Operating Systems Project Proposal

Group Members: George Engel, Derek Jacobson, and William Kingsley

Project Proposal Overview:

In order to further our knowledge of operating systems, our group decided to focus on the implementation of emulating hardware/software on modern operating systems. In other words, for our project we want to build a simple emulator as a proof of concept to display how emulators function, as it is our belief that there is much to be learned by imitating a hardware/software program/platform on another program or platform that it was not initially designed for.

To specify, we want to build a Chip-8 emulation virtual machine as our proof of concept, such that game binaries built for it such as classic Pong, Tetris, and Pacman will be able to run through our own emulator on a modern operating system such as Windows 10 or Linux.

As for why we chose Chip-8 as opposed to say, emulating a NES game console, it is due to the difference in complexity between the two projects. Given the time constraint of just a little over a month, we thought it wise to choose the system we are confident we can finish building an emulator for on time, and then consider some stretch goals for that emulator that could be useful in the demo.

Specific language we want to use?

We have decided that we will be using either Rust, Go, or C# to for this project.

Stretch goal: Implementation in C# for Unity as a game object.

We will be adding support for a bootloader for the games in our emulator

- The emulator bootloader will be used for the purpose of loading and launching playable Chip-8 games on the emulator.
- What we want to do: Emulate a Chip-8 virtual environment seeing as Chip-8 is a virtual machine that is used for the purpose of being able to play classic video games written in the Chip-8 programming language.

Deliverables:

- A functioning Chip-8 emulator

Priority	Name	Description
1	Basic Functioning Emulator	The basic emulator for executing and playing games written in Chip-8.
2	Emulator Bootloader	The bootloader for launching Chip-8 games with with emulator.

Stretch Goals:

Stretch goal title	Stretch Goal Description	Estimated Time Required
Save/Load States	Add save states for quick loading/saving the current state of the game being emulated.	Unknown
'Unify' the emulator	Work the CHIP-8 emulator into a Unity game object, such that it can select and load games, and play them with configurable input. This would allow us to, for example, have a working retro console emulated in VR.	12 hours assuming the original Emulator is written in C#

Division of Labor:

George Engel: Graphics, Bootloader, Unity Game Object Emulator

Derek Jacobson: Emulator, Bootloader, Game Object Emulator

William Kingsley: Emulator, Bootloader, Graphics

Project Schedule:

DATE	DESCRIPTION
11/9/2019	Research Necessary components for the basic version of the emulator
11/16/2019	Basic version of Emulator running (Not necessarily functioning as intended)
11/23/2019	Necessary refactoring and graphical additions
11/30/2019	Demo program complete and functioning as intended
11/30/2019	Demo presentation preparations completed
12/6/2019	Present the demo

Ideation:

Focus: Learning about the topic of Emulation through independent research.

Emulation: Create specific hardware in software.

We will mostly be highlighting the process of how we make running a program intended for one set of hardware run on another set of hardware effectively. This includes common optimizations and problems as well.

The end goal being to make a small emulator of our own, or contribute to an existing open source emulation project, such as CITRA, YUZU, or VBA.

If we end up taking the small emulator route, we will likely try and emulate a chip-8 emulator, as it is common practice for beginners.

The proof of concept will demonstrate working Chip-8 software running on a Linux or Windows machine via the emulator we hope to build.

What better way to gather an understanding of hardware and its respective operating system than to emulate it as software?

-Chip-8 virtual machine

Major parts to emulate:
-CPU
-Memory
-Graphics
-

It might be cool to try and make a Chip-8 interpreter in unity, that way it can effectively act as a game object with easy to configure inputs!

That would mean any games that run on Chip-8 could then be thrown into say VR!

Structure of project:

-A bit about Chip-8

- Go into what it is
- How it works

Inspiration for the project worth mentioning:

- Given the fact that older console game systems and cartridges have a finite life span, the only real way to preserve these systems/games is through emulation.