INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 16, No. 6
December 2018
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCON (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

Server on a USB Port: A custom environment for teaching systems administration using the Raspberry Pi Zero

Michael Black
mblack@southalabama.edu

Ricky Green
reg1521@jagmail.southalabama.edu

School of Computer and Information Sciences
University of South Alabama
Mobile, AL 36688 USA

Abstract

For students seeking careers not related to the computer sciences, the task of installation, configuration and maintenance of software on business class systems may be an essential skill. Dedicated computers are essential to these kinds of courses, but the constraints of university computer labs prohibit their use. Other solutions exist, but require more time to configure and require more computer skills than are typically found in students seeking these careers. The Raspberry Pi computer, introduced in 2012, has been used with some success for this purpose. However, its portability is impaired by the need for external peripherals to make it work, such as a keyboard, mouse, and display. The Raspberry Pi can utilize a laptop for these peripherals, but this requires some additional effort. The recently released Raspberry Pi Zero eliminates much of that effort.

Keywords: teaching material, Linux, Raspberry Pi, systems administration, server applications

1. INTRODUCTION

Designing computer laboratories for Computer Science related courses brings many challenges. The challenges of designing and implementing computer laboratories are even greater for courses that are not related to Computer Science, yet still require computers configured specifically for the course. For students seeking these careers, the task of installation, configuration and maintenance of software on business class systems may be an essential skill. The amount of computer knowledge required for this type of systems administration is not as extensive as it is for courses in the computer sciences, but it is more extensive than a basic course in general computing.

A specific requirement for these types of courses is the student must possess an account with elevated permissions to the operating system installed on the computer for performing administrative tasks, such as installing and maintaining software packages. Access control at this level is almost impossible due to the risk management practices at the local level (Davison, 2015). Also, computer labs that are designed for teaching advanced computing courses are typically isolated from the campus network and do not allow full Internet access (Conlon & Mullins, 2011). These constraints can only be overcome by using something other than the traditional computer lab.

In 2012, the University of St. Andrews recognized the constraints of the typical university computer laboratory when designing an Open Access Bioinformatics research course for biology students. This research course was designed to prepare biology students with basic systems administration skills and the confidence to discover software solutions and implement them as required on a Linux based server. This discovery and implementation process required a certain amount of system administration training suitable for undergraduate bioinformatics students without the technical skills that are typically introduced to computer science students (Barker et al., 2013).

A solution to the problem must allow elevated permissions to the student, which would necessarily allow the student to render the individual computer completely unusable. These mistakes can be costly and time-consuming to correct in a typical university computer lab environment. It may not be necessary for these students to learn how to repair the damage created by these mistakes, but the recovery time from them can be considerable. The solution must provide students with adequate amounts of time to experiment with the system and complete their assignments (Owen & Black, 2003). Therefore, the ideal solution should not attempt to prevent these events from happening but must implement a mechanism for quickly recovering from these mistakes while requiring little if any additional skills beyond what students would normally experience in a traditional computer lab.

An additional requirement of the solution must include a limit on the amount of class time it takes to prepare the solution for use by the students. The ideal solution would be a platform that is preconfigured and ready to use by all students on the first day of class. In summary, the requirements must allow elevated permissions to the student, provide a quick and easy recovery mechanism, be ready to use on day one, and be easy enough to use for students that are not seeking a degree in computer science.

Approaches to Alternatives

The baseline approach is a computer lab specifically configured for the course. A dedicated computer laboratory may be considered the ideal solution for students and instructors in these types of courses because the hardware and software can be easily isolated for security purposes and can be standardized to eliminate compatibility issues. Also, it is possible to plan a quick recovery mechanism when the hardware and software are standardized in the computer lab. However, each student would be required to use the exact machine for each assignment as they build their skills. Each student would need the ability to continue where they left off, so the state of each machine needs to remain intact throughout the course. These limitations make the dedicated lab solution very inflexible since the computers would not be available for any other purpose. This inflexibility makes it a costly solution to implement since each computer in the laboratory is essentially dedicated to a single student for the duration of the course.

A similar approach is to utilize a laptop for each student. Unlike the dedicated computer lab, this approach is specifically designed to provide a dedicated machine per student throughout the course. This solution is flexible if the laptop is compatible with the requirements of the course and remains serviceable. These laptops can be issued by the University, which may help with consistency in hardware configuration and support, but may be considered redundant when most students already possess their laptop and carry it with them to classes (Kay & Lauricella, 2014). Student-owned laptops are not always standardized, which presents potential compatibility issues with the software to be used in the course. Students will be tasked with installing and configuring the software environment on their laptops for the course, which requires additional time and computer skills that the student may not possess. Also, there is no quick recovery mechanism in the event of a mistake when using student laptops. Utilizing university owned laptops could eliminate many of these constraints, but this approach is at least as expensive to implement and maintain as a traditional computer lab while maintaining the same inflexibility.

An iterative approach is to provide students with standard virtual machine images using virtualization software such as VirtualBox or VMWare Workstation. These virtual machine images can be run on any computer, from lab computers to student laptops. Since the hardware is identical on each virtual machine, and the virtualization software is available for all major operating systems, this approach can help eliminate the inconsistent configurations and compatibility issues including the inconsistent hardware among the student laptops. Many of these virtual machine images can reside on the same hardware and booted on demand, reducing the inflexibility of the dedicated computer lab approach (Murphy & McClelland, 2009). This approach takes additional time and skill to download and import the virtual machine images, especially when using student laptops. Recovery from most mistakes can be quick but requires a few more skills than using a real computer with a locally installed operating system. The virtual machine solution eliminates many problems but still fails in the requirement of being ready to use on the first day and requires a few extra skills to use effectively.

Another approach is to utilize Live CD's or bootable USB sticks that are configured to boot an entire operating system that is independent of the operating system installed on the computer itself. This approach is portable, inexpensive and flexible, but may require configuration changes to the computer to enable it to work. Newer computers and laptops have replaced the standard BIOS with UEFI (Unified Extensible Firmware Interface) and Secure Boot, which was designed to prevent the operating system from being compromised via rootkits (Wilkins & Richardson, 2013). As a result, these computers will not allow a USB stick to boot without making changes to the system itself. These changes require additional skill and time to setup and may render the installed operating system inoperative until the changes are reversed.

A central server approach with a separate login for each user is a flexible approach for most computer laboratories. This central server approach requires a small amount of initial setup for the student to install the client software on their laptop for remote control and may allow them to access the server from home. This approach incurs a high upfront cost to implement the hardware and infrastructure required to support the laboratory environment. It is also unwise to grant elevated permissions to multiple students on a central server, as one mistake by a student can take down the entire host. Because administrator access cannot be allowed, the shared server approach cannot be used for the key understanding of software installation and configuration (Barker et al., 2013). Due to this limitation, the central server approach is not suitable for this type of course.

A hybrid approach consisting of individual virtual machines on a central server could be a flexible solution. Each virtual machine could be configured for each student with the appropriate permissions, yet isolated from the host to eliminate the security risk of multiple users holding elevated permissions. This Virtual Computer Lab solution maintains the high upfront costs associated with the central server approach and may incur additional costs due to the higher hardware demands for executing multiple virtual machines simultaneously (Murphy & McClelland, 2009). This solution also requires additional administration for the host operating system and the supporting infrastructure. However, it does not require student time for setup, specific skills to use, and has a very quick recovery mechanism if the host administrator is available.

An alternative approach that is gaining popularity is the use of an inexpensive computer called the Raspberry Pi, which is the approach that was eventually used by the University of St. Andrews for the Bioinformatics course. In their approach, the University of St. Andrews loaned a Raspberry Pi computer and associated peripherals to students for the duration of the course (Barker et al., 2013). The Raspberry Pi is a very small, inexpensive single board computer that runs the Linux operating system. The peripherals consisted of a keyboard, mouse, power adapter, a preconfigured SD card with the operating system and associated applications, and a USB stick for backups. An LCD could not be issued as a peripheral, but the students could use an LCD from the computer lab during class. A lab kit such as this allows students to take the lab equipment home to complete their homework and experiments on their own time (Reck & Sreenivas, 2016). Even with the device limitations and the costs of the extra peripherals, the Raspberry Pi was considered an effective and low-cost approach (Barker et al., 2013). The Raspberry Pi solution allowed students to have full access to a suitable operating system on dedicated, standardized hardware without the limitations of the traditional approaches.

Reasoning for the Raspberry Pi Approach

The ideal solution is a dedicated computer for each student, with full administrative access so students can build their systems management skills throughout the course. The constraints of traditional and advanced computer labs essentially prevent their use in these non-computer science related courses (Barker et al., 2013). The Raspberry Pi device lowers the entry cost of a dedicated lab to the point where flexibility and cost are no longer a major concern. The Raspberry Pi approach presents a standard platform that meets the requirements for many system administration courses.

Problems with the Raspberry Pi Approach

Although the Raspberry Pi is a model for power efficiency and low cost, it's processing power and memory is limited. At the time the Bioinformatics course at the University of St. Andrews was initiated, the only model of the Raspberry Pi available was the $35 Model B, and the stripped down $25 Model A that was designed for low power embedded applications. These tiny computer boards were designed for power efficiency and low cost at the expense of processing and memory limitations. These early models utilized a single core 700Mhz 32bit ARMv6 processor with 512MB of RAM. The performance of this small computer is far slower than modern desktop and laptop computers, but the processing power was deemed more than adequate for the task of the bioinformatics administration course (Barker et al., 2013). Current models of the Raspberry Pi computer have greatly increased processing capability and memory up to 1GB for the same $35, but are still far behind the performance of standard computers and laptops. For other courses that require more processing power or memory, the Raspberry Pi may not be an adequate solution.

For courses where the processing power of the Raspberry Pi is adequate, the most cumbersome issue with the Raspberry Pi approach is the requirement of additional peripherals to make use of them. These peripherals also increase the both the cost of the solution and the amount of equipment the students must carry with them if they are to use them outside of the classroom.

Overcoming Problems with the Raspberry Pi Approach

External peripherals do not need to be large and cumbersome. Portable external accessories are available that can lighten the load of the student. Portable wireless keyboards with built-in touch pad pointing device, such as the $40 Logitech K400, are much easier to stow in a backpack than typical keyboards and reduces two peripherals to one. For the display, the $70 official Raspberry Pi 7” Touchscreen accessory utilizes the dedicated display connector on the Raspberry PI, providing 800x480 display resolution. When mounted in an appropriate case ($20-$30), the Raspberry Pi will essentially become a portable all-in-one touch enabled computer for under $200. However, at this price point, the cost of the system is in the territory of an inexpensive laptop that can provide much better performance without requiring extra time and skill to assemble all the pieces.

A potential solution to the problem with external peripherals is to use the Raspberry Pi without them. The Raspberry Pi can be configured to connect to a laptop using an Ethernet cable, allowing the laptop to create a remote session to the Raspberry Pi over this network connection. The Model B versions of the Raspberry Pi include built-in Ethernet ports that support 10/100Mbit Ethernet. The Ethernet ports on the Raspberry Pi feature auto crossover detection, so that they can be used with standard Ethernet patch cables. A short network patch cable between the Ethernet port of the Raspberry Pi and the Ethernet port on the laptop provides the necessary physical network connection between them. The Raspberry Pi board can be powered directly from a USB port on the laptop, so a short USB to MicroUSB cable can handle this function. A small amount of configuration on both the Raspberry Pi computer and the laptop is required to establish the network connection. Much of this configuration can be scripted or preprogrammed on the SD card image on the Raspberry Pi before the course starts.

Illustration 1: Raspberry Pi 3 configured as a portable computer with a 7-inch touch screen and portable wireless keyboard.

In late 2015, the Raspberry Foundation released a new model of Raspberry Pi called the Zero. This board is less than half the size of the traditional Raspberry Pi boards, yet retains full hardware compatibility with them. The reduction in board size also reduced the components on the board as well, creating a simplified design and lowering the cost to $5 per board. This board is deceptively limited, with only a single usable microUSB port for communication, a separate microUSB that is only used to supply power, and a mini HDMI port available for the display. The Raspberry Pi Zero can be used in a traditional mode, but will require an adapter for the mini HDMI port and a special USB OTG hub to break out the single microUSB port into several full-sized USB ports. The latest version of the Raspberry Pi Zero board now includes built-in Bluetooth and Wi-Fi capability, further reducing the number of peripherals required.

The hardware specifications for the Raspberry Pi Zero are nearly identical to the specifications of the original Raspberry Pi. The processors are the same model, but the Raspberry Pi Zero’s single core processor is clocked at 1Ghz to provide a noticeable speed boost over the older board. The memory remains the same as the older board at 512MB. What differentiates the Raspberry Pi Zero from all other models is that the microUSB communication port is configured as a USB OTG port, allowing it to act as either a USB host or client device. This new feature presents another approach that has the potential to reduce many of the problems of the traditional Raspberry Pi based solution.

With the USB OTG port and the Linux-based operating system that runs on the Raspberry Pi Zero, the device can present itself to a host computer as a USB peripheral that is defined by the software running on it. These devices include a keyboard, mouse, Ethernet adapter, Serial adapter, and USB Memory stick to name a few. The host laptop identifies these virtual USB devices and automatically loads the appropriate drivers for them when it is connected. When properly configured, the Raspberry Pi Zero can also present itself to the host computer as any combination of these devices at the same time. The only hardware required is the Raspberry Pi Zero board, a preconfigured micro SD card, the host laptop provided by the student, and a single USB cable that supplies both power and the communication connection to the Raspberry Pi Zero. The USB OTG port on the Raspberry Pi Zero makes the "Server on a USB port" solution possible.

2. HOW WE TESTED

Setup Environment

The Raspberry Pi Zero was configured to present itself to the host laptop as a combination of an Ethernet adapter, Serial Adapter, and USB Memory stick. By presenting itself as a USB Ethernet adapter for the laptop, the Raspberry Pi Zero can establish a TCP/IP based network between itself and the laptop just as a standard Raspberry Pi would when using an Ethernet patch cable. By presenting itself as a USB Serial port, the Raspberry Pi Zero can communicate with the host using a standard terminal program for text-only mode computing. By presenting itself as a USB Memory stick to the laptop, the Raspberry Pi Zero can store preconfigured programs and documentation that are usable on the host system to connect to the Raspberry Pi Zero itself. As a result, this approach presents all the benefits of the traditional Raspberry Pi approach using the laptop as a host, while reducing the time, skills, and extra equipment required for configuring the host laptop to connect to the Raspberry Pi.

Illustration 2: Proper connection of microUSB cable when connecting to a laptop.

Student Environment

Once the Raspberry Pi Zero is configured, the only requirements for using it are the Raspberry Pi device itself, the microSD card, a laptop computer, and a USB cable (Type A to microUSB). To use the device, connect the microUSB side of the USB cable to the Raspberry Pi Zero port labeled USB (Illustration 2) then connect the other end of the USB cable to the USB port of the laptop. The Raspberry Pi Zero LED will illuminate, and the boot sequence should be complete within 90 seconds. Once booted, the operating system installed on the laptop will detect the new USB hardware and install the drivers. One of these drivers will be for the mass storage device, which should be mountable to the host operating system and function just like a USB thumb drive with full permissions. Once mounted, this drive is configured to contain user documentation and applications suitable for the host Operating System installed on the laptop. The supported host Operating Systems include Windows, Apple OSX, and Linux. The applications consist of portable versions of PuTTY and VNC programs, each preconfigured to connect to the Raspberry Pi Zero itself over the USB network adapter driver. Each set of programs are organized in separate folders labeled for each of the compatible host Operating Systems. The student selects the folder appropriate for their laptop Operating System and executes the appropriate application. The VNC application presents the student with the graphical remote desktop interface of the Raspberry Pi Zero, and the PuTTY application presents the student with a remote terminal session to the Raspberry Pi Zero.

Illustration 3: Raspberry Pi Zero connected to a Windows 7 laptop with a single USB cable for power and communication. Shown with PuTTY and VNC sessions open.

3. RESULTS AND DISCUSSION

As of late 2016, 11 million Raspberry Pi computers were sold throughout the world since its release in 2012 (Thomas, 2016). Originally designed to promote computer science education in developing countries, the Raspberry Pi became much more than its developers anticipated. This computer is now in the hands of millions of people all over the world who are contributing software, hardware accessories, documentation, and training materials. All Raspberry Pi models are compatible with each other, including the new Raspberry Pi Zero that is the main topic of this paper.

The Raspberry Pi as a platform provides a general-purpose computing environment suitable for most courses designed to train students in administrative functions in a Linux based environment. It has the potential to meet all four of the requirements listed in the ideal solution when compared to the dedicated computer laboratory. It is less expensive than dedicated computer laboratories or server based solutions, and it is easier and quicker to implement and start to use than virtualized servers. Once the configuration is complete, the Operating System image can be replicated to multiple SD cards and distributed to the students with the Raspberry Pi board and cable. By configuring the SD card images ahead of time, the Raspberry Pi devices can be ready to use by the students on the first day of class. Backups of user data can be performed using built-in Linux utilities, and complete recovery from a mistake is as simple as swapping out an SD card for another pre-configured card.

4. CONCLUSION

The Raspberry Pi computing platform has already made history in teaching computing topics in all levels of education, from elementary schools to University campuses. It is recognized for its ease of use, extreme low cost, and the support it is receiving by educators, students, and enthusiasts throughout the world. The non-profit Raspberry Pi Foundation that manages the development of the platform has ensured that future versions of the platform are released slowly and remain compatible to ensure long term viability. The entire platform is based on Free and Open Source concepts, including most of the hardware and its GNU/Linux based operating system.

This paper began as a study towards making the Raspberry Pi less cumbersome for students by eliminating the need for using external peripherals by utilizing the student's own laptop computers. An exhaustive search through many peer-reviewed articles produced several instances of Raspberry Pi use in academia, but none of them discussed using the platform in the classroom with this configuration. One paper discussed the problem with external peripherals as a significant drawback when using the Raspberry Pi in the classroom (Barker et al., 2013).

Using two separate computers in a direct-attached networked configuration is not new or revolutionary. It is quite common for Raspberry Pi developers and enthusiasts to connect the device to their laptops with an Ethernet cable for communications and a USB cable for power. This is done to enhance portability and make use of advanced tools installed on the laptop to aid in development. The Raspbian Operating System that runs on the Raspberry Pi already has everything it needs for remote connection from a laptop, including SSH and VNC server services. Client software for the laptop, such as PuTTY and VNC client, is freely available online for Windows, Mac OSX and Linux operating systems. The most difficult part of this configuration is downloading, installing, and configuring the software and network settings on the host computer.

The Raspberry Pi Zero was introduced to the market as this study was being conducted, and presented a potentially simplified method of connecting the Raspberry Pi to a laptop. The USB2GO interface coupled with the GadgetFS file system available in the Raspbian operating system eliminates the need for the Ethernet patch cable and allows the device to automatically configure the network settings for itself and the host. The emulated mass storage device can contain files for the host, such as preconfigured portable versions of the afore-mentioned PuTTY and VNC client. Adding these programs to the device itself removes the need to download, install and configure local versions of these programs on the host computer. From the perspective of the student, the setup and configuration process is not just simplified, it is eliminated.

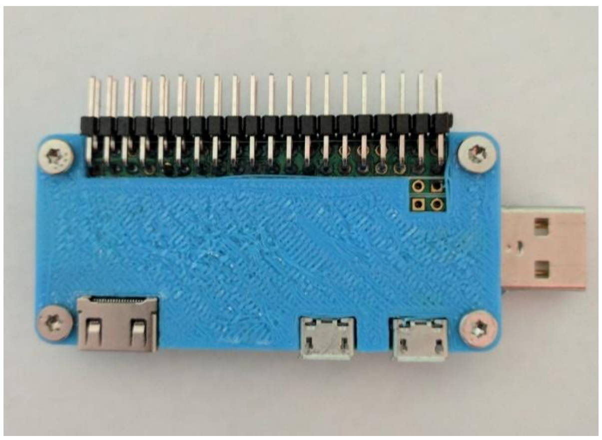

Illustration 4: Raspberry Pi Zero modified with built-in USB connector, GPIO Header, and 3D printed case.

5. REFERENCES

Barker, D., Ferrier, D. E., Holland, P. W., Mitchell, J. B., Plaisier, H., Ritchie, M. G., & Smart, S. D. (2013). 4273π: Bioinformatics education on low cost ARM hardware. *BMC Bioinformatics*, *14*, 243. https://doi.org/10.1186/1471-2105-14-243

Conlon, M. P., & Mullins, P. (2011). ISEDJ Creating and Using a Computer Networking and Systems Administration Laboratory Built Under Relaxed Financial Constraints. *Information Systems Education Journal*, *9*(4), 4.

Davison, C. B. (2015). Assessing It Student Performance Using Virtual Machines. *Tech Directions*, *74*(7), 23–25.

Kay, R. H., & Lauricella, S. (2014). Investigating the Benefits and Challenges of Using Laptop Computers in Higher Education Classrooms. *Canadian Journal of Learning and Technology*, *40*(2). Retrieved from https://libproxy.usouthal.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=eric&AN=EJ1030425&site=eds-live

Murphy, M., & McClelland, M. (2009). My Personal Computer Lab: Operating in the "Cloud." *Information Systems Education Journal*, *7*(93). Retrieved from http://isedj.org/7/93/index.html

Owen, W., & Black, M. (2003). Designing Labs for a Sequence of Network Courses. *Information Systems Education Journal*, *1*(31). Retrieved from http://isedj.org/1/31/index.html

Reck, R. M., & Sreenivas, R. S. (2016). Developing an Affordable and Portable Control Systems Laboratory Kit with a Raspberry Pi. *Electronics*, *5*(3), 36. https://doi.org/10.3390/electronics5030036

Thomas, J. (2016, November 27). Raspberry Pi computer sales hit 11 million. Retrieved April 1, 2017, from http://www.cambridge-news.co.uk/news/cambridge-news/raspberry-pi-computer-sales-hit-12237410

Wilkins, R., & Richardson, B. (2013, September). UEFI Secure Boot in Modern Computer Security Solutions 2013. Retrieved from http://www.uefi.org/sites/default/files/resources/UEFI_Secure_Boot_in_Modern_Computer_Security_Solutions_2013.pdf

Annexures

Platform/approach	Cost to implement/maintain per student	Allows elevated permissions	Setup time	Recovery time	Skill level to implement/use
Typical computer lab	$$$$	-	-	-	-
Dedicated computer lab	$$$$	X	High	High	Low
Dedicated laptop – university issued	$$$$	X	High	Medium	Low
Dedicated laptop – student owned	-	X	High	High	High
Virtual machine on student laptop	-	X	High	Medium	High
Virtual machine on shared server	$$$$	X	Medium	Medium	Medium
Live CD/USB	$$	X	Low	Low	High
Central server with remote access	$$$$	-	-	-	-
Traditional Raspberry Pi with peripherals	$$$	X	Low	Low	Low
Raspberry Pi Zero with student laptop	$$	X	Low	Low	Low