LYNXLINE

Professional Software Development Services

Lab 7, linking, planning next

By ADMIN | *Published:* DECEMBER 17, 2012

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

As we have the kernel almost compiled now it is time to check what is missing to have it linked. By checking header files and adding stubs into main.c we will get linked with very small changes as:
Changes to **main.c:**

```
#include "u.h"
#include "../port/lib.h"
#include "dat.h"
#include "mem.h"

Conf conf;
Mach *m = (Mach*)MACHADDR;
Proc *up = 0;

#include "../port/uart.h"
PhysUart* physuart[1];

int     waserror(void) { return 0; }
int     splhi(void) { return 0; }
void    splx(int) { return; }
int     spllo(void) { return 0; }
void    splxpc(int) { return; }
int     islo(void) { return 0; }
int     setlabel(Label*) { return 0; }
void    gotolabel(Label*) { return; }
ulong   getcallerpc(void*) { return 0; }
int     segflush(void*, ulong) { return 0; }
void    idlehands(void) { return; }
void    kprocchild(Proc *p, void (*func)(void*), void *arg) { return; }
ulong   _tas(ulong*) { return 0; }
ulong   _div(ulong*) { return 0; }
ulong   _divu(ulong*) { return 0; }
ulong   _mod(ulong*) { return 0; }
ulong   _modu(ulong*) { return 0; }

void    setpanic(void) { return; }
void    dumpstack(void) { return; }

void    exit(int) { return; }
void    reboot(void) { return; }
void    halt(void) { return; }

Timer*  addclock0link(void (*)(void), int) { return 0; }
void    clockcheck(void) { return; }

void    fpinit(void) {}
void    FPsave(void*) {}
void    FPrestore(void*) {}
```

Changes to **fns.h:**

```
#define coherence()     /* nothing needed for uniprocessor */
#define procsave(p)     /* Save the mach part of the current */
                        /* process state, no need for one cpu */
```

This list of functions defines very clear what are scopes of functionality is required to implement. For example, fpinit, FPsave, FPrestore are related to have floating point operations, “clock” functions to have system clocks to be programmed and tested. Some functions for rebooting, halting, dumping stack

– say they are “util” functions. And finally block of functions (which is first in list) is to be implemented in assembler to have kernel to use R-Pi platform.

You see that we make “empty” defines for *coherence()* and *procsave()*, so probably in future for graphics codes, the *coherence()* should have something on board, but for our current state it is fine.

My plan for a next lab is to describe memory model and initial initializations for memory pools.

Files:
os/rpi/fns.h
os/rpi/main.c
os/rpi/rpi

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

« *Lab 6, Compile something*

About Boost Multi-index Containers »