Customize Bootsplash Image

Sogand Salehi and Mahdi Shafiee

January 25, 2024

1 Project Description

The goal of this project is to modify the boot logo of the UEFI firmware using the EDK2 development environment. The project focuses on creating a UEFI application that allows users to interactively change the boot logo by renaming and replacing image files.

2 Project Steps

2.1 Environment Setup

To begin the project, the EDK2 development environment was set up, including the installation of necessary tools and libraries. The project utilizes UEFI development tools and the EDK2 framework for building UEFI applications.

2.2 Boot Logo Modification

The primary task involves creating a UEFI application that interacts with the file system to rename and replace the boot logo file. The application provides a simple command-line interface for users to execute commands to manipulate the boot logo.

2.3 User Interaction

The application interacts with users through the console input and output. Users can input commands to rename the boot logo file or exit the application. The application provides feedback on the success or failure of each operation.

3 Challenges Faced

3.1 EDK2 Instability

One significant challenge encountered during the project was the instability of the EDK2 framework and its various versions. Compatibility issues and

changes in API behavior posed challenges in maintaining a stable and functional codebase.

3.2 C Language Peculiarities

The code for UEFI applications uses the C language, but its usage within the UEFI framework may appear different from traditional C programming. Understanding and adapting to the specific conventions and limitations of UEFI C programming were challenges faced during the development process.

3.3 Virtual Machine Graphics Issues

Due to limitations in QEMU's graphics support, the project required dual-booting into a physical environment to test the UEFI application effectively. This was necessary because QEMU's graphics did not function correctly in the virtual machine due to graphics-related issues.

4 Code Explanation

4.1 RenameFile Function

```
#include <Uefi.h>
#include <Library/UefiLib.h>
// ... (other includes)

EFI_STATUS RenameFile(EFI_FILE_PROTOCOL *Current_Dir, CHAR16*
    OldName, CHAR16* NewName) {
    // Function implementation...
}
```

Listing 1: The RenameFile function handles the renaming of a file in the specified directory.

4.2 GetOldFileName Function

```
#include <Uefi.h>
#include <Library/UefiLib.h>
// ... (other includes)

EFI_STATUS GetOldFileName(CHAR16* Buffer, CHAR16** OldFileName) {
    // Function implementation...
}
```

Listing 2: The GetOldFileName function allocates memory for the old file name and copies the provided buffer into this allocated space.

4.3 PrintDirectoryInfo Function

```
#include <Uefi.h>
#include <Library/UefiLib.h>
// ... (other includes)

void PrintDirectoryInfo(EFI_FILE_PROTOCOL *Directory) {
    // Function implementation...
}
```

Listing 3: The PrintDirectoryInfo function prints information about files in the specified directory, focusing on files with the ".bmp" extension.

4.4 GetInput Function

```
#include <Uefi.h>
#include <Library/UefiLib.h>
// ... (other includes)

EFI_STATUS GetInput(CHAR16* Buffer, UINTN BufferSize) {
    // Function implementation...
}
```

Listing 4: The GetInput function reads user input from the console.

4.5 UefiMain Function

```
#include <Uefi.h>
#include <Library/UefiLib.h>
// ... (other includes)

EFI_STATUS
EFIAPI
UefiMain (
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
  )
{
    // Function implementation...
}
```

Listing 5: The UefiMain function serves as the entry point for the UEFI application.

5 Results Interpretation

The UEFI application successfully allows users to interactively change the boot logo by renaming and replacing image files. Users can input commands to manipulate the boot logo, and the application provides feedback on the success or failure of each operation.

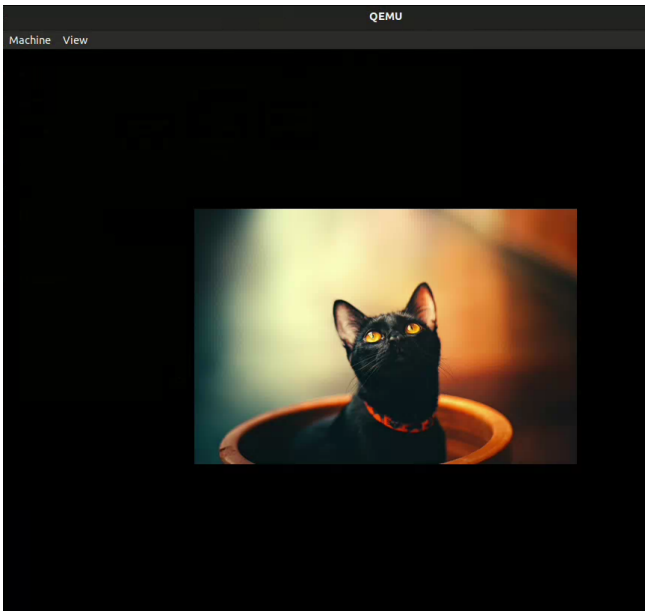

Figure 1: Visualization of the Boot Logo Modification

6 Conclusion

This documentation provides a comprehensive overview of the UEFI Boot Logo Modification project, detailing the project description, steps taken, challenges faced, code explanations, and interpretation of results. The project aims to enhance the user experience by allowing customization of the boot logo in UEFI firmware.