Lab 17, mmu init

By ADMIN | *Published:* NOVEMBER 15, 2013

What is fun about Inferno OS that our CPU may not have MMU support and still we are able to port Inferno to such architecture. That's because Inferno OS implements Dis virtual machine and all users processes are actually Dis byte-codes interpreters on JIT compiled. So address space is virtual by software layer and it is very good for portability of the system.

Anyway, still it is good to have the MMU initialized (may be needed later to work with framebuffer) and it looks simple enough to do.

So idea that we map 1:1 all memory that we have SDRAM banks, then we also map IO address space and also low-high interrupt vectors. Also some adjustments to **load.s** to have *mmuinit()* to called before *main()*:

```
01 void mmuinit(void)
02 {
03     PTE *l1, *l2;
04     uintptr pa, va;
05
06     l1 = (PTE*)PADDR(L1);
07     l2 = (PTE*)PADDR(L2);
08
09     /* map all of ram at KZERO */
10     va = KZERO;
11     for(pa = PHYSDRAM; pa < PHYSDRAM+DRAMSIZE; pa += MiB){
12         l1[L1X(va)] = pa|Dom0|L1AP(Krw)|Section|Cached|Buffered;
13         va += MiB;
14     }
15
16     /* identity map first MB of ram so mmu can be enabled */
17     l1[L1X(PHYSDRAM)] = PHYSDRAM|Dom0|L1AP(Krw)|Section|Cached|Buffered;
18
19     /* map i/o registers */
20     va = VIRTIO;
21     for(pa = PHYSIO; pa < PHYSIO+IOSIZE; pa += MiB){
22         l1[L1X(va)] = pa|Dom0|L1AP(Krw)|Section;
23         va += MiB;
24     }
25
26     /* double map exception vectors at top of virtual memory */
27     va = HVECTORS;
28     l1[L1X(va)] = (uintptr)l2|Dom0|Coarse;
29     l2[L2X(va)] = PHYSDRAM|L2AP(Krw)|Small;
30 }
```

There are also minor changes to **dat.h**, **mem.h**, etc

FILES:

rpi-lab17.zip

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
- Web

« *Qt and use of Cryptography – simple!*

Lab 18, we have a screen! »

Copyright LynxLine. All rights reserved.
Powered by lynxline.com, WordPress