

MEMOS-2: Example PM program (stub.S)

```
.text

.globl _start
_start:
    jmp     real_start
    # Multiboot header - Must be in 1st page of memory for GRUB
    .align 4
    .long  0x1BADB002 # Multiboot magic number
    .long  0x00000003 # Align modules to 4KB, req. mem size
    # See 'info multiboot' for further info
    .long  0xE4524FFB # Checksum

real_start:
    #TODO: Setup a proper stack for C
    #TODO: Prepare the boot information to pass to kmain
    call   kmain
    hlt
```

MEMOS-2: Example PM program (kentry.c)

```
#include "memos.h"
static unsigned short *videoram = (unsigned short *)0xB8000; //Base address of the VGA frame
buffer
static int attrib = 0x0F; //black background, white foreground
static int csr_x = 0, csr_y = 0;
#define COLS 80

void putc(unsigned char c){
    if(c == 0x09){ // Tab (move to next multiple of 8)
        csr_x = (csr_x + 8) & ~(8 - 1);
    }else if(c == '\r'){ // CR
        csr_x = 0;
    }else if(c == '\n'){ // LF (unix-like)
        csr_x = 0; csr_y++;
    }else if(c >= ' '){ // Printable characters
        *(videoram + (csr_y * COLS + csr_x)) = c | (attrib << 8); // Put the character w/
attributes
        csr_x++;
    }
    if(csr_x >= COLS){ csr_x = 0; csr_y++;} // wrap around!
}

void puts(char *text){
    for (int i = 0; i < strlen((const char*)text); i++) // You know how to implement strlen ;)
        putc(text[i]);
}

void kmain(boot_info_t* binfo){
    puts("MemOS: Welcome *** Total Free Memory: ");
}
```

MEMOS-2 Example PM program (link.ld)

```
OUTPUT_FORMAT("elf32-i386")
OUTPUT_ARCH(i386)
ENTRY(_start)

SECTIONS {
    . = 0x100000;
    .ksection : {
        *(.*);
        . = ALIGN(0x1000);
    }
}
```

MEMOS-2: Example PM program (Makefile)

```
as --32 stub.S -o stub.o
```

```
gcc -m32 -fno-stack-protector -fno-builtin -  
nostdinc -c kentry.c -o kentry.o
```

```
ld -m elf_i386 -T link.ld stub.o kentry.o -o  
memos2.elf
```

MEMOS-2: Configuring GRUB

- Install grub on your disk image
- Copy your ELF binary image to the disk image
- Configure grub (through menu.lst) to load your ELF binary as a kernel

```
title MEMOS-2  
root (hd0,0)  
kernel /path/to/memos2.elf
```