Intel ISIS and tools

The itools directory and sub-directories contain the binary images of the Intel ISIS tools. This is a subset of a much larger collection of applications that I have.

The tools serve two purposes, firstly some of them are invoked via the thames emulator to perform compilations etc. Secondly they are used as reference binaries for the decompilations to check for byte level matching, or at least equivalence.

With the exception of plm80.lib, all directories have a subdirectory for each available version. plm80.lib is the exception as there is only one known version.

At the end of this document is a summary of the differences in the known ISIS-I and ISIS-II versions.

isis & pds

These directories contain the versions of ISIS that I have, the difference between each version is summarised at the end of this document.

The oldest verison if ISIS v1.1 which is a 16k version of ISIS. This repository is the only known copy of this on the internet and was found by Jon Hales a volunteer for the Cambridge Centre for Computing History who located it in their archive.

Decompilation of the ISIS v1.1 tools has identified that ISIS.BIN appears to have been originally written in PL/M but then hand optimised. It also omits the SEEK and CONSOLE system calls.

Note, the ISIS v1.1 applications will not run under thames or any ISIS-II system as they have a start address at 1C00H which is in the OS itself on ISIS-II and the load format is not OMF85 but the same format as used by ISIS.BIN. As I have decompiled the applications, other than as80 and edit, it would be possible to modify the source to locate at the ISIS-II based address and change the makefile to build and OMF85 file.

The versions 1.1 & 2.2 were compiled with the fortran cross compiler, or rather a version somewhere between the published v2 and v4 versions of the cross compiler. The plm81 & plm82 versions shared in the tools directory are patched with the appropriate fixes from v4.

toolbox

These directories contain the Intel toolbox software, published source and libraries. The decompilation of the libraries can be found under the src directory, see IntelSrc.md for details.

Note I have added pause which was missing from the original v1.0 release

plm80.lib & fpal.lib

These are the plm80 library and Intel's floating point library. The system.lib file can be found under the associated isis, pds or utils tree.

asm80, fort80, ixref, lib, link, locate, pasc80, plm80

Intel build tools for 8080 applications.

asm86, conv86, lib86, link, loc86, oh88, plm86

Intel's ISIS II hosted build tools for 8086 development

asm48, asm51, lib51, rl51

Intel's ISIS II hosted build tools for 8048 & 8051 development

toolbox

toolbox_1.0 and toolbox_2.0 - the isis toolbox applications

basic

Interactive basic for ISIS II.

Note due to current implementations of thames, occasionally you will need to make sure cr is pressed before the interpreter recognises that characters have been typed. This restriction may be removed in future.

utils

These are the utilities from the Intel ISIS.EXE package and contain the latest versions of hexobj, objhex and system.lib

async and ftrans

Async communications application and Intel's file transfer application.

Note there is a decompilation of the ftrans application in the src tree.

genpex & genpex.txt

These are not in sub-directories. They are Intel's genpex utility and documentation. They are for information purposes only, as ngenpex has fewer idiosyncrasies and handles local variables.

ISIS-I and ISIS-II

Version 1.1 (16k ISIS)

Hardware: single SD controller at 78H, supporting 2 disks

Syscalls: 0-12 except SEEK (5) and CONSOL (8), plus the undocumented 'D' system call for direct disk I/O

Devices: F0 F1 TI TO VI VO I1 O1 TR HR R1 R2 TP HP P1 P2 LP L1 CI CO BB

ISIS.T0 does not probe available hardware

Load uses the format used by ISIS.BIN and not the OMF85 format used in ISIS-II

Applications load from address 1C00H, which is part of the OS in later ISIS versions.

Version 2.2

Hardware: 2 SD controllers one at 78h and one at 88h each supporting 2 disks

Syscalls: 0-13 plus the undocumented 'D' command that provides direct disk I/O

Devices F0 F1 F2 F3 TI TO VI VO I1 O1 TR HR R1 R2 TP HP P1 P2 LP L1 BB CI CO

ISIS.T0 does not probe available hardware

Note version 2.2. does not support the syscall PATH(14) which is used by some of the intel tools.

Version 3.4

Hardware: Disk types are 1->DD, 2->SD, 3->ISD

- Controller 1 @ 78h, drives 0-3 for DD or 0-1 for SD
- Controller 2 @ 88h, drives 2, 3 for SD or drives 4,5 when 0-3 is DD
- ISD @ 0c1h, drive 0 if system disk else drive 4

Syscalls: 0-14 plus the undocumented 'D' command that provides direct disk I/O

Devices F0 F1 F2 F3 F4 F5 TI TO VI VO I1 O1 TR HR R1 R2 TP HP P1 P2 LP L1 BB CI CO

ISIS.T0 probes disk types and copies the configuration into ISIS.BIN - only disk types are copied

ISIS.BIN hdr block is as follows

- DD (track 2, sector 6)
- SD (track 2, sector 4)

Version 4.0

Hardware: Disktypes are 1->DD, 2->SD, 3->ISD, 4->HD

- controller 1 @78h
- controller 2 @88h
- ISD @0c1h
- HD @68h

Drive numbering is as per version 3.4 unless a hard disk is present in which case the HD is allocated 4 logical drives F0-F3 and all the other drives are moved up 4 slots

Syscalls: 0-14 plus the undocumented 'D' command that provides direct disk I/O

Devices: F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 TI TO VI VO I1 O1 TR HR R1 R2 TP HP P1 P2 LP L1 BB CI CO

ISIS.T0 probes disk types and copies configuration tables into ISIS.BIN, this provides information on up to 10 drives, specifically

dkcftb	Controller logic: 1->DD, 2->SD, 3->ISD, 4->HD
dkcttb	Disk controller: 1->controller 1 (78h), 2->controller 2 (88h), 3->ISD (0c1h), 4->HD (68h)
unitmp	Bitmask to select drive
rdymsk	Bitmask to determine if the drive is ready
tracks	Last logical track on the drive (DD, SD, ISD) ->76, HD->199
sectors	Number of logical sectors per track (DD)->52, (SD, ISD)->26, HD->144

ISIS.BIN hdr block is as follows

- DD (track 2, sector 6)
- SD (track 2, sector 4)
- HD (track 2, sector 31)

Supports boot from F4, the ISD drive

Version 4.1

Hardware: as version 4.0

Syscalls: as version 4.0 plus 'E' which sets error handler

Devices: as version 4.0

Version 4.2

Hardware: as version 4.1

Syscalls: as version 4.1 plus GETAT(17) and GETD via ISIS.OV0

Devices: as version 4.1

Note ISIS.OV0 itself supports local and remote disks, but ISIS II does not have remote disk support. I assume this is for later versions of ISIS

Version 4.2w

Hardware: as version 4.2 but supports Winchester Disk (type 9) rather than the standard HD

Syscalls: as version 4.2

Devices: as version 4.2

WD is sized as 525 cylinders, 5 heads, 70 sectors per track, 128 byte sectors, 13 alternate cylinders

The WD is used as 4 logical drives, controller logic (9), drive type (9), 211 last logical track, 210 sectors per track

ISIS.BIN header block is located at track 2 sector 46

ISIS.T0 also initialises the WD and sets up the control blocks in memory starting at 0044h

For the WD the directory name is hashed to spread the names across the ISIS.DIR and help to locate directory entries quicker. Files prefixed with ISIS are always in bucket 0 to support normal location of key ISIS files.

Version 4.3

Hardware: as version 4.2

Syscalls: as version 4.2 plus the undocumented GETCLIBUF(28) which get the internal buffer used to keep CLI information across applications runs and to support command line editing. Note the real syscall name is unknown

ISIS.CLI supports command line editing

Version 4.3w

Hardware: as version 4.2w

Syscalls: as version 4.3

Devices: as version 4.3

isis.t0 is basically identical to 4.2w with the exception of the references to the internal tables in ISIS.BIN and different compiler versions and optimisation settings used.

Directory hashing is as per 4.2w

```
Updated by Mark Ogden 17-Sep-2020
```