John Hardy
Posted on 18 Dec 2019 • Updated on 22 Mar 2021

Structured Programming in Z80 Assembly

struct-z80 - Structured Programming in Z80 Assembly Language

```
title: Structured Programming in Z80 Assembly Language
published: true
description: Using assembly macros to write high level code
tags: Z80, macros, assembler, structured programming
```

One of the great pains of writing assembly language for old-school microprocessors such as the Z80 is the complexity of implementing algorithms due to the lack of high-level control and looping structures. All you have are conditional jumps and labels and nothing to help you enforce the structure of your code.

It's not exaggerating when someone claims that GOTOs are considered harmful ...at least to your state of mental well-being! ;-)

This lack of structure is what often ends up driving programmers in the direction of high-level languages such as C which people often think of as "low level", "assembler++" or "close to the metal". On 8-bit CPUs, however, this far from the truth and C adds a lot of overhead to your machine-cycle and memory-cell constrained code. This is why assembly language is still the tool of choice for 8-bit programming despite it also being a major source of frustration.

Macros are a huge boon to writing assembly language. Recently I developed a set of macros that were inspired by a coding pattern invented by Garth Wilson and (quite separately by) Dave Keenan which enabled me to write structured programs in assembly language.

See:

Program Structures with Macros

Adding Structured Control Flow to Any Assembler

Both authors were heavily influenced by the Forth programming language and the way that it introduced high-level structured programming concepts to low level programming years before systems languages like C and Pascal became commonplace.

The examples I'm giving here were written using the asm80 macro system but I'm sure they could be easily adapted to your own favourite assembly's macro syntax.

See:

Asm80

Asm80: Macros

Macro library contents

1. Control structures
2. Looping structures
3. Structure stacks

Installation

A repo of all the macros discussed here can be found here:

struct-z80

Include the following files in your project

For `_if, _else, _endif, _switch, _case, _endcase, _enddo`

```
.include "struct-macros.z80"
```

For `_do, _while, _until, _break, _continue, _endo`

```
.include "dloop-macros.z80"
```

Contact me: jh@lagado.com GitHub: jhlagado Twitter: @jhlagado

Top comments (0)

Code of Conduct • Report abuse

Thank you.

Thanks for visiting DEV, we've worked really hard to cultivate this great community and would love to have you join us. If you'd like to create an account, you can sign up here.

John Hardy

LOCATION
Australia

WORK
Lead Developer

JOINED
27 Jun 2018

Trending on DEV Community 🔥


Meme Monday
#discuss #watercooler #jokes


Skill section or not?
#help #discuss #webdev #portfolio


Which Programming Language Did You Choose to Start with, and Why Did You Choose It?
#discuss #beginners #codenewbie

You can customize your theme, font, and more when you are signed in.