243 Booker Dr.
Lonoke, Arkansas 72086

24th November 2015

Dept. Chair Prof. Dr. Yoshigoe
UALR, College of EIT, Dept. of CS
2801 S. University Ave
Little Rock, Arkansas 72204

Dear Prof. Dr. Yoshigoe,

The use of Unix-like operating systems is a hallmark of a computer science program, as well as a vital skill required of computer science students in their careers. For both historical and modern reasons, much of the most influential education material on topics such as operating system design, compilers, and networking exists solely or superiorly for Unix-like operating systems. Some of the most prestigious universities such as Princeton, Stanford, M.I.T. [5], UC Berkeley, et al., and as a growing computer science college, we benefit from following in their footsteps. As well, many of our best professors and those we wish to entice learned and prefer Unix-like systems.

In this proposal I will establish an argument against the use of Microsoft Windows (Windows), and an argument for the use of Unix-like operating systems.

1   Argument Against Windows

Windows is a popular operating system originally designed for the Intel $x86$ series of Central Processing Unit (CPU) [2]. It's long history has lead from Disk Operating System (DOS) to Windows version 3 to the newest Windows versions running the New Technology (NT) kernel.

1.1   Proprietary Nature

Windows is a proprietary operating system. This restricts a user's ability to understand to functioning of the underlying operating system because it is inherently illegal to investigate, e.g. use a debugger to dissassemle it. This is starkly contrast to the goal of an educational institution such as University of Arkansas at Little Rock (UALR). See Richard Stallman's short story depicting a dystopian world where this mindset is carried to an extreme [4].

2 Introduction

The UALR Department of Computer Science teaches exclusively on the Microsoft Windows platform. Educating using a proprietary and shrinking platform not only restricts the student's ability to understand the underlying operating system, but stunts his understanding of Unix-like systems he will likely have to work with in his career. Not only is the student disadvantaged, but he has been exposed to and taught that proprietary software is good. These teachings undermine the purpose of the university in computer science as well as the free software movement. This proposal will outline the history of Unix and its associated creations as well as Unix-like operating systems and the involvement of the free software movement, the growing market share that Linux posesses, and the moral highground that free software posseses. Following this, it will outline the changes needed to bring about a solely Unix-like Operating System (OS) based environment.

As a Linux user and avid developer and programmer, I have installed and maintained many personal Linux, OpenBSD, and other Unix-like systems. I have also programmed in many languages and produced many small programs for Linux systems. I have experience in IT as well as professional development work. In this way, I have the expertise to estimate the cost of this proposal and the effort it would take to implement a Unix-like system.

3 History of Unix-like OSs

For many years Unix-like operating systems have dominated the computer science culture. From their inception at Bell Labs in 1969 and simultaneous creation of the C and C++ languages, Unix and Unix-like operating systems have been the standard at prominent computer science universities such as the famed University of California, Berkeley. It is so ingrained in the culture of these universities that several of the important freely avaliable versions of Unix, such as the Berkeley Software Distribution, Minix, and Linux arose from them.

The C and C++ languages arose from and are best at home on Unix-like systems. As are many newer languages like Rust from Mozilla [7], and Go from Google [1] (both concurrent, practical languages). These companies put some effort into portability, but it is not crucial because the vast majority of users and programmers will be running it on Unix-like operating systems. By using Windows, the university is depriving students of the ability to experiment with the newest advancements in computer programming languages. For instance, knowledge that a language with bounds checking like Go could prevent buffer overflows could prevent another **Heartbleed** vulnerability. What better way to learn about Tony Hoar's breakthrough paper on concurrency than to see it in action with Go? In that effect, what better way to learn about networking than

to read the Linux kernel's source code and perhaps write a kernel module or use the new Berkeley Packet Filter API to filter packets to userspace programs.

The Unix-like operating systems have become increasingly prominent in computer science work in industry. For instance Apple's iOS and OS X run a proprietary version of an open source BSD distribution known as "Darwin," which itself is from another open source and widely used BSD distribution known as "FreeBSD," which descends from the original Unix through the original BSD. Android, the popular mobile phone operating system, is based on Linux, a Unix-like operating system. Which was created based partially on Minix, a microkernel OS created for educational purposes to help convey pricipals of OS development espoused in its creator, Andrew S. Tanenbaum's, book [6] To the credit of that OS, it is now an open source operating system which attempts to be highly reliable and self healing. In newer versions having such interesting features as runtime replacement of daemons via Low Level Virtual Machine (LLVM) bytecode. Another great use case is Linux itself, which runs in a variety of distributions (CentOS, Fedora, Red Hat, CoreOS, . . .) on the vast majority, 79%, of all servers. A figure that is growing according to the Linux Foundation.

The free software movement has worked for upwards of twenty years with help from heroes like Richard Stallman, GNU, and the Free Software Foundation. Although their views may sometimes seem extreme, they have brought about a near utopia of free software. Recent advancements include the domination of Linux, open sourcing of .NET, open sourcing of the LLVM toolchain [3], and open sourcing of Google's TensorFlow [8]. All huge advancements for the user as well as the programmer.

These open source systems are fast becoming the standard. In fact, the .NET core by Microsoft was implemented using the open sourced LLVM compiler infrastructure, as was the Rust compiler. The Go compiler has a counterpart in the GNU Compiler Collection (GCC) toolchain which uses that infrastructure.

Moving to a Unix-like operating system requires installing that operating system on all computers provided for student use. The vast majority of these are Linux thin client machines. This process is fairly trivial as the popular window server "X" is a server-client architecture which can be run from a central server, allowing all students to time share on the same system similar to the Windows VM system currently in use. The Linux server, which should be used as it is the most popular Unix-like operating system, could be installed on the central server.

<div align="center">Sincerely,</div>

<div align="center">Connor Taffe</div>

Acronyms

CPU Central Processing Unit. 1

DOS Disk Operating System. 1

NT New Technology. 1

OS Operating System. 2, 3

UALR University of Arkansas at Little Rock. 1, 2

Windows Microsoft Windows. 1

References

[1] The Go Authors. *The Go Programming Language*. URL: `http://golang.org/` (visited on 23/11/2015).

[2] NetMarketShare. *Desktop Operating System Market Share*. October 2015. URL: `https://www.netmarketshare.com/operating-system-market-share.aspx` (visited on 23/11/2015).

[3] LLVM Project. *The LLVM Compiler Infrastructure*. 1st September 2015. URL: `http://llvm.org/` (visited on 23/11/2015).

[4] Richard Stallman. 'The Right to Read'. In: *Communications of the Association for Computing Machinery (ACM)* 40.2 (February 1997).

[5] MIT Information Systems and Technology. *Linux Support Statement*. URL: `https://ist.mit.edu/os/linux` (visited on 23/11/2015).

[6] Andrew S Tanenbaum and Albert S Woodhull. *Operating Systems Design and Implementation (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2005. ISBN: 0131429388.

[7] The Rust Team. *The Rust Programming Language*. URL: `https://www.rust-lang.org/` (visited on 23/11/2015).

[8] Wired. *Google Open-Sourcing TensorFlow Shows AI's Future Is Data*. 11th November 2015. URL: `http://www.wired.com/2015/11/google-open-sourcing-tensorflow-shows-ais-future-is-data-not-code/` (visited on 23/11/2015).