Overview of Intel80tools repository

This repository started off as a place for me to share some of my reverse engineering attempts of the Intel ISIS 8085 development tools.

Over the years this has grown to include reverse engineering of various ISIS releases and applications and a smaller number of CP/M tools. Also a number of ports of the applications to C have been done.

In addition to the reverse engineering efforts, the repository includes a development environment to support my re-engineering. Most of the tools included are provided with source code.

Note, some of the development utilities are written in perl, if you wish to use these or the makefiles I supply use them, then you will need to make sure a version of perl is installed on your system.

The documents in this directory are written using markdown, however I have also supplied html and pdf versions for convenience .

Documentation

Overview.md

You are reading it.

CPMSrc.md

A small number of CP/M decompilations are documented in this document.

Cports.md

One of the earliest attempts at reverse engineering in 2007 was to translate the Intel PL/M V4.0 compiler into C++ directly from the disassembled code. This historic version is in the repository but now considered depreciated and is certainly not modern C++.

My current approach is to decompile into PL/M and then to translate from PL/M to C so that the C code closely follows the PL/M version. Once in C I have access to modern tools that allow me to understand the original code better and create meaningful comments and variable names.

The document provides information on the ports and also provides notes on how I go about porting from PL/M to C, which may be of interest to others.

IntelSrc.md

Documents the various reverse engineering efforts and includes notes on how I go about reverse engineering.

IntelTools.md

In the repository there is a subset of my overall collection of Intel ISIS tools and applications. Some are used as build tools using thames. Additionally they are useful as reference images to verify that the reverse engineering is correct.

The document describes how the tools are organised in the repository. It also documents the differences between the known versions of ISIS-I and ISIS-II.

makefile.md

This describes my makefile support I used to rebuild and test my reverse engineering efforts.

There are a number of macros and rules that simplify the makefiles.

The makefiles in the rest of the distribution give many examples of its use.

misc.md

Contains documentation on my the packed source file format I use and notes on my approach to decompiling the applications.

msdos.md

Documents a number of Intel's msdos tools in the repository.

thames.md

Originally developed by John Elliott, this application emulates enough of an ISIS-II environment to allow many of the ISIS tools to work on a modern OS.

I have added a number of enhancements that make the tool work much better with Windows and Unix variants, whilst also intercepting the error message from the ISIS tools to generate an error code that can be used in makefiles.

More recently support for creating dependency information has been added and for mapping temporary files to unique names. This later functionality allows make to run parallel builds

Tools.md

Over several years I have developed a number of tools, mainly in C and perl, to support my reverse engineering efforts.

I have included these tools in the repository, along with the source code for most of them.

The document provides information on the tools along with basic information on the command line syntax.

```
Updated by Mark Ogden 13-Sep-2020
```