HSC
Henrietta Semiconductor Corporation

Milestone 1: SeeGOL

(Shoyler's Extremely Experimental Graphical Open Library)

Schuyler Martin <sam8050@rit.edu> <http://shoyler.com>
Computer Science, BS/MS
Rochester Institute of Technology
Computer Science MS Project, CSCI-788-02

The Premise

(Recap)

Henrietta Semiconductor Corporation

In an alternative universe...

File Edit Windows View Disk Special Colors Extras
AndysHD
ProDOS 6 items 2.3 MB used 7.6 MB
ProDOS
System
BASIC
Icons
Documents
Applications
AndysHD
Trash
IBM

HSC
Henrietta Semiconductor Corporation

A monument to compromise

16-bit 8086

32-bit i386+
(with 16-bit Real Mode)

Henrietta Semiconductor Corporation

Project So Far

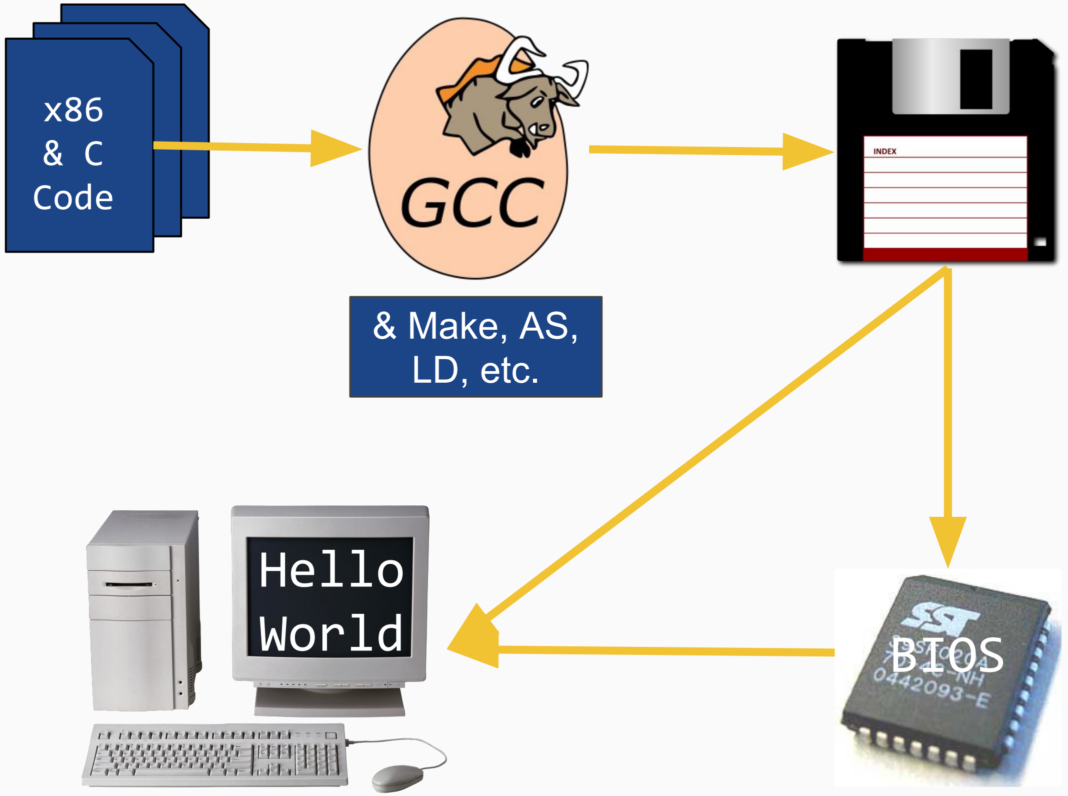
Henrietta Semiconductor Corporation

Stage 0:
The Bootloader

- It works!
- Dumps “BL” to the console for debugging purposes
- Initializes text mode and switches to the kernel loop

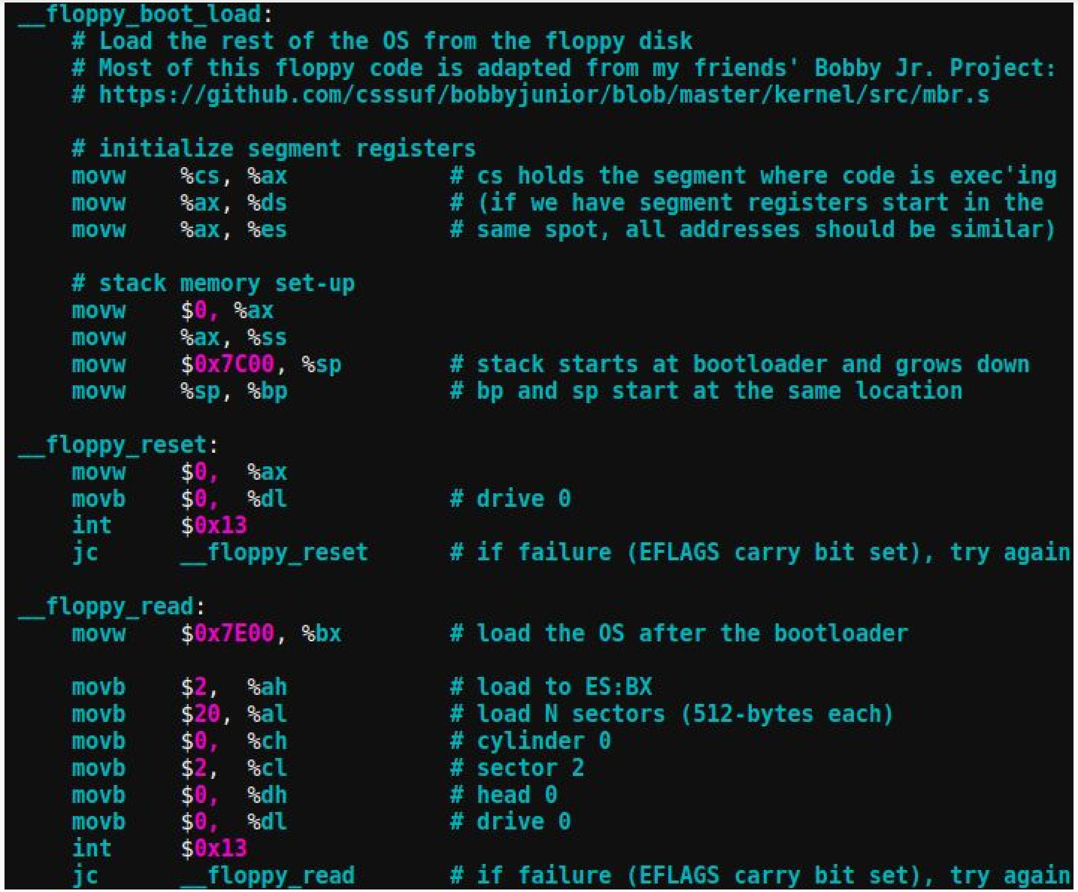
Henrietta Semiconductor Corporation

Development Toolchain and Booting



Henrietta Semiconductor Corporation

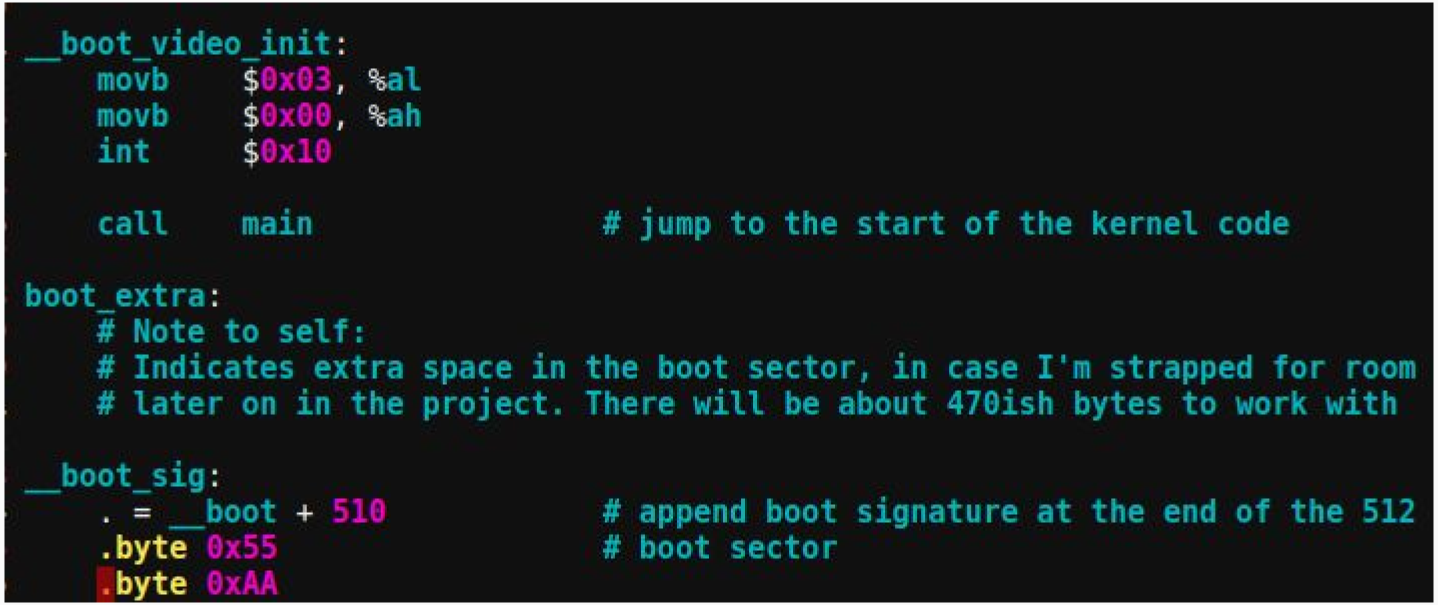
Load OS from the Floppy to Memory



```
__floppy_boot_load:
    # Load the rest of the OS from the floppy disk
    # Most of this floppy code is adapted from my friends' Bobby Jr. Project:
    # https://github.com/csssuf/bobbyjunior/blob/master/kernel/src/mbr.s

    # initialize segment registers
    movw    %cs, %ax                # cs holds the segment where code is exec'ing
    movw    %ax, %ds                # (if we have segment registers start in the
    movw    %ax, %es                # same spot, all addresses should be similar)

    # stack memory set-up
    movw    $0, %ax
    movw    %ax, %ss
    movw    $0x7C00, %sp            # stack starts at bootloader and grows down
    movw    %sp, %bp                # bp and sp start at the same location

__floppy_reset:
    movw    $0,  %ax
    movb    $0,  %dl                # drive 0
    int     $0x13
    jc      __floppy_reset          # if failure (EFLAGS carry bit set), try again

__floppy_read:
    movw    $0x7E00, %bx            # load the OS after the bootloader

    movb    $2,  %ah                # load to ES:BX
    movb    $20, %al                # load N sectors (512-bytes each)
    movb    $0,  %ch                # cylinder 0
    movb    $2,  %cl                # sector 2
    movb    $0,  %dh                # head 0
    movb    $0,  %dl                # drive 0
    int     $0x13
    jc      __floppy_read           # if failure (EFLAGS carry bit set), try again
```

Henrietta Semiconductor Corporation

Kernel Switch Over and Boot Signature



```
__boot_video_init:
    movb    $0x03, %al
    movb    $0x00, %ah
    int     $0x10

    call    main                # jump to the start of the kernel code

boot_extra:
    # Note to self:
    # Indicates extra space in the boot sector, in case I'm strapped for room
    # later on in the project. There will be about 470ish bytes to work with

__boot_sig:
    . = __boot + 510            # append boot signature at the end of the 512
    .byte 0x55                  # boot sector
    .byte 0xAA
```

Henrietta Semiconductor Corporation

Stage 1: Debugging Tools

- They work!
  - Some minor graphical bugs exist with scrolling
- Includes a C stdio-like printf() output function
  - Limited to two arguments of any type
- Includes a Python-like input() prompt

HSC
Henrietta Semiconductor Corporation

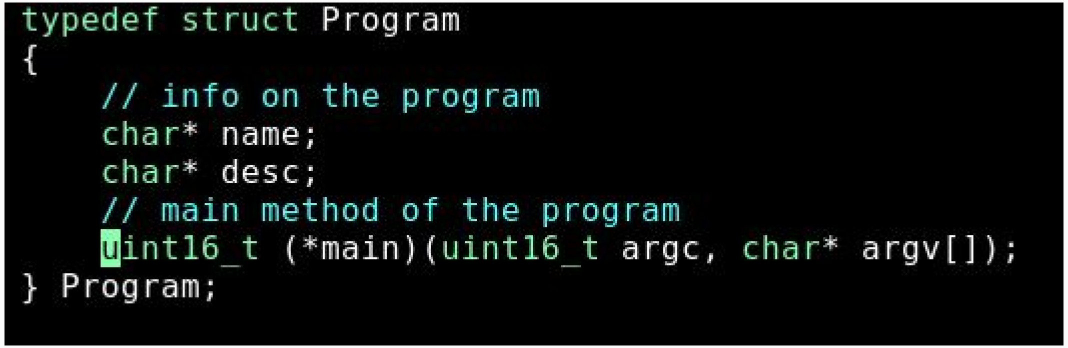
Stage 1.5:
SeeSH (Shoyler's Extremely Experimental SHell)

- Basic Shell, text interface

- Holds references to other "installed" user programs
  - They are just baked into the OS
  - Each program is defined in a function

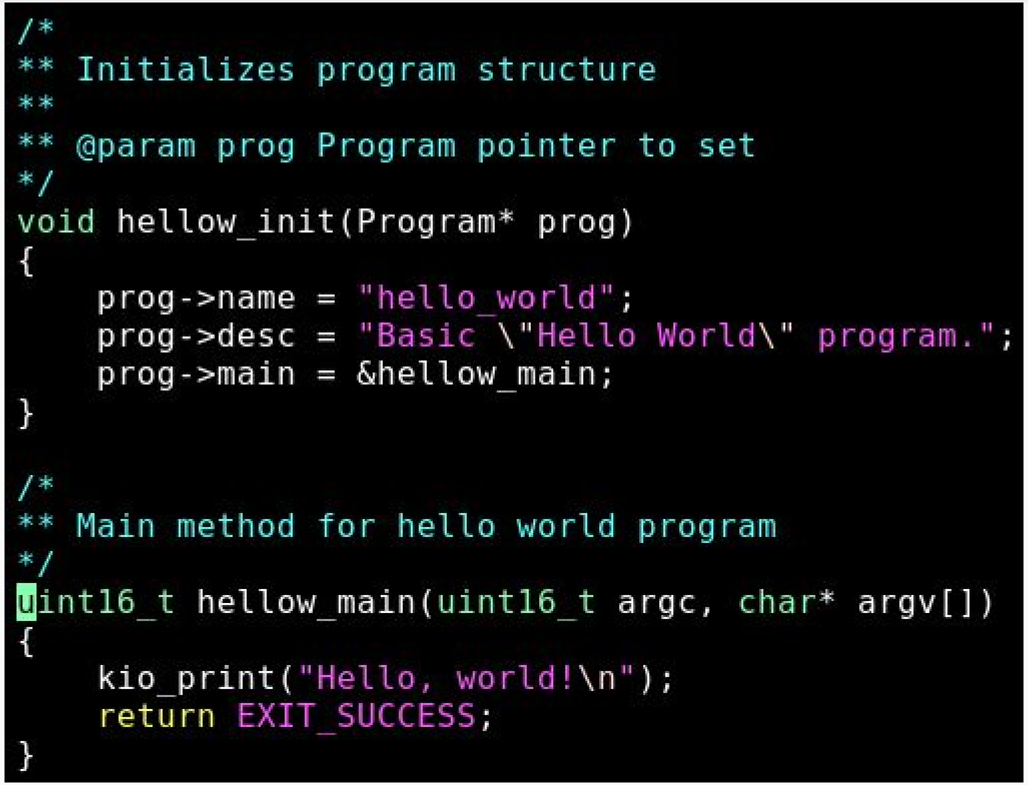
Henrietta Semiconductor Corporation

Hello World Example: Program Definition


```
typedef struct Program
{
    // info on the program
    char* name;
    char* desc;
    // main method of the program
    uint16_t (*main)(uint16_t argc, char* argv[]);
} Program;
```

Henrietta Semiconductor Corporation

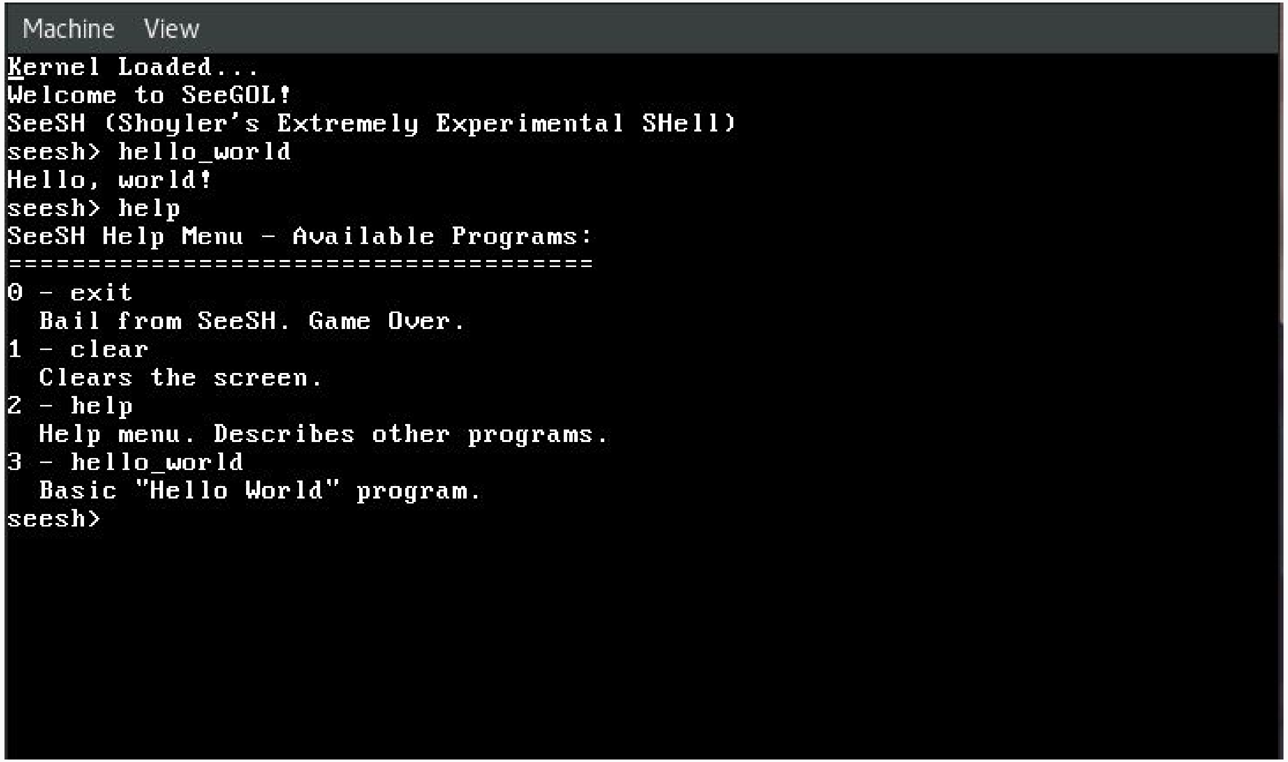
Hello World Example: Actual Program



```c
/*
** Initializes program structure
**
** @param prog Program pointer to set
*/
void hellow_init(Program* prog)
{
    prog->name = "hello_world";
    prog->desc = "Basic \"Hello World\" program.";
    prog->main = &hellow_main;
}

/*
** Main method for hello world program
*/
uint16_t hellow_main(uint16_t argc, char* argv[])
{
    kio_print("Hello, world!\n");
    return EXIT_SUCCESS;
}
```

Henrietta Semiconductor Corporation

SeeSH Demo



```
Kernel Loaded...
Welcome to SeeGOL!
SeeSH (Shoyler's Extremely Experimental SHell)
seesh> hello_world
Hello, world!
seesh> help
SeeSH Help Menu - Available Programs:
=====================================
0 - exit
  Bail from SeeSH. Game Over.
1 - clear
  Clears the screen.
2 - help
  Help menu. Describes other programs.
3 - hello_world
  Basic "Hello World" program.
seesh>
```

Henrietta Semiconductor Corporation

Sources

[1] Image content comes from freely available online resources

[2] Diagrams and Code Snippets by Schuyler Martin

[3] HSC Logo created by Kailey Martin

[4] List of resources that were deemed helpful while making this project:
https://github.com/schuylermartin45/seegol/blob/master/docs/links.txt

HSC
Henrietta Semiconductor Corporation

Special Thanks

[1] Prof. Warren Carithers - Advisor
Warren, taught me almost everything I know about Systems Programming and Computer Graphics. Without him, none of this would be possible.

[2] Prof. Sean Strout - Mentor
Sean is a close friend of mine and initially sparked a lot of my interest in becoming a C wizard.

[3] Prof. Thomas Kinsman - Mentor
Thomas has taught me how to think creatively with visual problems

HSC
Henrietta Semiconductor Corporation

Questions?

Project available at https://github.com/schuylermartin45/seegol

HSC
Henrietta Semiconductor Corporation