

Intel hex parser, Project Summery

Group-H:

Haran, Himanshu Kumar (448428), Nengwenani, Mpho Cavin (448385) & Ravi Shankar Yadav (448416)

Introduction

Intel HEX is a file format that conveys binary information in ASCII text form. It is commonly used for programming microcontrollers, EPROMs, and other types of programmable logic devices. An Intel hex parser is implemented to parse and validate an input file of either of IHEX8 or IHEX16 or IHEX32 type. It read input file byte by byte and validate every field.

Program detail

Can be compiled only on Linux platform as it uses some Linux function open(), read() and close() for file handling.

Lines of code: 600 approx

Input

- **I8HEX** files use only record types 00 and 01 (16 bit addresses)
- **I16HEX** files use only record types 00 through 03 (20 bit addresses)
- **I32HEX** files use only record types 00, 01, 04, and 05 (32 bit addresses)

Output

- Validation (format, address & check sum)
- Blocks of Address
- Information for each type of records

Compilation:

\$ g++ -Wall -Wextra intelHexParser.cpp -o intelhexparser

To print elaborated information of all records compile with DEBUG option

\$g++ -Wall -Wextra intelHexParser.cpp -D DEBUG -o intelhexparser

Running: **\$/IntelHexParser <input_file>**

Test Files: Testfiles (52) are stored in folder testfiles, contains following files

Valid files - LCD1200.HEX, MULTIPLIER.HEX

Invalid files - TESTINPUT_Radmasa_%n.HEX (50 files **n=1-50**) generated using **Radamsa**.

Testing: Tested using *G++(-Wall -Wextra)*, *cppcheck*, *valgrind* and *radamsa* fuzzer,

Challenges

- Verification of valid address ranges.
- Different address calculation for IHEX8, IHEX16, IHEX32
- Address are not always incrementing linearly
- Checking of new line character