Linux Online Virtual Environments in Teaching Operating Systems

Olena Holovnia[0000-0003-0095-7585]

Zhytomyr Polytechnic State University, 103, Chudnivska str., Zhytomyr, 10005, Ukraine
olenaholovnia@gmail.com

Abstract. The article provides an overview of Linux ready-made online environments based on various virtualization technologies. Linux ready-made online virtual environments are systematized, the advantages and disadvantages of considering groups of online virtual environments are listed. The work also gives the comparison of standalone online Unix/Linux terminals in terms of their possible use for a course in operating systems. The author analyzes the experience of teaching a Linux operating system to students of different specializations, and in particular, discusses the virtualization tools used, including Linux online environments (Amazon EC2, Cloud 9, a virtual machine for NDG Linux Essentials course). The study states the need for developing an individually designed system to provide Linux online virtual environments and elaborates the requirements to this system. The work also discusses potential challenges which the developers of Linux online virtual environments for teaching operating systems may face, including the problem of finding a balance between availability and stability of virtual environments on the one hand and the realism of students' learning experience on the other hand.

Keywords: virtualization, Linux, operating systems.

1 Introduction

A course in operating systems (OS) usually includes practical assignments on Unix-like operating systems (particularly, Linux). However, at the same time, university lab computers, students' laptops and home desktop computers mostly come with Windows installation. That is one important reason why different virtualization technologies are often used for a course in operating systems. These technologies enable working with a target operating system regardless of a locally installed operating system and without compromising the security of university lab computers. Students, who take a course, may come to class with their own laptops, as well as use lab computers, besides in some circumstances they take a course distantly. So, hardware configuration, host operating system, characteristics of network connections and other conditions vary widely. Besides, students must get feedback on their practical assignments as soon as possible, although this task could be challenging for small teaching staff working with a large number of students during the semester. Some degree of automation on checking and grading of assignments would decrease the amount of routine work and give teaching

staff the ability to pay attention to more creative assignments. In light of the above, Linux online virtual environment running from a web browser could make practical assignments more independent of software and hardware. It also could help students to get well-timed feedback on simple mistakes.

There are numerous Linux online virtual environments with different characteristics and different underlying technologies. Most of them could be at least partially used in teaching operating systems. Conversely, although we generally appreciate the idea of combining multiple virtualization tools for one course, there is still the need for a basic tool to be applied most of the time.

The purpose of the article is to investigate available ready-made Linux online environments based on different virtualization technologies and, thus, elaborate the requirements to an individually designed system which provides Linux online virtual environments.

2 Related work

Available works on development and usage of online Linux virtual environments vary greatly focusing on different approaches as well as different courses' needs. David Croft and Matthew England [1] describe the experience using Codio at Coventry University in teaching programming. Students use online virtual Linux environments provided by Codio, while teaching staff add supplemental tasks with automated feedback. David J. Malan [2] describes moving CS50, Harvard College's introductory computer science course, into the cloud provided by Amazon Elastic Compute Cloud (Amazon EC2).

Bobby Powers, John Vilk and Emery D. Berge in the work [3] present Browsix, posing it as a framework between Unix-like operating system and a web browser. Browsix is emulating core Unix features using JavaScript. Rémi Sharrock, Lawrence Angrave and Ella Hamonic [4] offer WebLinux tool providing a Linux terminal and an integrated development environment (IDE) in the browser. In this case, Linux is also emulated using JavaScript. Aristogiannis Garmpis and Nicolaos Gouvatsos [5] suggest an online environment including emulated Ubuntu Linux operating system with Graphical User Interface (GUI).

Nadiia Balyk, Yaroslav Vasylenko, Vasyl Oleksiuk and Galina Shmyger [6] describe a virtual cloud laboratory based on Apache CloudStack and EVE-NG Community and its use for teaching the CCNA Cyber Operations course. Andrii Ya. Batiuk, Dmytro Ye. Vankevych, Hryhorii H. Zlobin in [7] focus on an experience of using Proxmox VE along with OpenVZ virtual containers and KVM hypervisor for Linux System Administrating special course.

Earlier, the study on the experience of teaching OS at higher educational establishments of the world has been done [8], showing a number of common features of analysed OS courses, including the following.

- The high priority of studying Unix-like OSs (ranging from teaching OSs like PintOS to popular full-function Linux distributives like Ubuntu).

- The use of a wide variety of virtualization tools (from full emulators like QEMU to type II hypervisors like VMware Workstation, VMware Fusion or VirtualBox, as well as more specific tools like Vagrant, mCertiKOS or Cygwin).
- In many cases, the availability of at least one alternative virtualization tool along with the one applied by default.

All the above-mentioned studies have been analysed and taken into consideration while elaborating the requirements to an individually designed system which provides Linux online virtual environments.

3 Elaborating the requirements to Linux online virtual environment for teaching the operating systems course

To formulate the requirements to the system which would provide Linux online virtual environments we need to analyse the needs of the course, taking into account its possible future development.

3.1 Previous theoretical and practical research

Our previous work [9] develops the methodology for using Unix-like OS virtualization technologies in training bachelors of Informatics based on the varied approach to applying before-mentioned technologies.

The approach involves combining several virtualization tools for meeting individual features of a course, individual students' needs and ensuring hardware and software fault tolerance. The work specifies the procedure for selecting virtualization tools in teaching OS to bachelors of Informatics. Following the suggested methodology, the study also developed the *Operating Systems and System Programming* handbook for future bachelors of Informatics and methodological guidelines *Virtualization Technologies in Teaching Operating Systems to Bachelors of Informatics* for teachers of higher educational establishments. During the implementation of the methodology in Zhytomyr Ivan Franko State University (2016–2018) we have been using the following tools.

- Oracle VirtualBox
- Amazon EC2 t2.micro instances
- Cloud 9 IDE
- Laptops with Linux (Ubuntu distributive)

The situation with Cloud 9 IDE is described in subsection 3.3. Amazon EC2 instances were used as a supportive tool mainly because of low network bandwidth causing considerable latency.

3.2 Reflection on recent practical experience

Later experience involved teaching two courses in operating systems in Zhytomyr Polytechnic State University (from September 2019 till now).

The first course, Unix/Linux Operating Systems, was held from September 2019 to December 2019 and was attended by third-year students of Computer Engineering and Cybersecurity specializations.

The second course, Operating Systems, started in February 2020 and was attended by second-year students of Computer Engineering, Cybersecurity, Software Engineering specializations.

Throughout the mentioned courses, the students and the teaching staff were using multiple tools for practical assignments. These tools are listed below and include both locally installed applications and remotely accessed environments.

- Virtual machine from NDG Linux Essentials course [10] (available on CISCO Networking Academy)
- Oracle VirtualBox
- Laptops with Linux (namely Ubuntu and Arch distributives)
- Amazon EC2 t2.micro instances (intermittently)

Experience of teaching these two courses revealed the need for a virtual environment which would provide access to fully functioning Linux installation, supporting most of the course practical assignments.

Concerning Unix/Linux Operating Systems course, virtual machine from NDG Linux Essentials met almost all the requirements of the course, except networking (and, thus, manipulating software packages) and the impossibility to save students' files between reboots.

According to the needs of our current courses in operating systems, the list of major features an online environment should have is as follows.

- The support of most administrative and non-administrative commands the typical Linux installation would have.
- Giving administrative privileges to students inside the virtual environment.
- The support of basic networking operations (changing network configuration, *ping* etc.).
- The support of uploading/downloading files to/from virtual environments.
- The ability to keep guest OS up-to-date.
- The support of bash scripting.
- The ability to save students' files in guest OS between reboots.

3.3 An overview of available ready-made Linux online virtual environments

Online tools for working with Unix/Linux operating systems through the web browser or other networking clients vary greatly depending on the functionality they provide, conditions they are available on, underlying technologies et al. All the Unix/Linux online virtual environments could be divided into the following groups.

– Standalone online Unix/Linux terminals.
– Online IDEs which include Unix/Linux terminals.
– Full-function Unix/Linux virtual environments in the cloud.

Standalone online Unix/Linux terminals. These terminals are often free of charge. They also are not necessarily based on a real OS kernel. A considerable part of standalone online terminals emulates the effect of some commands through JavaScript. However, other online Unix/Linux terminals are based on full emulators (e. g. QEMU), and thus give the user more realistic experience. *Examples:* JSLinux [11], Copy.sh Virtual x86 [12], Webminal [13], Linuxzoo [14] (based on full emulation), JS/UIX [15], Weblinux [16], Browsix [17], CB.VU [18] (simplified emulation).

The Table 1 gives a comparison of different standalone online Unix/Linux terminals.

The main advantage of standalone online Unix/Linux terminals is their easy accessibility. Among the disadvantages of these terminals are a limited set of commands available, obsolescence, limitations of root access and/or of network access. Furthermore, since most of standalone online Unix/Linux terminals appear to be projects developed and supported by few enthusiasts, there are some doubts about the future of these projects.

Online integrated development environments (IDEs) which include Unix/Linux terminals. These tools' main function is writing code. However, some online IDEs also include Linux terminal with programming commands installed. Some other online IDEs do not provide a terminal, but allow writing and running bash scripts. Root access and networking from the virtual environment are usually provided in paid online IDE services.

Examples: Codio [19], Cloud 9 [20], Codeanywhere [21], Paiza.io [22] etc. Among the advantages of online IDEs is multifunctionality and usually more recent Linux releases than in standalone online terminals. The disadvantages of these environments include a stronger focus on programming than on administrating causing that some components are not available.

Terms of use of free online IDEs may be changed. That was the case with Cloud9 IDE. Originally the service offered a virtual machine with Ubuntu Linux, full-function terminal and an IDE for collaborative programming for all the registered users. Later, due to numerous security violations from these virtual machines, the provider changed the terms. According to the new terms, users have to connect virtual machines from other services.

Full-function Unix/Linux virtual environments in the cloud. These environments are virtual machines or virtual containers in the cloud, usually with administrative rights and network available. Commonly such services conform to the IaaS model (Infrastructure as a Service). Some of the IaaS services offer a long free trial period and special programs for educational institutions. Examples: Amazon EC2 [23], Google Cloud Platform [24], Microsoft Azure [25]. The main advantage of the full-function Unix/Linux virtual environments in the cloud is access to most of OS commands,

including administrative and networking. The main disadvantage is that regular usage is non-free.

Table 1. Comparison of standalone online Unix/Linux terminals

Service name and developers	Virtualization / Guest OS	Options
JSLinux, Fabrice Bellard	Full emulator (compiled to JavaScript) / Linux 4.12.0 (Buildroot), Linux 4.15.0 (Buildroot, Fedora 29)	*Progress saving:* Yes, through VFsync *Networking:* Yes, bandwidth up to 40 kB/s, up to two connections per public IP address *Root privileges:* Yes *GUI:* Yes
CB.VU, Colin Barschel	Very simplified emulator (JavaScript) / None, emulates few Unix commands	*Progress saving:* No *Networking:* No, emulates few networking commands (e. g. *ping*) *Root privileges:* No *GUI:* No
JS/UIX, mass:werk, Norbert Landsteiner	Simplified emulator (JavaScript) / None, emulates limited number of Unix commands	*Progress saving:* Partially yes (demo version) *Networking:* No *Root privileges:* Yes *GUI:* No
Copy.sh Virtual x86, Fabian Hemmer et al.	Full emulator (compiled to JavaScript) / ReactOS, FreeBSD 10.2, Oberon 2.3.6, Arch Linux, KolibriOS, Linux 2.6, Linux 3.18, OpenBSD, Solar OS; supports OS images, uploaded by a user	*Progress saving:* No *Networking:* Yes, in some cases (e. g. Linux 3.18) *Root privileges:* Yes *GUI:* Yes, in some cases (e. g. KolibriOS)
WebLinux, Rémi Sharrock	Simplified emulator (JavaScript) / None, emulates limited number of Unix commands	*Progress saving:* No *Networking:* No *Root privileges:* Yes *GUI:* No
Browsix, Bobby Powers, John Vilk, Emery D. Berger	Emulation of Unix abstractions through browser API (JavaScript)	*Progress saving:* No *Networking:* No *Root privileges:* No *GUI:* No
Webminal	Full emulator / Ubuntu Linux	*Progress saving:* Yes *Networking:* Yes, with restrictions *Root privileges:* Yes, in paid accounts *GUI:* Yes, in paid accounts
Linuxzoo, Gordon Russel	Full emulator (QEMU) / CentOS 7	*Progress saving:* No *Networking:* No *Root privileges:* Yes *GUI:* Yes, through Java VNC

3.4 The requirements for Linux online virtual environments for teaching the operating systems course

The analysis of free ready-made Linux online virtual environments currently available along with the analysis of an experience of teaching courses in operating systems showed the need for an individually designed system to provide Linux online virtual environments. The environments should meet the following requirements.

- **Realism.** Linux, virtualized through the environments, should behave the same way the real system would behave most of the time (throughout most of the practical assignments).
- **Relevancy.** The system should meet the requirements of the course.
- **Availability.** Virtual environments should be available to all the students attending the course after they had successfully proved their affiliation with the university.
- **Stability.** Failures need to be addressed properly and within a predictable time.
- **Scalability.** The system should be able to evolve for enabling new features or supporting larger hardware capacity. New features we are particularly interested in is the automation of simple assignments checking and grading (e. g. the correctness of commands used).
- **Security.** Security threats should be minimized. The effective mechanisms of dealing with potential security breaches need to be developed.

However, the listed requirements pose serious challenges that need to be overcome.

The challenge of a wide choice. There are different virtualization technologies and platforms that could be applied to complete the job. Analysing them and selection of an appropriate among a great variety of options is a resource demanding task.

The challenge of implementing. Design, development and support of the target system is another resource-demanding task.

The challenge of network isolation. The mentioned above problem of Internet access from inside the online Linux virtual environment is typical for all virtualized systems which are publicly available. Granting root/sudo access to students may produce serious security challenges. The problem is greatly exacerbated if network access is also provided.

The popular solution lies in making virtual environments isolated from the internet. Students may reach virtual terminals via the internet to input commands (e. g. through the web browser), but the internet access from within the virtual environment itself is forbidden. Such an approach is entirely appropriate for teaching the vast majority of Operating Systems course units, while it is inadequate for practising on network configuration or installing software from network repositories.

Another possible solution to the network access problem is getting real personal data of the online user, who works with virtualized OS remotely. Notably, cloud service providers, giving virtual machines in the cloud with both root access and networking available (Amazon Web Services, Google Cloud Platform, Microsoft Azure etc.),

affiliate user account with a valid credit card number. This does not guarantee the legitimate use of an account, but facilitate investigation of possible security violations.

The third solution lies in giving students limited network access within the online virtual environment (enough for testing network configuration). A local repository for learning commands like *apt* may also be organized.

The challenge of a "playground". Conversely, another question arises: will the always-on system specially tailored for the learning needs be able to give students an experience of dealing with some problems real administrators may face? We find it crucial to find an optimal balance between stability and realism. Will such a "playground" environment be enough to motivate students to continue learning Linux after they finish the course? Given the foregoing, some assignments may be still done using less predictable tools like VirtualBox-based virtual machines. This affects some requirements for online Linux virtual environment. Namely, if certain assignments will be done outside the environment, then:

- a command-line interface may be enough for the online environment (even if the course include assignments on the Linux GUI);
- an implementation plan may start from isolated environments without the ability to reach the network from inside out, as long as these environments may be easier to manage on this stage.

The challenge of a transition. Besides the expenditure of resources on design, development and implementation of the system, the latter will imply a phase of transition. The transition from using tools described in subsection 3.2 to using online Linux virtual environments could be smoothed by applying additional tools, which need to be prepared in advance according to the varied approach mentioned earlier. These additional tools may include some previously used tools (e. g. locally installed hypervisors) as well as online Linux terminals (e. g. virtual machines from NDG Linux Essentials course, Webminal etc).

4 Conclusions

The work has made a survey on Linux online environments based on various virtualization technologies with the objective to elaborate the requirements to an individually designed system which provides such environments.

Linux ready-made online virtual environments are organized into three major groups: standalone online Unix/Linux terminals; online integrated development environments (IDEs) which include Unix/Linux terminals; full-function Unix/Linux virtual environments in the cloud. The main advantages and disadvantages of each group, as well as a comparison of standalone online Unix/Linux terminals, is done.

We also analyzed our previous and current teaching experience to find out online tools that proved efficient and the reasons why they still do not meet part of the course needs.

The study concluded the need for designing and implementing a virtual Linux environment with remote access to meet individual requirements of the Operating Systems course. The virtual Linux environment should be realistic (supporting most Linux commands through real execution, not imitation); relevant (meeting requirements of the course); available (accessible through the internet, for everyone who can prove his or her affiliation with the university); stable (failures should be addressed effectively); scalable (support increasing hardware capacity, be able to evolve); secured (security threats should be minimized, security breaches should be effectively handled). Another point of interest concerns the partly automation of practical assignments, including training and grading.

Future studies should focus on the investigation of virtualization technologies suitable for Linux online environment, including private open-source cloud platforms.

References

1. Croft, D., England, M.: Computing with Codio at Coventry University: Online virtual Linux boxes and automated formative feedback. In: Proceedings of the 3rd Conference on Computing Education Practice (CEP ’19), Article 16, pp. 1–4. ACM, New York (2019). doi:10.1145/3294016.3294018
2. Malan, D.J.: Moving CS50 into the Cloud. Paper presented at the 15th Annual Conference of the Northeast Region of the Consortium for Computing Sciences in Colleges, Hartford, Connecticut, April 2010
3. Powers, B., Vilk, J., Berger, E.D.: Browsix: Bridging the Gap Between Unix and the Browser. ACM SIGPLAN Notices 52(4), 253–266 (2017). doi:10.1145/3093336.3037727
4. Sharrock, R., Angrave, L., Hamonic, E.: WebLinux: a scalable in-browser and client-side Linux and IDE. In: L@S '18: Proceedings of the Fifth Annual ACM Conference on Learning at Scale, Article 45, pp. 1–2 (2018). doi:10.1145/3231644.3231703
5. Garmpis, A., Gouvatsos, N.: Innovative teaching methods in Operating Systems: The Linux case. In: Innovative approaches in Education: Design and Networking, Patras, Greece, pp. 155–163 (2012). doi:10.13140/2.1.1611.4888
6. Balyk, N., Vasylenko, Ya., Oleksiuk, V., Shmyger G.: Designing of Virtual Cloud Labs for the Learning Cisco CyberSecurity Operations Course. In: Ermolayev, V., Mallet, F., Yakovyna, V., Kharchenko, V., Kobets, V., Korniłowicz, A., Kravtsov, H., Nikitchenko, M., Semerikov, S., Spivakovsky, A. (eds.) Proceedings of the 15th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer (ICTERI, 2019), Kherson, Ukraine, June 12-15 2019, vol. II: Workshops. CEUR Workshop Proceedings **2393**, 960–967. http://ceur-ws.org/Vol-2393/paper_338.pdf (2019). Accessed 30 Jun 2019
7. Batiuk, A., Vankevych, D., Zlobin, G.: Vykorystannia tekhnolohii virtualizatsii v spetskursi “Systemne administruvannia OS LINUX” (Using virtualization technologies in courses “System administration OS Linux”). Electronics and information technologies 3, 220–225 (2013)
8. Holovnia, O.S.: Methodology for using Unix-like operating systems virtualization technologies in training bachelors of Informatics. Dissertation, Institute of Information Technologies and Learning Tools of NAES of Ukraine (2019)

9. Spirin, O.M., Holovnia, O.S.: Using Unix-like operating systems virtualization technologies in training the bachelors of Computer Science. Information Technologies and Learning Tools **65**(3), 201–222 (2018). doi:10.33407/itlt.v65i3.2055
10. Linux Essentials - Online Courses & Labs Training | NDG. https://www.netdevgroup.com/online/courses/open-source/linux-essentials (2020). Accessed 7 Jun 2020
11. Bellard, F.: JSLinux. https://bellard.org/jslinux (2019). Accessed 21 Mar 2020
12. copy's home. https://copy.sh/. Accessed 21 Mar 2020
13. Webminal - Learn and Practise Linux online, Programming online. https://www.webminal.org/ (2019). Accessed 21 Mar 2020
14. Russell, G.: Welcome to linuxzoo. https://linuxzoo.net/ (2020). Accessed 21 Mar 2020
15. Landsteiner, N.: JS/UIX – Terminal. https://www.masswerk.at/jsuix/ (2003). Accessed 21 Mar 2020
16. System Programming. https://remisharrock.fr/sysbuild/#/VM. Accessed 21 Mar 2020
17. Browsix: Unix in the browser tab. https://browsix.org/. Accessed 21 Mar 2020
18. CB.VU, http://cb.vu/ (2008). Accessed 21 Mar 2020
19. Codio Home. https://www.codio.com/ (2020). Accessed 21 Mar 2020
20. AWS Cloud9 Amazon Web Services. https://aws.amazon.com/cloud9 (2020). Accessed 21 Mar 2020
21. Codeanywhere. Cross Platform Cloud IDE. https://codeanywhere.com/ (2020). Accessed 21 Mar 2020
22. Online PHP/Java/C++... editor and compiler | paiza.IO. https://paiza.io/en (2014). Accessed 21 Mar 2020
23. Amazon EC2. https://aws.amazon.com/ec2 (2020). Accessed 7 Jun 2020
24. Compute Engine: Virtual Machines (VMs) | Google Cloud. https://cloud.google.com/compute (2020). Accessed 21 Mar 2020
25. Cloud Computing Services | Microsoft Azure. https://azure.microsoft.com (2020). Accessed 21 Mar 2020