

## Table of Contents

1. Introduction . . . . .	1
2. History of RetroBSD . . . . .	1
3. Hardware . . . . .	1
3.1. PIC32 Development Board . . . . .	1
3.2. STM32 Development Board . . . . .	1
4. Host Development Environment . . . . .	1
4.1. Development Tools on OpenBSD . . . . .	1
5. Kernel Operation Overview . . . . .	1
6. System Startup . . . . .	1
6.1. Linker Script and Bootstrapping . . . . .	1
6.2. Kernel Initialization . . . . .	1
6.2.1. Assembly-language Startup . . . . .	2
6.3. Kernel Configuration . . . . .	2
6.4. Getting to main() . . . . .	2
6.5. Getting to /sbin/init . . . . .	2
7. Userland . . . . .	2
8. Future Work . . . . .	2
9. Conclusion . . . . .	2

# **Porting the Unix Kernel**

*Christopher K. Hettrick*

University of Victoria  
Department of Computer Science  
CSC490  
Supervised by Bill Bird

## *ABSTRACT*

This report describes the process of porting a variant of the Unix kernel from the MIPS architecture to the Arm architecture. A heavily modified 2.11BSD version of the Unix kernel called RetroBSD is used as the basis for this development. The goal of this project is to run this ported kernel on both a simulator and on a physical embedded development board. An additional portion of this work is devoted to adapting the large-scale codebase of RetroBSD to more modern and sustainable development standards that will facilitate future ports to other platforms and architectures.

28 November 2020

# **Porting the Unix Kernel**

*Christopher K. Hettrick*

University of Victoria  
Department of Computer Science  
CSC490  
Supervised by Bill Bird

## **1. Introduction**

Porting the MIPS32® M4K® architecture to the Arm® Cortex®-M4 architecture.

## **2. History of RetroBSD**

## **3. Hardware**

### **3.1. PIC32 Development Board**

### **3.2. STM32 Development Board**

## **4. Host Development Environment**

### **4.1. Development Tools on OpenBSD**

## **5. Kernel Operation Overview**

## **6. System Startup**

### **6.1. Linker Script and Bootstrapping**

### **6.2. Kernel Initialization**

#### **6.2.1. Assembly-language Startup**

### **6.3. Kernel Configuration**

### **6.4. Getting to main()**

### **6.5. Getting to /sbin/init**

## **7. Userland**

## **8. Future Work**

## **9. Conclusion**