In Entry.S, we have a call to the function i386_init

```
void
i386_init(void)
{
        extern char edata[], end[];

        // Before doing anything else, complete the ELF loading process.
        // Clear the uninitialized global data (BSS) section of our progr
        // This ensures that all static/global variables start out zero.
        memset(edata, 0, end - edata);

        // Initialize the console.
        // Can't call cprintf until after we do this!
        cons_init();

        cprintf("6828 decimal is %o octal!\n", 6828);

        // Lab 2 memory management initialization functions
        mem_init();

        // Drop into the kernel monitor.
        while (1)
                monitor(NULL);
}
```

Edata is a symbol points to the end of the .data section.
End is a symbol points to the end of .bss section

```
        .data : {
                *(.data)
        }

        PROVIDE(edata = .);

        .bss : {
                *(.bss)
        }

        PROVIDE(end = .);
```

```
npages_basemem =160   , npages_extmem = 16383
Physical memory: 64M available, base = 640K, extended = 63M
first nextfree= 0 , *first nextfree =53
 n= 4096
 npages= 16639
 end = 0xf0113970 ,  edata = 0xf0113300
 the inside of if = 0x13feb000  and in decimal = 335458304
 result= f0114000
kern_pgdir= f0114000
first nextfree= 1000 , *first nextfree =0
 n= 133112
 npages= 16639
 end = 0xf0113970 ,  edata = 0xf0113300
 the inside of if = 0x40fe000  and in decimal = 68149248
 result= 1000
```

If you take the IOPHYSMEM 0xA0000, and divide it by the page size⇒ you will get 160 !!
Btw 64M = 67108864
n_pages_extenedn* 4096 = 67104768

```
    npages = (EXTPHYSMEM / PGSIZE) + npages_extmem;
```

This calculates the entire number of pages including the whole, why? Because we used it later in a for loop and we exclude the hole ourselves

65M in hex ⇒ 4100000 , so the inside of if is missing 8192 bytes which is exactly 2 pages….

```
Sections:
Idx Name          Size      VMA       LMA       File off  Algn
  0 .text         00001fc1  f0100000  00100000  00001000  2**4
                  CONTENTS, ALLOC, LOAD, READONLY, CODE
  1 .rodata       00000c50  f0101fe0  00101fe0  00002fe0  2**5
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  2 .stab         00004645  f0102c30  00102c30  00003c30  2**2
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  3 .stabstr      00001c70  f0107275  00107275  00008275  2**0
                  CONTENTS, ALLOC, LOAD, READONLY, DATA
  4 .data         0000a300  f0109000  00109000  0000a000  2**12
                  CONTENTS, ALLOC, LOAD, DATA
  5 .bss          00000670  f0113300  00113300  00014300  2**5
                  ALLOC
```

The .text is stored in 1M and linked 1M+kernbase
Kernbase is 3.75 Giga

```
 the middle of if nextfree= f0114000
```

```
end = 0xf0113970 ,  edata = 0xf0113300

the middle of if nextfree= f0114000 , *second nextfree =0
```

The difference between end and the first assigned nextfree because nextfree must rounded to multiple of 4096

```
kern_pgdir= f0114000
```

```
kern_pgdir= f0114000
```

The first n requested is 4096

```
enum {
    // For page_alloc, zero the returned physical page.
    ALLOC_ZERO = 1<<0,
};
```

```
npages_basemem =160  , npages_extmem = 16383
Physical memory: 64M available, base = 640K, extended = 63M
first nextfree= 0 , *first nextfree =53
 n= 4096
 npages= 16639
 end = 0xf0113970 ,  edata = 0xf0113300


 the second nextfree= f0114000 , *second nextfree =0
 the inside of if = 0x13feb000  and in decimal = 335458304
 result= f0114000
kern_pgdir= f0114000
first nextfree= 1000 , *first nextfree =0
 n= 133112
 npages= 16639
 end = 0xf0113970 ,  edata = 0xf0113300


 the second nextfree= 1000 , *second nextfree =0
 the inside of if = 0x40fe000  and in decimal = 68149248
 result= 1000
```