COMPILING AND DEFINING WORDS IN CREOLE FORTH

JOSEPH M. O'CONNOR

MAY 2025

OVERVIEW

- Defining words – have two actions ; one at compile time, another at run time.
- Compiling words – execute at compile time, usually for the purposes of generating code into the parameter field.
- In both cases, there is a difference between compile-time code and run-time code, which can be tricky to visualize.

QUINTESSENTIAL DEFINING WORDS

- : CONSTANT CREATE , DOES> @ ;
- : VARIABLE CREATE , DOES> NOP ;

SOME COMPILING WORDS

- DO…LOOP/LOOP+ - Executes code in between a definite number of times.
- BEGIN…UNTIL– Executes code in between until some specified condition is met.
- IF…ELSE…THEN – conditional execution

POINTS TO REMEMBER

- Defining words and compiling words both have separate compile-time and run-time actions.
- Compiling words are used most often when setting up control structures.
- Defining words are a method of creating a new compiler.
- Some consider defining words to be an early form of object-oriented programming due to words inheriting functionality from previously defined words and the grouping of data and functionality together.

DEFINING WORD EXAMPLE – CONSTANT

- : CONSTANT CREATE , DOES> @ ;
- Compile time 1 : tokens for the primitives doCreate, doComma, compileDoes, and doFetch are placed in the parameter field.
- Compile time 2 : 3 CONSTANT THREE – compiles 3 in the parameter field, followed by its own address token* and the runtime code. Code field is set to doDoes.
- Run time : Puts the index field of the current word onto the stack has doColon execute @. This retrieves the value in the first member of the Parameter field (3), then pushes it onto the stack.

COMPILING WORDS EXAMPLE – IF ELSE THEN

- Compile Time:
- (1) compileIf
- 1. Adds the doZeroBranch (0BRANCH) code and a placeholder -1 to store the address to jump to.
- 2. Pushes the location of the placeholder onto the stack.
- (2) compileElse
- 1. Adds the doJump (JUMP) primitive and a -1 placeholder next to it.
- 2. Pops the position of the zero branch address to jump to off the stack,
- 3. Stores the location to jump to in that position.
- 4. Concludes with pushing the jump address location next to the jump command onto the stack.
- (3) compileThen
- 1. [Pops the location information off the stack and puts it in the -1 placeholder,
- 2. adds a doThen

COMPILING WORDS EXAMPLE – IF ELSE THEN

Run Time:

- (1) do0Branch jumps to the address beside it if it consumes a 0 value, otherwise just advances the parameter field pointer by 1.
- (2) doElse and doThen are synonyms for doNop (no operation). Their location is what is important, not what they do.

ASSEMBLY OF PARAMETER FIELD – IF/ELSE/THEN

Python method	Dictionary def	Contents of Parameter field	Stack
compileIf	IF (-- loc1)	0BRANCH -1	1
doHello	HELLO (--)	0BRANCH -1 HELLO	1
compileElse	ELSE (loc1 -- loc2)	0BRANCH 5 HELLO JUMP -1 doElse	4
doTulip	TULIP (--)	0BRANCH 5 HELLO JUMP -1 doElse doTulip	4
compileThen	THEN (loc2 --)	0BRANCH 5 HELLO JUMP 7 doElse doTulip doThen	empty
doSemi	;	Final tokens in parameter field – [56, 5, 5, 57, 7, 58, 6, 59]. Also pops IMMEDIATE vocabulary off of the stack	

CREOLE FORTH COMPILING WORDS - DIFFERENCES

- Most Forths define a state variable to differentiate between compiling and not compiling.
- A word that is marked immediate will execute during compilation, not compile.
- Creole Forth lacks the state variable.
- Instead, all immediate words are in the IMMEDIATE vocabulary.
- During compilation, the IMMEDIATE vocabulary is placed on top of the vocabulary stack, therefore this vocabulary is always searched first.
- At the end of compilation, the IMMEDIATE vocabulary is knocked off the vocabulary stack, which makes the words in that vocabulary inaccessible afterwards.

CREOLE FORTH COLON COMPILER

- Definitions are built in the PAD data structure
- Each word in the definition is looked up.
- The fully qualified name (name + vocabulary), token or address, and compile-time action are placed in a CompileInfo object.
- For ordinary words, the compile-time action is COMPINPF, which is a synonym for , (COMMA).
- For compiling words, this action is EXECUTE.
- Once the name-token-action triplets are all in PAD, the respective tokens and actions are passed to an interpreter. Each token is pushed onto the stack, then consumed by its compile-time action.

IF-ELSE-THEN IN PAD

WORD	FQ NAME IN PAD	TOKEN/ADDRESS IN PAD	COMP ACTION IN PAD
IF	IF.IMMEDIATE	53	EXECUTE
HELLO	HELLO.FORTH	5	COMPINPF
ELSE	ELSE.IMMEDIATE	54	EXECUTE
TULIP	TULIP.FORTH	6	COMPINPF
THEN	THEN.IMMEDIATE	55	EXECUTE
;	;.IMMEDIATE	42	EXECUTE

FINAL OBSERVATIONS

- The more recent versions of Creole Forth (Python, VB, etc) appear to blur the distinction between compilation and metacompilation.
- In a metacompilation loop, each symbol encountered is first **executed**, not compiled.
- This execution causes it to compile its target compilation address into its target.
- In Creole Forth, each token in pad is executed by its compilation action, which is embedded in its definition.
- Ordinary words have their tokens placed ‘as is’ in the parameter field.
- Compiling words are executed, which generates code into the parameter field (it could generate code elsewhere too).
- This may offer advantages in modularity over a single global compilation process with a STATE variable.

QUESTIONS/COMMENTS?