

swviz: Software Visualizer  
Harsh Vardhan Dwivedi

Qualcomm Innovation Center

hdwivedi@codeaurora.org  
[https://github.com/hdwiv/  
swviz](https://github.com/hdwiv/swviz)

# What is swviz?

## Generate and query full program call graph



Clang

**Linker**

**swviz**

Qualcomm Innovation Center (QuIC) Inc.

# Purpose of swviz?

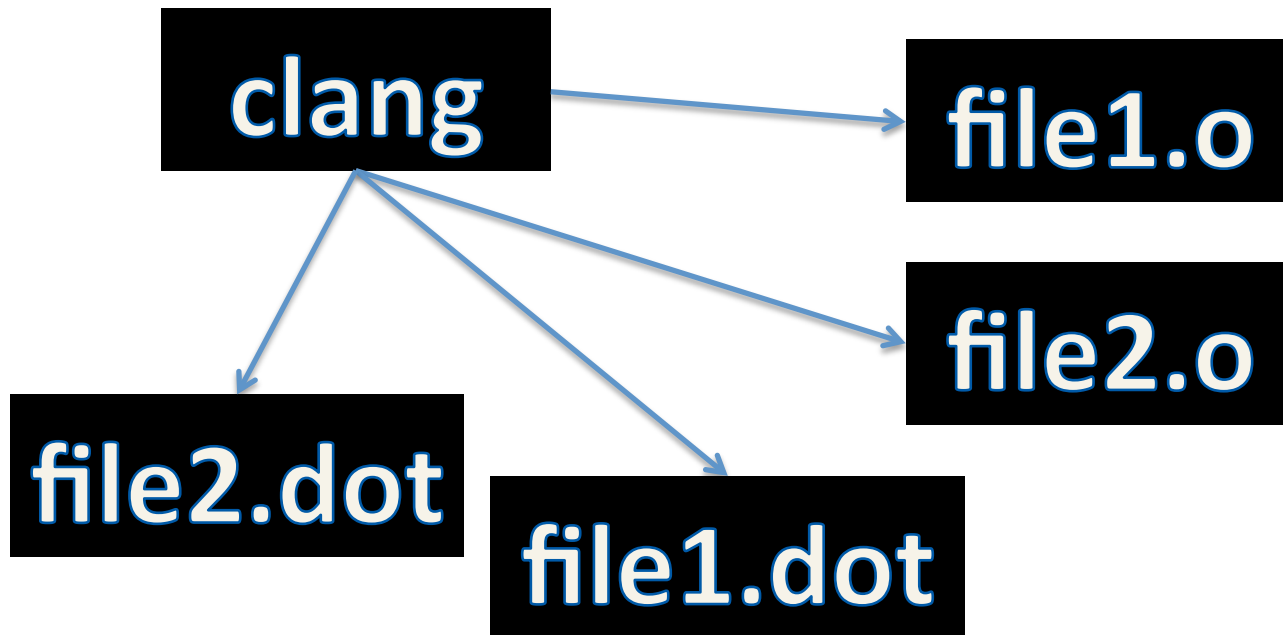
- Triaging bugs
  - Figure out quickly all the locations a function is getting called from
  - Get list of all possible paths between two functions
  - Anything more that can be gleaned by querying the callgraph
- Helps to quickly understand the overall structure of any code

# How does it really work?

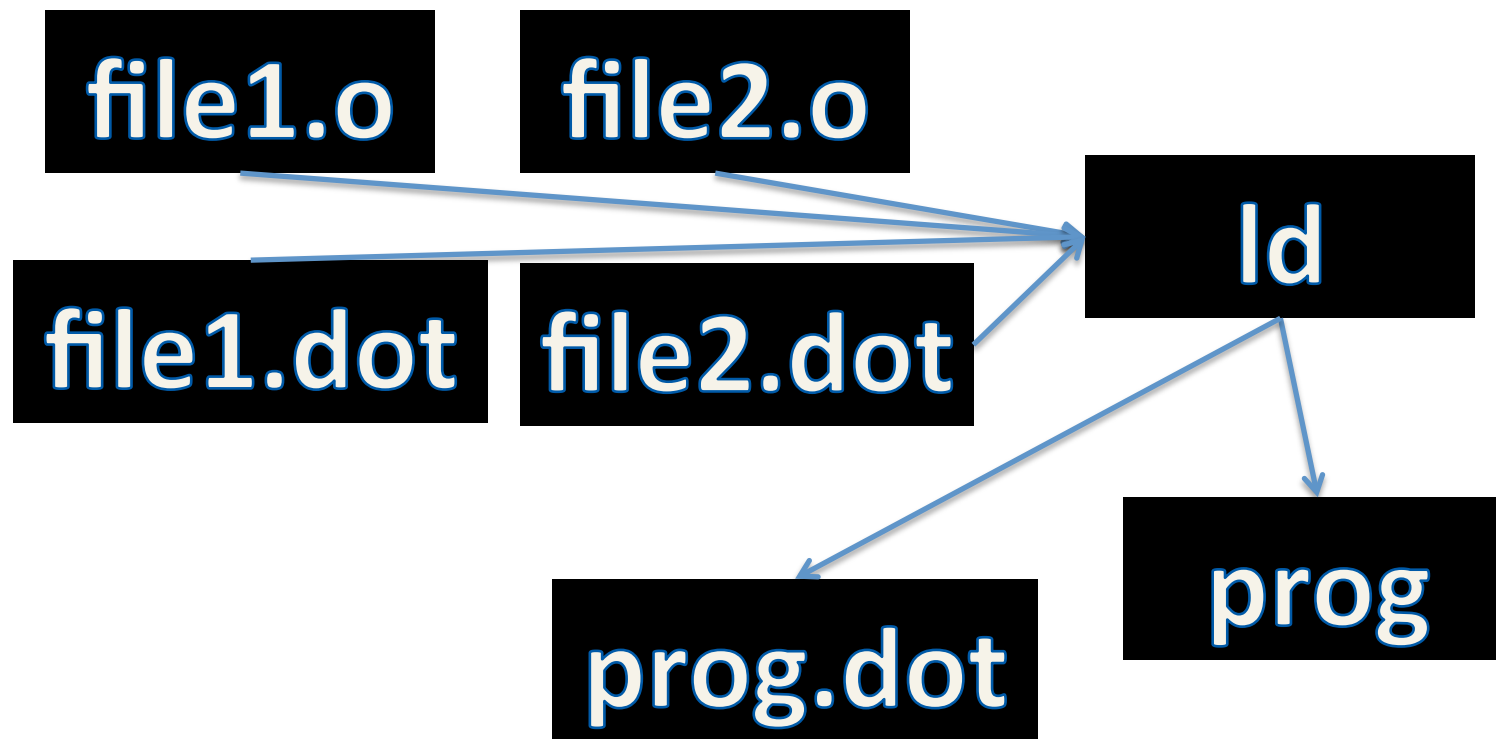
- Use clang callgraph generation option to generate call graph of each individual translation unit
- Construct a full program call graph by modifying the linker to generate combined call graph.
- Since the linker combines the object files anyway, it is ideal to invoke our graph stitching from the linker

# How does it really work?

– Eg: `clang file1.c file2.c -o prog`



# How does it really work?



# How does it really work?

- Use Python Networkx library to load and analyze full program call graph
- Use a web front-end to query the program call graph

# Questions?

- Let the tools do the “grep”ing and let the devs do the analysis
- Piggyback on the linker since it already combines disparate units to produce your program, no need to reinvent major logic to stitch individual graphs
- Follow swviz at: <https://github.com/hdwiv/swviz>
- Send any questions:
- [hdwivedi@codeaurora.org](mailto:hdwivedi@codeaurora.org)
- Profit!!