Lab 24, network, part 1

By ADMIN | *Published:* FEBRUARY 9, 2014

This lab will be short; we will add all network modules needed for network layout and make a roadmap how to make the Raspberry Pi ethernet driver.
First have a look at other ports and inferno and find what is needed to compile with network support.
1. dev section needs:

```
+       ip  ip ipv6 ipaux iproute arp netlog ptclbsum iprouter plan9
nullmedium pktmedium netaux
+       ether netif netaux
```

2. add ip section

```
+ip
+       il
+       tcp
+       udp
+       ipifc
+       icmp
+       icmp6
```

3. mod section needs:

```
+       crypt
+       ipints
```

Then we will add etherusb.c from 9pi project (according to 9pi: it takes over the
job of copying packets between the usb pipe and the kernel ether i/o
queues, to save the extra overhead of doing it in user mode in the
usb/ether driver.) Later we may not need it, but for now take it just to have complete compilation.

Then link section:

```
link
        usbdwc
+       etherusb
+       ethermedium
+       loopbackmedium
```

In archpi.c we will add ethernet initialization:

```
#include "dat.h"
 #include "fns.h"

+#include "../port/netif.h"
+#include "etherif.h"
+
 static void
 linkproc(void)
 {
...
 void
 validaddr(void*, ulong, int) {}

+int
```


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

```
+archether(unsigned ctlrno, Ether *ether)
+{
+    switch(ctlrno) {
+    case 0:
+        ether->type = "usb";
+        ether->ctlrno = ctlrno;
+        ether->irq = -1;
+        ether->nopt = 0;
+        ether->mbps = 100;
+        return 1;
+    }
+    return -1;
+}
+
+/*
+ * stub for ../omap/devether.c
+ */
+int
+isaconfig(char *class, int ctlrno, ISAConf *isa)
+{
+    USED(ctlrno);
+    USED(isa);
+    return strcmp(class, "ether") == 0;
+}
```

That’s enough for now. We can compile everything, we will have appropriate modules, but no ethernet driver yet.
As next actions I see two ways of implementing/porting it.

1. Raspberry Pi Usb driver for Plan9 written in C to be ported to Limbo
2. Raspberry Pi Usb driver for Plan9 written in C just used as some fileserver which is initialized from Usb Limbo layout (which /dev/usb files to use to read/write, etc)

CHANGES:
Revision on code.google.com

« *Jongling models beetween threads*

Lab 25, network, part 2 »