

Lab 26, floating point

By ADMIN | Published: MARCH 1, 2014

In Lab 19 we found a problem that kernel raises exception “Undefined instruction” and stops working. It happens on running JIT arm codes produced by Dis. Now it is time to have little review. In Lab 19 I just added sources used in 9pi for handling floating point exceptions but commented real call *fpemu()* – so the call *fpfarm()* only shifting PC to bypass the place of wrong instructions.

By looking at /libinterp/comp-arm.c it looks like that Dis Jit for Arm currently is not generating arm codes for floating point logic – those code blocks have *if(o) {...}* but it just put some pseudocodes which should generate exception and be interpreted in **trap.c** – software emulation of floating point operations.

Then my initial attempt of using source codes from 9pi project is wrong as they will not process the dis pseudocodes – that why it didn’t work either. Now as another attempt I just used source codes from ipaq1110/sa1110 – it has the software emulation of floating point.

We copy **fpi.h** and **fpiarm.c** from **sa1110/**. Then add **fpi.\$O**, **fpimem.\$O**, **fpi.h** to **mkfile**. Remove from **rpi** from port section fpi and fpimem. Also we need “**-f**” flag to the linker – the **5l** loader currently has support for *vfp2* used in Raspberry Pi CPU.

And that’s all. Let just add some floating point code to rpiinit.b just to test:

```
1 x := 10.25;
2 y := 734.;
3 sys->print("\n\nfloat point div %f/%f=%f\n\n", x,y,x/y);
```

Compile, run and see:

```
1 ...
2 Starting init0()
3 Initial Dis: "/osinit.dis"

4
5
6 float point div 10.250000/734.000000=0.013965
7
8 ...
```

Success! Testing dis apps – ls, etc – all works okay.

PS: Ideally the Dis JIT should generate the vfp2 arm instructions to do the fast floating point calculations. I contacted Charles Forsyth and he proved that the support of vfp2 was implemented in 5l but not in Dis, though it is not much efforts to get Dis supporting this – let’s hope he will have this implemented soon!

This entry was posted in *Blog, Inferno OS, Raspberry Pi, Research*. Bookmark the *permalink*. *Post a comment* or leave a *trackback*: *Trackback URL*.

Categories

- [Blog](#)
- [Boost](#)
- [C++](#)
- [Cryptography](#)
- [Embedding](#)
- [Hybrids](#)
- [Inferno OS](#)
- [MacAppStore](#)
- [Misc](#)
- [Models](#)
- [Projects](#)
- [PyQt](#)
- [PySide](#)
- [Qt](#)
- [QtSpeech](#)
- [Raspberry Pi](#)
- [Research](#)
- [Ru](#)
- [TogMeg](#)
- [Trac](#)
- [TTS](#)
- [Tutorial](#)
- [Undo](#)
- [Usb](#)
- [Web](#)

