X86 Operating System From Scratch

How Hard Can It Be?

Bill Gates jumping an office chair

D. Barry[1]

[1]Computer Science Society
University of Hertforshire

January, 2017

Table of Contents

What Is An OS?

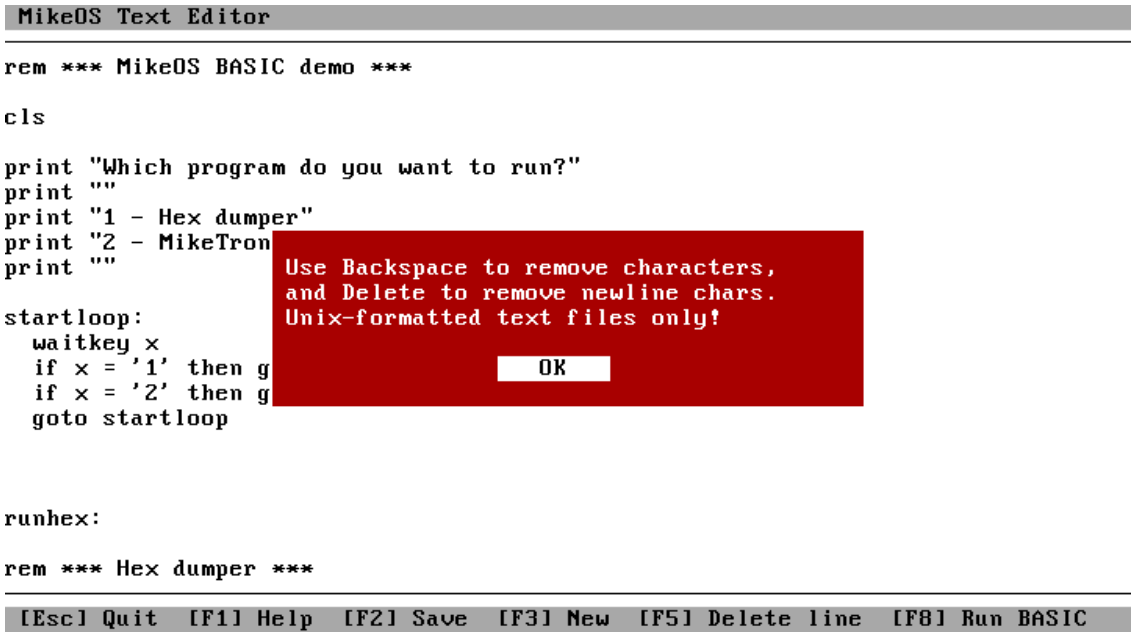
Let us define it by what we expect from it:

- Boots the computer and prepares it for a program to run.
- Provides a level of abstraction from the low level hardware (e.g. memory management, disk operations, etc).
- Allows programs to use predetermined functions, allowing for easier writing of programs.

[1]

Why Build An OS?

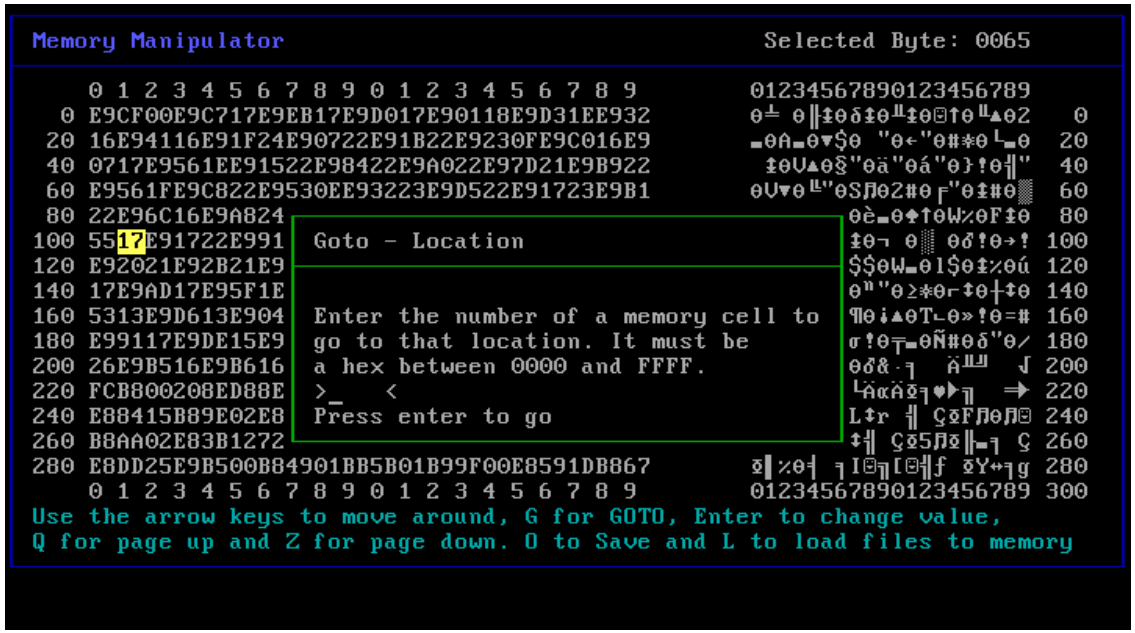
- General curiosity
- Advancing knowledge
- Fun project
- I always wanted my own [My Name] OS
- I want to be [Bill Gates/Steve Jobs/Linus Torvalds]



[2]

What OS Will We Build?

- 16 bit
- 16kB RAM
- Custom File System
- Command Line
- Monolithic



[2]

Resources

You will need:

- Some virtualiation software (`https://www.virtualbox.org/` or `http://www.vmware.com/`)
- Image Creator (can be downloaded)
- NASM (can be downloaded)

You can download the core resources from `http://coffeespace.org.uk/downloads/os-from-scratch.zip`.

[2]

Image Creator

- Allows the custom creation of disk images
- Command line utility, use: `./ic.jar -h` for information
- Custom software written in Java

```
ic [OPT] <FILE 1> .. <FILE N>

  Legal
    USE THIS PROGRAM AT YOUR OWN RISK.

  Description
    This program has been design to generate an bootable image
    for a specific test Operating System with a custom file
    table. This program is to be used for the Operating System
    described in the book by Daniel Barry.

  OPTions
    -a
    --about         Information about program
    -b
    --boot          Set the bootloader
                    Default: boot.bin
    -h
    --help          Displays this help
    -o
    --output        Set the output
                    Default: os.img
    -s
    --size          Set the media size (bytes)
```

NASM

- Compiler for assmebly code to machine code
- Well tested, well documented
- Other compilers will unlikely work

```
497 string_compare:
498   push si
499   mov si, ax
500   push ax
501   mov dl, ch ; Store for later
502
503 .repeat:
504   cmp ch, UBYTE_MIN ; Make sure we have not exceeded the search
505   je .exit ; If all searched, exit pass
506
507   lodsb ; Get next byte from SI
508   mov ah, byte [bx] ; Retrieve byte from BX
509   inc bx ; Manually increment BX
510
511   dec ch ; Decrement counter
```

[3]

Getting Started

- A good starting point is on page 5 of the book in the download, under the heading “Build Environment”.
- Advanced users can skip to page 8, under the heading “Bootloader”.

References

[1] PinStake, "Windows Vs Mac Vs Linux", *PinStake.com*, `http://pinstake.com/windows-vs-mac-vs-linux-windows-mac-linux`, 2017.

[2] Saunders, M. "X86 Operating System", *SourceForget.net*, `http://mikeos.sourceforge.net/`, 2017.

[3] Anvin, H., Kukunas, K., Gorcunov, C., Kotler, F. "NASM", *nasm.us*, `http://www.nasm.us/`, 2017.