Project Presentation: Task 1 - Bootable Linux Image via QEMU - [GitHub repository](https://github.com/mpredut/initramfs).

**Overview**

Task 1 aims to create a bootable Linux filesystem image using QEMU that prints "hello world" upon successful startup. This is achieved through a series of shell scripts that automate the process ensuring minimal user interaction and focusing on modularity and reusability.

**Structure of the Project**

The project is organized into several key scripts and components:

1. **build\_initramfs.sh**
   * Builds the initial ramdisk (initramfs) required for booting (builds manually).
2. **build\_kernel.sh**
   * Downloads and compiles the Linux kernel or uses a pre-built kernel if available.
3. **extract\_initramfs.sh**
   * Extracts the initramfs to prepare it for modifications or usage. (not used)
4. **init**
   * Initialization script for the initramfs, responsible for setting up the environment.
5. **main.sh**
   * The main script coordinates the entire process, including building the initramfs, compiling the kernel, and launching QEMU.
6. **runqemu.sh**
   * Script that launches QEMU with the created filesystem image and displays the "hello world" message.

**Methodology**

1. **Kernel Modification**
   * The "hello world" message is embedded directly into the kernel code, though this method is susceptible to changes if the kernel is updated.
2. **Shell Script Integration**
   * The message is also added via shell script, which provides a more stable and robust solution against kernel updates.

**Usage**

git clone https://github.com/mpredut/initramfs.git

cd initramfs && ./main.sh

This will execute all the necessary steps to create and run the bootable image in QEMU.

**Assumptions and Limitations**

* **Assumptions:**
  + The project assumes the use of Ubuntu 20.04 LTS
  + Superuser privileges may be required for certain operations and additional dependency installation if missing.
* **Limitations:**
  + Direct modifications to the kernel code may not be reliable after kernel source code updates.
  + Using “Init” script design to add the message ensures no dependency and better stability and robustness.

**Minimal Environment Creation**

To maintain a minimal and simple environment for testing, I manually constructed the initramfs. This approach was preferred to have detailed control and to ensure minimal overhead for the intended purpose

**Modularity and Reusability**

* The project is designed to be highly modular, allowing each script to handle a specific task. This structure enhances reusability and simplifies maintenance.

For more detailed information and to access the source code, visit the [GitHub repository](https://github.com/mpredut/initramfs).