DWARF & ELF Utilities

1. Documentation for libdw &
2. Docs libelf (but that is much easier) - just what is different than generic libelf
3. Thread safety and async safety info for functions
    a. Iteration over CU DIEs, FDEs and CIEs threadsafe
    b. Processing DWARF line info (will be used from cilk)
    c. Iterating over the DWARF tree -- no static state
    d. Will allow API changes.
4. .eh_frame support
5. DWARF producer - something that works at the context of the reader. So when they mutate the code they can augment it in the libdw data structures and then re-emit the DWARF.
6. DWARF5 whatever the compilers emit.
7. Suggested trying DWZ as a validation test

Cross validating DWARF structures using binary analysis

From software diversity - project

Stack frame analysis - If we have complete understanding of sections of code that access variables on the stack then we can manipulate the code and the order of variables on the stack frame to create a new variant of the function adding diversity.

1. Stack structure
2. Local variable locations.
3. Function bounds
4. Much of this is done in the data flow analysis.
5. DWARF over approximates -- would rather truth vs. enclosing approximation
    a. like eliding the last couple of instructions which and unwind recipe doesn't work.
    b. variable spilled to stack not represented in DWARF
    c. Would prefer more location lists vs. nested ranges for variable scoping because it introduces the possibility of erroneous interpretation.
6. Check EULA on commercial compilers