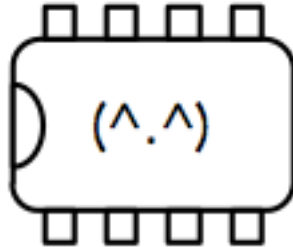


4th Saga ch1

A Taxonomy of Forths



www.ZENMCU.com

Draft 2 2023-01-25

Copyright © 2023 by Trenton Henry, All rights reserved

A Taxonomy of Forths #1

I wrote the original draft of this on 11/03/21 to try to work through a discussion on Discord and come to a better understanding of the numerous variations on Forth. I decided today to resurrect and expand on this just because the differences between s4th and x4th, and the various design options I have explored, seemed to indicate that I might have left some possibilities out.

A vast number of Forth implementations exist, and even more Forth-like language implementations. It is kind of difficult to define just what exactly is a Forth. So I read up, and listened to discussions, and thought "hmmm..." and this is what I came up with for everyone to disagree on.

Any individual language implementation may fit one or more of these categories.

An **RPNNotForth** is any RPN language that is not Forth. I.e, just using RPN does not count as a Forth.

A **Forth-like** language is a concatenative language with similarities to Forth, such as Joy, or Factor.

A **standard** Forth is one that conforms to a specific published Forth standard like and ANSI/ISO Forth.

A **non-standard**, or **Forth-inspired**, 'Forth' is one which is not compliant with any of the various Forth standards specifications.

A **hosted** Forth is something like GForth which runs on a Windows/Mac/Linux/etc workstation.

A **PlatForth** is non-hosted Forth that implements its own platform/operating system. These are more common for microcontrollers and embedded systems.

A **self-hosted** Forth is one that implements a dictionary and is capable of defining new words at runtime.

A **cross-compiled** Forth is one whose dictionary is created on a workstation and the code portions is downloaded into the Forth executable. Often these are not self-hosted and cannot define new words at runtime.

Of course, there are also a number of design considerations which may or may not lend themselves to helping define a given system's "Forthiness". Obviously there are the many threading models that are a lively topic of discussion amongst Fortherers. As is the memory model in general.

Does that help at all?
Nope? Oh well, I sort of tried.

End