

# MCLinker

## Design and Implementation of a Fragments-based Target-independent Linker

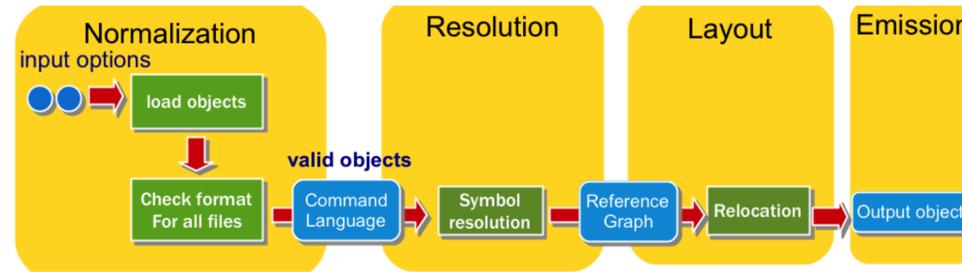
<http://code.google.com/p/mclinker>

### MCLinker

- MCLinker is a **full-fledged system linker**
- The architecture is based on **LLVM Machine Code (MC) layer**
- MCLinker is designed for **on device linking**
- MCLinker is **fast, small with low memory usage**
- MCLinker is a candidate linker of **Android** and **BSD** standard systems
- MCLinker provides its own **intermediate representation (IR)** for efficient transformation and analysis

### Linking Stages

- **Normalization** – parse the command line language
- **Resolution** – resolve symbols
- **Layout** – relocate instructions and data
- **Emission** – emit file by various formats



### Comparison of Modern Linkers

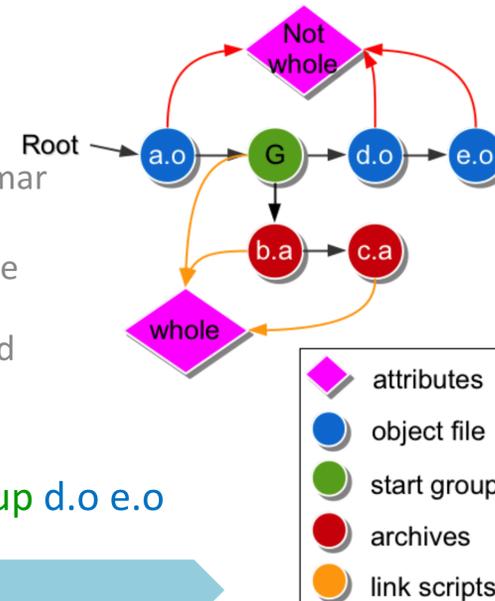
	GNU ld	Google gold	MCLinker
<b>License</b>	GPLv3	GPLv3	<b>UIUC BSD-Style</b>
<b>Target Platform</b>	All Linux mainstream devices	ARM, X86, X86_64, (Mips, SPARC)	ARM, X86, X86_64, Mips (,X32, Mips64 and Hexagon)
<b>Object Format</b>	COFF, a.out, ELF	ELF only	ELF, <b>extensible</b>
<b>Line of Code</b>	500+K	100+K	50+K
<b>Performance</b>	-	Fast	<b>Fastest</b> Steadily x2 than GNU ld, x1.3 than Google gold
<b>Intermediate Representation</b>	The BFD library for reference graph	None	<b>Command line language and reference graph</b>

### Intermediate Representation (IR)

- MCLinker is the first \*ELF linker to provide an **intermediate representation (IR)**
- MCLinker provides IR on two levels
  - Linker Command Line Language
  - Fragment-Reference Graph
- Fragment is the basic linking unit, it can be
  - A section (coarse granularity)
  - A block of code or instructions (middle granularity)
  - An individual symbol and its code/data (fine granularity)
- MCLinker can **trade linking time for the output quality**

### Linker Command Line Language

- The meaning of a **option** depends on
  - their positions
  - the other options
  - Some options have its own grammar
- The **input File Tree**
  - Each vertex represents an input file with its attributes
  - Linkers resolve symbols by DFS and merge sections by BFS

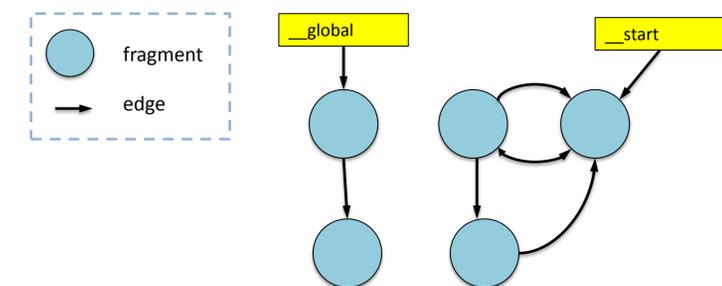
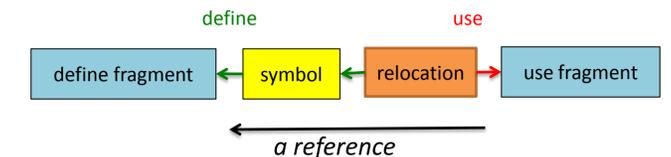


Example:

```
$ ld a.o -start-group b.a c.a -end-group d.o e.o
```

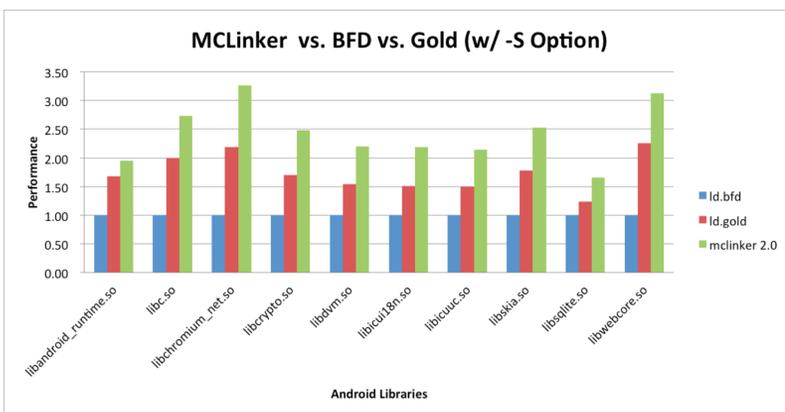
### Fragment-Reference Graph

- A **fragment** is a block of instruction code or data in a module
- A **reference** is a symbolic linkage between two fragments
- A **relocation** represents an **use-define relationship** between two fragments
- Optimization: fragment stripping, branch optimization, low-level inlining



### Comparison of Linking Speed

### Future Objectives



- MCLinker has successfully linked **Android** and **BSD** base system
- The next step is to link Linux kernel and provide more processors support
- MCLinker is contributed by many people worldwide

