

A Nomadic Testbed for Teaching Computer Architecture

Pablo D. Godoy^{1,2,3}, Carlos G. García Garino^{1,3}, and Ricardo L. Cayssials⁴

¹ITIC, Universidad Nacional de Cuyo, Mendoza, Argentina

²FCEN Universidad Nacional de Cuyo, Mendoza, Argentina

³Facultad de Ingeniería, Universidad Nacional de Cuyo, Mendoza, Argentina

⁴Universidad Tecnológica Nacional - FRBB, Buenos Aires, Argentina

{pablodgodoy@gmail.com}

Abstract. A nomadic laboratory or testbed, based on Raspberry Pi 3 computers and Arduino microcontrollers, has been developed in order to teach subjects related to computer architecture. The testbed can be transported to the classroom. Students can access it through the available network, which can be a wireless LAN, wired LAN or a custom network. The student can access without constraints to the platforms, therefore there are a wide range of possible experiments. This laboratory was used during 2017 for practical works in the course Introduction to Technology, and during 2018 in the course Computers Architecture at Universidad Nacional of Cuyo. Some of the experiments that are been carried out by students are: to explore and analyse the architecture of the computers through Linux commands, write and run programs on different programming languages, input and output operations through memory mapped addressing and isolated addressing, write interrupt service routines in order to service interrupts, multithreading programming, explore memory maps, CPU features, etc. This paper describes the testbed architecture, experiments performed by students in the mentioned subjects, present the students feedback, and describes the possible methods in order to integrate it to a remote laboratory.

Keywords: Nomadic testbeds; Teaching platform; Computer architecture

1 Introduction

Computer architecture is a central subject in careers such as engineering or bachelor degrees in computer, information systems, electronics, information and communication technology, etc. Laboratory experiences and practical activities about computer architecture usually implies to recognize all components, their functions and features in different computer systems, such as the processors, data memories, data buses, bridges, etc., to employ operating system tools to visualize and analyze the processors workload, memory usage, peripheral status, etc., and to build programs to perform input/output operations, multithreading

task over superscalar processors or multiple core processors, etc. In order to perform laboratory experiences and practical activities, students need to access to different kind of computers, in order to understand and compare different computer architectures. As a result, it is very useful to build platforms to enable students carrying out practical works and laboratory experiences.

University computer laboratories usually have computers based on x86 processors (Intel, AMD, etc.). This limits the kinds of architectures over which students carry out experiments. As a result, it is useful to deploy platforms and facilities with different architectures, for example: ARM, in order to give students more options about different computer architectures for performing experiments.

In the literature several experimental platforms have been proposed, with educational and scientific purposes [1][2][3][4]. Horneber and A. Hergenröder [2] distinguish between two kinds experimentation platforms or testbeds, nomadic testbeds and remote testbeds. Nomadic testbed can be completely or partly movable to the place where the experiments will be performed. The main advantage of nomadic testbeds for teaching purposes is that students can put hands on the equipments. But the main disadvantage is the limited time for performing the experiments. Remote testbeds can be accessed remotely through Internet. The advantage of remote testbeds is that they can be accessed all time, but the main disadvantage is that students can not put hands on the equipments under test. Our platform can be entirely transported to the classroom in order to students carry out experiments. But also can be remotely accessed by ssh protocol. Therefore our platform have features of both kind of experimental platforms.

However, until today our platform only has been used as a nomadic testbed. As a result, we classify our platform as a nomadic testbed for teaching purposes, with capacity to be transformed in a remote and nomadic testbed.

The main contributions of this papers are two:

- The proposed architecture to build a nomadic experimental platform to perform experiments related with computer architecture based on ARM processors.
- The proposed experiments for putting in practice theoretical knowledges about computer architectures, that students acquire during theoretical classes.

2 Remote Laboratory Architecture

2.1 Equipment

The central part of the laboratory are Raspberry Pi 3 computers [5]. The election of this computer is due to they have processor of four cores, which enables multicore and multithreading experiments. These kinds of experiments are needed in the last part of the subject *Computer Architecture*, in order to students can write programs over parallel architectures. On the other hand, computer laboratories for teaching purposes usually are built with PC computers

with x86 processors. One objective of our proposed testbed is to use ARM processors, in order to extend the architecture types over which students carry out practical works. In addition, the small size of these computers are a very important feature due to the laboratory must be transported. An auxiliary board with leds and buttons is attached to the Raspberry Pi 3 computer. The leds and buttons are connected to the general purpose input/output pins of the Raspberry Pi 3, to enable performing input and output experiments

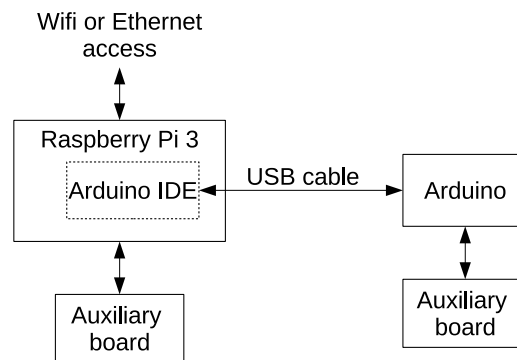
In addition, an Arduino UNO board [4, 6] is attached to each Raspberry Pi 3 computer, for enabling experiments over a simpler computer and over other kind of peripheral, like analogical to digital converters, etc. An auxiliary board with leds and buttons is attached to the Arduino board. The leds and buttons are connected to the general purpose input/output pins of the Arduino, to enable performing input and output experiments. The Arduino Integrated Development Environment (IDE) is installed on the Raspberry Pi 3 computer. Both computers are attached by a USB cable.

The equipment is completed by:

- The teacher computer: that enables monitoring the students activities through ssh or vnc protocols.
- An access point: that enables wireless access to the Raspberries.
- Ethernet cables: that enable wired access to the Raspberries.

The kind of access is selected in function of the infrastructure and network available in the classroom. The figure 1 shows a block diagram of a node.

Fig. 1. Node block diagram



2.2 Access to the Equipment Under Test

The access to the equipment under test is through SSH or VNC protocols. In some experiments students need to consult configuration files or run Linux

commands. To other experiments, students need to write and run programs. To almost all activities, both SSH and VNC can be used equally. The VNC protocol is mandatory only to employ the tools provided by the Raspberry Pi operating system. Both tools, SSH and VNC, have been used in several testbeds presented in the bibliography, for example: [7][8]. The experiences carried out during 2017 and 2018 shown that students prefer VNC, maybe due to students prefers to interact through a graphical user interface (GUI) proper of desktop operating systems. As a result, students are encouraged to use both: SSH and VNC, in order to acquire experience in the use of a wider range of tools.

In order to access to the equipment, three methods are provided:

- Through Ethernet network.
- Through Wifi Network.
- Through a cloud services provider accessible through Internet.

The first method can be used if there is an Ethernet network deployed in the classroom. The second method, through Wifi, is always available due to the platform has an Wifi access point as a part of its equipments, and each Raspberry Pi 3 computer can be used as an access points for deploying a wireless network. The access through a cloud platform was implemented using the services of Remote-IoT [9]. Through this service, a computer can be accessed remotely through Internet, via SSH and VNC protocols. The access through a cloud platform will be investigated and analyzed in greater details in future works. The three methods were used during 2017 and 2018, in the courses of *Computers Architecture* and *Introduction to Technology*, with successful results.

3 Performed Experiments

In this section, some experiments carry out by students during 2017 and 2018 over the presented testbed are described.

3.1 Architecture Analysis

In these experiments, students analyse the architecture of a computer through Linux commands. The commands employed were:

- lshw (HardWare LiSter for Linux): tool to show information about the hardware configuration of the computer. The information shown by lshw include: (a) memory configuration, (b) CPU features and speed, (c) cache memory configuration, (d) type of buses and speeds, (e) bridges between buses, (f) massive storage devices, (g) different peripheral, etc.
- lspci: shows information about PCI buses and devices attached to these buses.
- free: shows information about memory, used and free physical and swap memory, and memory used as buffers.
- uname: displays information about the operating system (especially the options: -srpo)

- df: shows information about the hard disk (option used: -h)
- ifconfig: displays information about the network interfaces.

These commands and most of their options and output formats have been selected for performing this practical activity (see [10] for detailed information about these commands and their options). The activity carried out by the students consisted in to execute these commands, to analyze the information produced, to draw a block diagram of the computer architecture and specify their characteristics.

3.2 Input-Output Operations

The objective of this activity is to distinguish between memory mapped and isolated addressing of input/output devices [11]. The practical work consists of switching off and on leds and to read the state of input pins through two methods:

- Memory mapped I/O devices
- Isolated I/O devices

For Arduino processor, students can use both methods, and for Raspberry Pi 3 only memory mapped access is available. In order to see the differences between these two methods, students must program the processors in assembler language, since a high level language hides these details for programmers. The students write code segment in assembly language embedded in C, C ++ or Python language (known languages for students).

3.3 Writing of Interrupt Service Routines

Writing of Interrupt Service Routines in Arduino UNO The practical activities consist of writing service routines to attend interrupts caused by signals detected in input pins. For this purpose, students are provided with a guide about Arduino IDE programming (obtained from [6]) and the instruction set of the ATmega328P processor [12], that is the processor used by the Arduino UNO platform. In addition, a pre-written code that include the scheme of an Arduino program in C language and the code to embed assembler instructions are provided.

Students write service routines for the two interrupt sources of the Arduino Uno platform, the pins 0 and 1. These service routines inform when an interrupt is detected. After, students carry out a set of actions to analyze the interruption priorities.

Writing of Interrupt Service Routines in Raspberry Pi 3 A similar activity is carry out over Raspberry Pi 3 platform. Two differences with Arduino UNO activity are proposed to students:

- Write programs using Python language.

- Use the configuration files of the Linux filesystem in order to control the hardware instead of assembly language.

These changes have the purpose of give new knowledge to students.

Similar to the previous activity, students write routines of service for two interrupt sources of the Raspberry Pi 3 platform. These interrupts are triggered by signals over different input pins. These service routines inform when an interrupt is detected. After, students carry out a set of actions to analyze the interruption priorities.

In addition, students can analyze interrupts statistics through the `/proc/interrupts` Linux file.

3.4 Multithreading Programing

The objective of this activity is to show to students the speed-up that can be achieved in some task using multithreading programing. For this activity, Raspberry Pi 3 computers were used due to they have processors with four cores. Because students at this points of theirs career have reduced knowledge of programing, this activity was performed using C++ language, due to it enables to write multithreading programs in an easy way. Other options (like Python, Java or C) require to use classes and objects, and students still does not have knowledge about these topics. The activity consists of writing a program that uses a 200x200 matrix of float point variables, and several subroutines that perform the following tasks:

- Modify a randomly chosen elements of the matrix with a random value every time that this function is called.
- Calculate the root mean square value of all elements of the matrix.
- Obtain the highest value inside the matrix.
- Find the row whose sum of their elements has the highest value.

After verify the proper execution of their programs, students verify some statistic of execution through Linux commands, like number of threads, cores usage, execution speed of every function, etc.

Then, students are instructed to divide their programs in several threads using the functions provided by C++ language. For simplicity, students are suggested to execute every subroutine in a different thread. After verify the proper functioning of their programs, students verify again the above mentioned execution statistics, and compare against the prior values. Students can see how the parallel execution of their functions increases the processor usage and the speed of every function.

Some students, for their own initiative, went beyond what was requested them and divided their programs in more threads (a group divided its program in 64 threads), and a competition for achieve a higher number of threads was generated between students. Students obtained higher and lower speed-ups. This result allowed showing to students that the parallelization must be performed in an efficient way (students will acquire knowledge about efficient parallelization and execution of tasks in later subjects of their career).

3.5 Interference in Wireless Networks

The objective of these experiments was to show the effect of interference on wireless networks (IEEE 802.11). For this purpose, the Raspberries Pi 3 computers were used as wireless access points of IEEE 802.11 wireless networks on infrastructure mode. Wireless access points based on Raspberry Pi 3 computers are chosen due to they enable to configure a wider range of parameters than commercial wireless access points, like the interframe time for different kind of data.

The experiments performed by students consisted of measuring the following parameters:

- Data rate while a large size file is transferred.
- Latency for a ping command during the transference.

These parameters were measured under the following conditions:

- Three wireless access points in different wireless channels.
- Three wireless access points in the same wireless channel with time interframe values similar to commercials wireless access points.
- Three wireless access points in the same wireless channel with time interframe values modified a their minimum value [13].

The measurements were performed while three users were transferring big size files between the wireless access points to their computer through SSH, being connected every user to a different wireless access point. The values of the above mentioned parameters were obtained through the values displayed by the ping and SCP (secure copy protocol) commands.

Students can see how the channel occupation affect the individual data rate and latency. The methodology of these experiments was obtained of a similar experiment, carried out with scientific purposes, using a wireless sensor network [14].

4 Answer to an Anonymous Survey about the use of the Testbed

Eighteen students that took the course “Technology Introduction” of the Computer Science career at Universidad Nacional de Cuyo, were invited to answer an anonymous survey about the use of the platform. The questions of the survey and the students answers were:

1. ¿Was the proposed platform useful in your process of acquiring new knowledge?
Answers:
 - (a) It was not useful: 0%
 - (b) It was useful: 20%
 - (c) It was very useful: 80%

2. ¿Were the experiments interesting and amused?

Answers:

- (a) The experiments were boring and no interesting: 0%
- (b) The experiments were interesting and amused: 10%
- (c) The experiments were very interesting and amused: 90%

3. ¿Were the experiments easily to perform and could you understand the theoretical concepts behind each activity?

Answers:

- (a) The experiments were complicated and I could not understand the concepts: 0%
- (b) The experiments were easy to follow and I could understand the concepts: 20%
- (c) The experiments were easy to follow and I could understand very well the concepts: 80%

In addition, student were invited to make suggestions and comments. Some of these sugerencias and comments (literally transcribed) were:

- “*It was very amused and motivating*”.
- “*It was a very good experience*”.
- “*It would be good to have more time to work with the platform*”.

5 Future Work

Remote laboratories, also known as remote testbeds, are platforms that allow remote access to different types of equipments, devices, laboratories, etc. through a LAN networks or Internet. They are complex systems that include a large number of components with functions such as: access control, monitoring of activities, information storage, etc. They may be aimed at scientific research, application development, training, teaching, etc. [3, 15, 16].

Cloud computing is a model for allowing on demand and ubiquitous access to a configurable, virtualized and shared computational resources set (for example networks, servers, storage, applications and services), that can be quickly provided and released with minimal management effort and interaction with the cloud services provider. These resources are offered by cloud computing services providers to external customers through Internet as web services, based on negotiated agreements between the service provider and customers [17].

The immediate future work will be to integrate the proposed testbed to a cloud computing platform, similar to [18, 19]. The objective of this integration is to provide access to the testbed all time through Internet, overcoming the time limitation for accessing the nomadic testbed mentioned by students.

6 Conclusions

The experiments performed by students during 2017 and 2018 show the versatility of the proposed platform for performing different kind of experiments, including:

- Analyze the internal architecture of different computers.
- Perform input and output operations and to write interruption service routines.
- Basic parallelization of tasks.
- Analysis of wireless networks behavior.

During the experiments it can be noted that students adopted very well the proposed educational testbed, and they wanted to exploit it adding activities different to the proposed ones. In addition, the feedback obtained from students through an anonymous survey was very positive. This shows the motivation of students to work with new and different technology. For this reason, it is needed to increase the number of nodes of the platform and to add a mechanism that enable remote access to the platform. In this way, students would be able to access the platform from their homes.

Acknowledgment

The authors acknowledge the financial support received from the *Universidad Nacional de Cuyo* through the project *B041 "Implementación de laboratorios remotos basados en cloud computing"*. Also, authors acknowledge to the professors *Lucas Iacono* and *Oswaldo Marianetti*, professors of *Introduction to Technology* and *Computer Architecture* respectively, at *Universidad Nacional de Cuyo*, for allow us to use our platform on their courses.

References

1. L. Steyn and G. Hancke, "A survey of wireless sensor network testbeds," in *AFRICON, 2011*, pp. 1–6, sept. 2011.
2. J. Horneber and A. Hergenröder, "A survey on testbeds and experimentation environments for wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1820–1838, 2014.
3. P. Orduna, A. Gomez-Goiri, L. Rodriguez-Gil, J. Diego, D. Lopez-de Ipina, and J. Garcia-Zubia, "wCloud: Automatic generation of WebLab-Deusto deployments in the Cloud," in *Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on*, pp. 223–229, Feb 2015.
4. M. Hakima, B. Abdelhalim, and B. A. Adda, "Design of a low cost remote electronic laboratory suitable for low bandwidth connection," *Computer Applications in Engineering Education*, vol. 25, no. 3, pp. 480–488.
5. Raspberry Pi Foundation website, "Raspberry pi 3 single-board computer." Retrieved in 2018.
6. Arduino s.r.l, "Arduino uno rev3 documentation." Retrieved in 2018.

7. M. N. Kabiri and M. Wannous, "An experimental evaluation of a cloud-based virtual computer laboratory using openstack," in *2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, vol. 00, pp. 667–672, July 2017.
8. T. Gerhard, D. Schwerdel, and P. Müller, "A networkless data exchange and control mechanism for virtual testbed devices," in *Testbeds and Research Infrastructure: Development of Networks and Communities* (V. C. Leung, M. Chen, J. Wan, and Y. Zhang, eds.), (Cham), pp. 14–22, Springer International Publishing, 2014.
9. Remote-IoT, "Remote-iot website." Accessed in 2018.
10. Canonical Ltd, "Ubuntu manpages website." Accessed in 2018.
11. W. Stallings, *Computer organization and architecture designing for performance*. Pearson, 9 ed., 2013.
12. Microchip Technology Inc., "DS40001984A: Complete Datasheet of the ATmega328/P." Accessed in 2018.
13. IEEE Institute, "IEEE std 802.15.4-2003," pp. 1–670, 2003. Retrieved in 2015.
14. P. D. Godoy, R. L. Cayssials, and C. G. G. Garino, "Communication channel occupation and congestion in wireless sensor networks," *Computers & Electrical Engineering*, 2018.
15. P. Godoy, R. Cayssials, and C. García Garino, "A WSN testbed for teaching purposes," *IEEE Latin America Transactions*, vol. 14, pp. 3351–3357, Jul. 2016.
16. M. Waldrop, "Campus 2.0," *Nature*, vol. 495, no. 7440, pp. 160–163, 2013.
17. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599 – 616, 2009.
18. P. Godoy, R. Cayssials, and C. García Garino, "A cloud based WSN remote laboratory for user training," in *Proceedings of TE&ET (Technology in Education and Education in Technology)* (U. of Morón, ed.), 2016.
19. P. Godoy, R. Cayssials, and C. García Garino, "Laboratorio remoto para la formación de usuarios basado en el cloud," *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación*, pp. 7–18, 12 2016.