Advanced Programming in the UNIX Environment

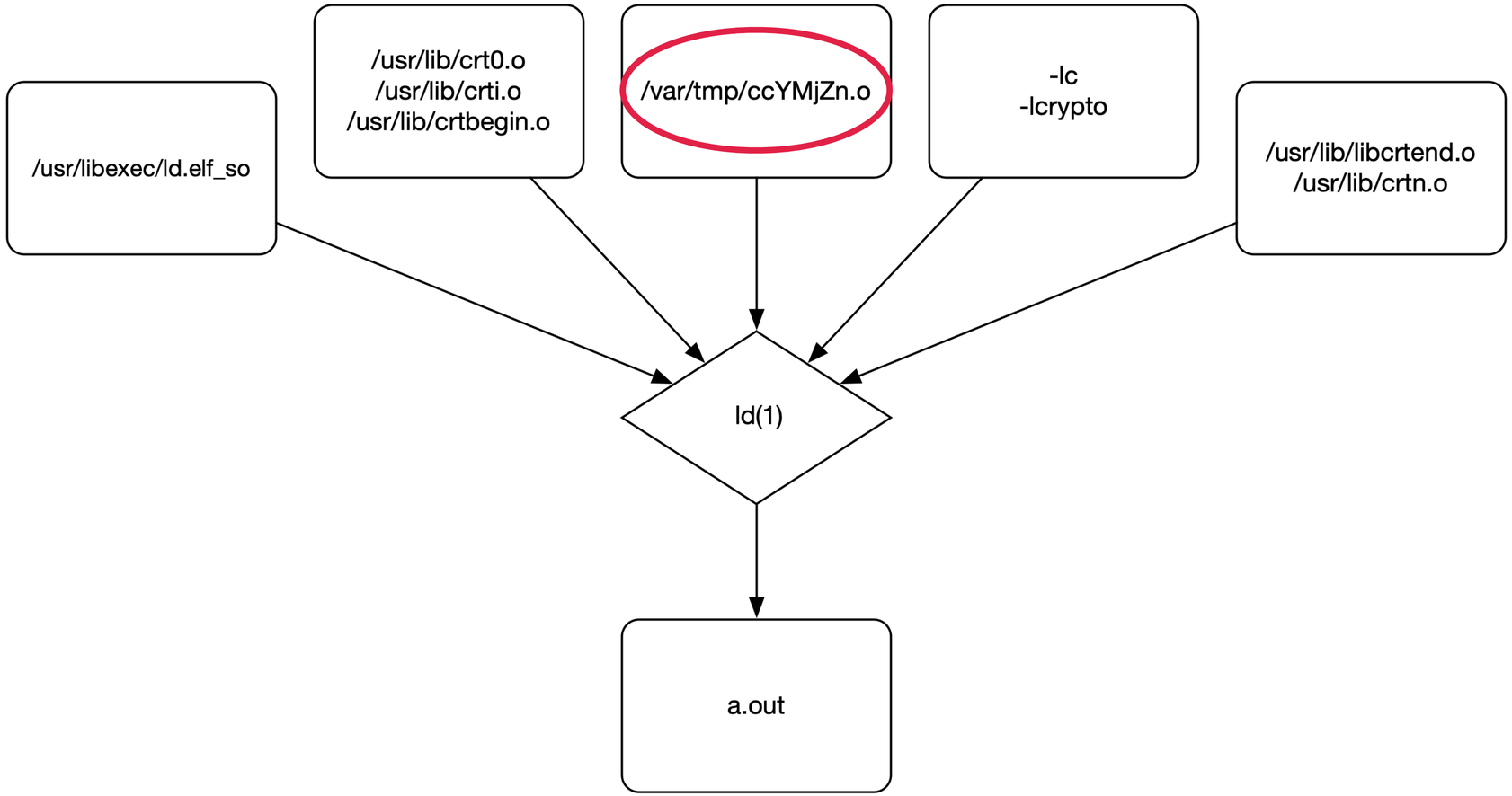
Week 05, Segment 6: Unix Development Tools: The Compiler Chain, Part IV

Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/631/

The GNU Compiler Collection

The compiler chain or driver usually performs preprocessing (*e.g.* via `cpp(1)`), compilation (`cc(1)`), assembly (`as(1)`) and linking (`ld(1)`).




Terminal — 80×24
MPC version 1.1.0, isl version none
GGC heuristics: --param ggc-min-expand=99 --param ggc-min-heapsize=131015
Compiler executable checksum: 64237a38f87656fa051aa4784bb3316c
COLLECT_GCC_OPTIONS='-Wall' '-Werror' '-Wextra' '-v' '-mtune=nocona' '-march=x86-64'
as -v -o /tmp//ccDHCm6E.o /tmp//ccGSitxl.s
GNU assembler version 2.31.1 (x86_64--netbsd) using BFD version (NetBSD Binutils nb1) 2.31.1
COMPILER_PATH=/usr/libexec/
LIBRARY_PATH=/usr/lib/
COLLECT_GCC_OPTIONS='-Wall' '-Werror' '-Wextra' '-v' '-mtune=nocona' '-march=x86-64'
ld -plugin /usr/libexec/liblto_plugin.so -plugin-opt=/usr/libexec/lto-wrapper -plugin-opt=-fresolution=/tmp//ccaScqYN.res -plugin-opt=-pass-through=-lgcc_s -plugin-opt=-pass-through=-lgcc -plugin-opt=-pass-through=-lc -plugin-opt=-pass-through=-lgcc_s -plugin-opt=-pass-through=-lgcc --eh-frame-hdr -dc -dp -e _start -dynamic-linker /usr/libexec/ld.elf_so /usr/lib/crt0.o /usr/lib/crti.o /usr/lib/crtbegin.o /tmp//ccDHCm6E.o --as-needed -lgcc_s --no-as-needed -lgcc -lc --as-needed -lgcc_s --no-as-needed -lgcc /usr/lib/crtend.o /usr/lib/crtn.o
COLLECT_GCC_OPTIONS='-Wall' '-Werror' '-Wextra' '-v' '-mtune=nocona' '-march=x86-64'
apue$ ./a.out
avocado: great on anything.
apue$

The GNU Compiler Collection

The compiler chain or driver usually performs preprocessing (*e.g.* via `cpp(1)`), compilation (`cc(1)`), assembly (`as(1)`) and linking (`ld(1)`).

- `cc(1)` drives the entire compilation process
- flags passed to `cc(1)` may then be passed on to the tools it invokes
- order of the command line flags may matter
- built-in defaults and environment variables may further influence the behavior
- to save typing and to build more complex projects, use `make(1)`