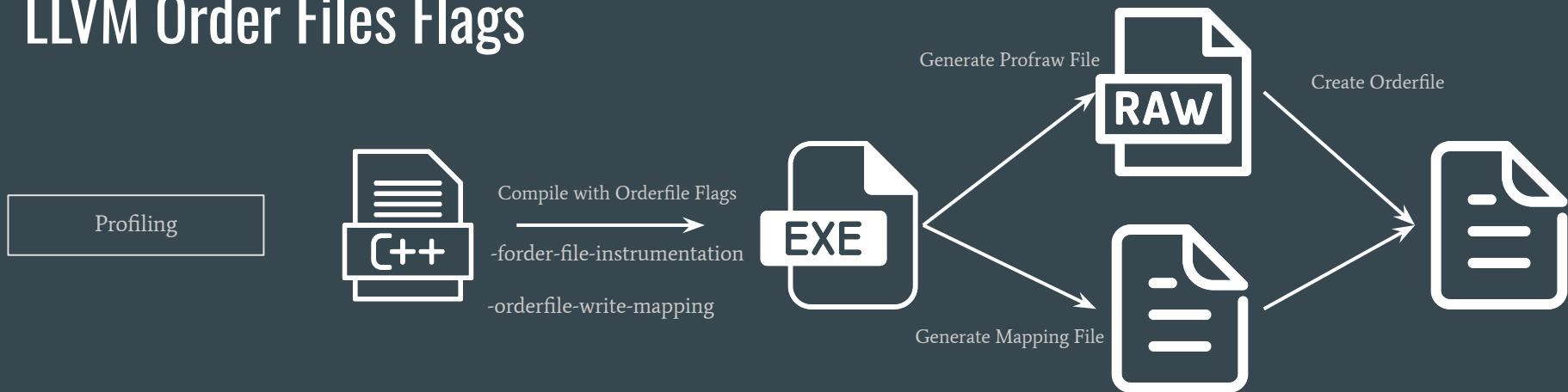
```
0000000000401000 T _init
0000000000401040 T _start
0000000000401070 T _dl_relocate_static_pie
0000000000401080 t deregister_tm_clones
00000000004010b0 t register_tm_clones
00000000004010f0 t __do_global_dtors_aux
0000000000401120 t frame_dummy
0000000000401130 T main
0000000000401180 T _Z5mergePiiii
0000000000401560 T _Z9mergeSortPiii
00000000004015c0 T _Z5splitPiii
0000000000401680 T _Z9quickSortPiii
0000000000401780 T __libc_csu_init
00000000004017f0 T __libc_csu_fini
00000000004017f8 T _fini
```

Before Order File

```
0000000000201710 T main
0000000000201760 T _Z5splitPiii
0000000000201820 T _Z9quickSortPiii
0000000000201920 T _Z5mergePiiii
0000000000201d00 T _Z9mergeSortPiii
0000000000201d60 T _start
0000000000201d90 t _dl_relocate_static_pie
0000000000201da0 t deregister_tm_clones
0000000000201dd0 t register_tm_clones
0000000000201e10 t __do_global_dtors_aux
0000000000201e40 t frame_dummy
0000000000201e50 T __libc_csu_init
0000000000201ec0 T __libc_csu_fini
0000000000201ec8 t _init
0000000000201ee4 t _fini
```

After Order File

LLVM Order Files Flags



Order File Cleaning

- Profiling can lead to more functions in order files than needed
 - Shared Library functions not used in application's compilation
 - Functions after the startup point
- Functions not found can be ignored in LLVM with flags
(`--no-warn-symbol-ordering`)
- Removing these functions from order files will lead to better ordered binaries

Many Order Files

- Order files can differ across many executions
- How do you choose the order file for compilation?
 - Compile with each order file and run regression tests (Takes too long)
 - Merge order files
- Different groups use different merging algorithms
 - Android uses a DFS method [5]
 - Meta uses a balance partitioning technique [6]
- Merging might lead to bad order files so need validation

Validation Criteria

- No duplicates of any functions if merging
- Symbols that should (not) be present in orderfile [Program specific]
- Minimum number of symbols to make an orderfile good for page layout purposes
- Defining a partial order priority that needs to be in the orderfile

Partial Order Priority Check

- Checks all functions in the order file are in the correct order based on some partial order
- This partial order can be taken from control-flow graphs

```
main
foo
bar
baz
```

main, baz



convert, split



foo, temp, baz



baz, bar



bar, baz, main



baz, temp, foo



Thank You

References

[1]<https://docs.emergetools.com/docs/launch-booster-ios>

[2]<https://developer.apple.com/library/archive/documentation/Performance/Conceptual/CodeFootprint/Articles/ImprovingLocality.html>

[3]<https://medium.com/@michael.eisel/improving-app-performance-with-order-files-c7fff549907f>

[4]<https://developer.android.com/ndk/guides/orderfile>

[5]https://android-review.googlesource.com/c/toolchain/llvm_android/+2755594

[6]<https://reviews.llvm.org/D147812>

Thanks for icons:

<https://www.flaticon.com/authors/secret-studio> (Cross and Tick Images)

<https://www.flaticon.com/authors/kiranshastry> (C File Image)

<https://www.flaticon.com/authors/freepik> (Raw File)

<https://www.flaticon.com/authors/kliwir-art> (Executable, Text File)