Build own operating system

First stage :

1. Build bootloader

\* what is the bootloader?

- A boot loader, also called a boot manager, is a small program that places the operating system (OS) of a computer into memory.

- Bootloaders usually contain several ways to boot the OS kernel and also contain commands for debugging and/or modifying the kernel environment.

\* Resources

( Writing a Bootloader )

[http://3zanders.co.uk/2017/10/13/writing-a-bootloader/](http://3zanders.co.uk/2017/10/13/writing-a-bootloader/?fbclid=IwAR2YGobwQOxszXHpCg1NvfZny9k7AjqXdXYpli00_t7paSMk3eaZAoRGetg)

( Write your own x86 bootloader )

[https://hackaday.com/2017/10/23/write-your-own-x86-bootloader/](https://hackaday.com/2017/10/23/write-your-own-x86-bootloader/?fbclid=IwAR1OwReshWTF2Ox_L71nfBVlP3L9uQu-DDcOV7t6nctJ9E8DESQBLNuNrR4)

# ( Bootloader Programming Tutorial: How To Develop Your Own Boot Loader )

[https://www.apriorit.com/dev-blog/66-develop-boot-loader](https://www.apriorit.com/dev-blog/66-develop-boot-loader?fbclid=IwAR2niRmw3EVF1uar8KbUuMcWg5NqQDdbKF4p_n6pWzk9eHE9t3FkzL9FgwU)

( X86 Assembly/Bootloaders )

[https://en.wikibooks.org/wiki/X86\_Assembly/Bootloaders](https://en.wikibooks.org/wiki/X86_Assembly/Bootloaders?fbclid=IwAR0Vuif3ACWreQC6UaR49tLN81K8ObXgMpApDUPvHC9FuznGkqU4-kzL3eY)

# ( Hello World bootloader in assembly )

[https://blog.sugoi.be/asm-hello-world-bootloader.html](https://blog.sugoi.be/asm-hello-world-bootloader.html?fbclid=IwAR0uMG1Rvaxn16wEjrtSLNhODNmfK0lg94_gXVNMYKdLIro2xWL0NmZ4eUU)

# ( How to write a simple operating system )

# [http://mikeos.sourceforge.net/write-your-own-os.html](http://mikeos.sourceforge.net/write-your-own-os.html?fbclid=IwAR0OtfG8BxRQ1Hs8__RFRJgXBzM493eAsH0aIxTSYBTGHVvp5y0iXsUuV8g)

# Writing an x86 "Hello world" bootloader with assembly

# [https://50linesofco.de/post/2018-02-28-writing-an-x86-hello-world-bootloader-with-assembly](https://50linesofco.de/post/2018-02-28-writing-an-x86-hello-world-bootloader-with-assembly?fbclid=IwAR3OJcqQA0UeZ9Wz15jYfG0Nkvqq5H4-ohIva_exC5GD4vRP4PH_w_xG55Y)

Writing a Tiny x86 Bootloader

[http://joebergeron.io/posts/post\_two.html](http://joebergeron.io/posts/post_two.html?fbclid=IwAR0uMG1Rvaxn16wEjrtSLNhODNmfK0lg94_gXVNMYKdLIro2xWL0NmZ4eUU)

# Creating a Bare Bones Bootloader

[https://www.reinterpretcast.com/creating-a-bare-bones-bootloader](https://www.reinterpretcast.com/creating-a-bare-bones-bootloader?fbclid=IwAR0blMY6LwYWGkjuWhlL3WXX016TkBm96Wqj0zWr0qymQRNVDGNKLC97cSg)

# Bootloader 1

[https://youtu.be/avfQyYnaJIY](https://youtu.be/avfQyYnaJIY?fbclid=IwAR3WnHBbVMUhUkxUGgCnKZ0jBOwTyRHisDe-tn3FocCruvOy1DNwOKx5XiQ)

# Apache Mynewt Build Bootloader for STM32 Blue Pill

[https://youtu.be/LWJIAOFQBe0](https://youtu.be/LWJIAOFQBe0?fbclid=IwAR2iCbTl0fSovXOWNwjeUfv3JIlP7vzD91IJq4NGU27MnhQONPkNm2h0RA8)

Operating systems development for Dummies

<https://medium.com/@lduck11007/operating-systems-development-for-dummies-3d4d786e8ac>

# How to write a simple operating system

[http://mikeos.sourceforge.net/write-your-own-os.html](http://mikeos.sourceforge.net/write-your-own-os.html?fbclid=IwAR0OtfG8BxRQ1Hs8__RFRJgXBzM493eAsH0aIxTSYBTGHVvp5y0iXsUuV8g)

# How to make real 16-bit OS using Dash (command operating system)

<https://www.youtube.com/watch?v=lmH2_nc_dVY&feature=youtu.be&fbclid=IwAR0YpuRrK8Ekpb9LtT1RcDuklTlpyBWIgQbetH8rzGCLpNpYhxckRx68YsI>

# Boot process in Linux

<https://www.youtube.com/watch?v=RgLMBXg5b9I&feature=youtu.be>

# How to Customize Grub Bootloader,

# <https://www.youtube.com/watch?v=ijPb8f7oVXU&feature=youtu.be>

# Clover EFI Bootloader Manual Installation via Ubuntu (UEFI)

# <https://www.youtube.com/watch?v=YPWWinxwOcY&feature=youtu.be>

# How to write a program that boots without an operating system

<https://www.youtube.com/watch?v=I9NDhDF49FA&feature=youtu.be>

# COSMOS Tutorial C# Part 1

<https://www.youtube.com/watch?v=oKW3BrclAUY&feature=youtu.be>

## Visual Studio Code in Action

<https://code.visualstudio.com/docs/?dv=win>

Source approved for implementation:

# Create Your Own Operating System (OS)

# <https://www.youtube.com/watch?v=l2wZf45ZcAg&t=1772s>

Tools :

* Linux mint
* **Open source ,** **Free,** **Runs on any hardware,** **High security,** **Ease of maintenance**
* Nasm ( **National Academy of Sports Medicine )**

Using for compile assembly code

To install ( apt-get install nasm)

* Qemu ( **Quick EMUlator)**

QEMU is a generic and open source machine emulator and virtualizer .

And it emulates the machine's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems.

To install (apt-get install qemu)

Lama almassry