MathWorks Compiler Course – Day 9

- Interpreter

MathWorks Compiler Course – Day 9

- Interpreter
  - Source interpretation

MathWorks Compiler Course – Day 9

- Interpreter
  - Source interpretation
  - Tree interpretation

MathWorks Compiler Course – Day 9

- Interpreter
  - Source interpretation
  - Tree interpretation
  - Making P-code
    - From shift/reduce sequence
    - From tree

MathWorks Compiler Course – Day 9

- Interpreter
  - Source interpretation
  - Tree Interpretation
  - Making P-code
  - Interpreting P-Code

MathWorks Compiler Course – Day 9

- Interpreter
- JIT

MathWorks Compiler Course – Day 9

- Interpreter
- JIT
  - Pcode driven
  - Directly from source

MathWorks Compiler Course – Day 9

- Interpreter
- JIT
- Hybrid JIT-Interpreter
  - Switches between machine code and Pcode
  - Compile time vs. execution efficiency

MathWorks Compiler Course – Day 9

- Interpreter
- JIT
- Hybrid JIT-Interpreter
- Compiler
  - Module at a time
  - Linkers

MathWorks Compiler Course – Day 9

- Interpreter
- JIT
- Hybrid JIT-Interpreter
- Compiler
- Code Optimization

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifying meaning

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifiying meaning
  - “as-if” rule

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifying meaning
  - “as-if” rule
  - Cache aware

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifying meaning
  - “as-if” rule
  - Cache aware
  - Local optimization
    - Constant folding
    - Peephole optimization
    - Identities

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifying meaning
  - “as-if” rule
  - Cache aware
  - Local optimization
  - Non-local optimization
    - Common subexpression elimination
    - Hoisting

MathWorks Compiler Course – Day 9

- Code Optimization
  - Specifying meaning
  - “as-if” rule
  - Cache aware
  - Local optimization
  - Non-local optimization
  - Heavy-duty optimization
    - Flow analysis
    - Register allocation