Lab 10, Bss, memory pools, malloc

By ADMIN | *Published:* FEBRUARY 16, 2013

Time to have a look at initial memory initializations.
First we need to initialize BSS (more details: http://en.wikipedia.org/wiki/.bss), statically-allocated variables. We do it in next way:

```
1 memset(edata, 0, end-edata);
```

From Plan9 manual:
*“The loaded image has several symbols inserted by the loader: **etext** is the address of the end of the text segment; **bdata** is the address of the beginning of the data segment; **edata** is the address of the end of the data segment; and **end** is the address of the end of the bss segment, and of the program.”*

When it is done, we init with zeros also *Mach* obejct:

```
1 memset(m, 0, sizeof(Mach));
```

Also let’s move *pl011* stuff into **pl011.c** and add signatures into **fns.h**:

```
1 void    pl011_putc(int);
2 void    pl011_puts(char *);
3 void    pl011_serputs(char *, int);
```

Specify amount of *Mach* objects:

```
1 conf.nmach = 1;
```

Specify a function for serial console writing:

```
1 serwrite = &pl011_serputs;
```

Next is *confinit()* call to initialize *Conf* object:

```
01 extern int main_pool_pcnt;
02 extern int heap_pool_pcnt;
03 extern int image_pool_pcnt;
04
05 void
06 confinit(void)
07 {
08     ulong base;
09     conf.topofmem = 128*MB;
10
11     base = PGROUND((ulong)end);
12     conf.base0 = base;
13
14     conf.npage1 = 0;
15     conf.npage0 = (conf.topofmem - base)/BY2PG;
16     conf.npage = conf.npage0 + conf.npage1;
17     conf.ialloc = (((conf.npage*(main_pool_pcnt))/100)/2)*BY2PG;
18
19     conf.nproc = 100 + ((conf.npage*BY2PG)/MB)*5;
20     conf.nmach = 1;
21
22     print("Conf: top=%lud, npage0=%lud, ialloc=%lud, nproc=%lud\n",
```


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

```
            conf.topofmem, conf.npage0,
            conf.ialloc, conf.nproc);
}
```

Plus we need to add *ulong npage;* and *ulong topofmem;* into *Conf* structure
and definition *#define PGROUND(s) ROUND(s, BY2PG)* into **mem.h**
Three variable: int main_pool_pcnt; int heap_pool_pcnt; int image_pool_pcnt; we define in **rpi** config file:

```
code
    int kernel_pool_pcnt = 10;
    int main_pool_pcnt = 40;
    int heap_pool_pcnt = 20;
    int image_pool_pcnt = 40;
```

Finally we can initialize memory pools:

```
static void
poolsizeinit(void)
{
    ulong nb;
    nb = conf.npage*BY2PG;
    poolsize(mainmem, (nb*main_pool_pcnt)/100, 0);
    poolsize(heapmem, (nb*heap_pool_pcnt)/100, 0);
    poolsize(imagmem, (nb*image_pool_pcnt)/100, 1);
}
```

And in *main()*:

```
xinit();
poolinit();
poolsizeinit();
```

As memory pools are ready then malloc() is functional we can do much more in next labs.

FILES:
dat.h
fns.h
mem.h
main.c
pl011.c
rpi
mkfile
bss.pdf

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

« *Lab 9, coding assembler part*

Overhead cost of using Qt models »