

Using Active Learning to Teach Unix Commands

Karina Mochetti¹, Carlos Henrique D. C. Santos¹, João Pedro Abreu de Souza¹

¹ Computer Institute – Fluminense Federal University (UFF)
Niterói – RJ – Brazil

`kmochetti@ic.uff.br, carloshdcs@id.uff.br, jp_abreu@id.uff.br`

***Abstract.** Nowadays, it is vital for a computer scientist to be prepared for different environments such as new frameworks, different programming paradigms and multiple operational systems. Unfortunately, it is common that freshman students are only used to Microsoft Windows' Operating System and no proper class or activity is done to revert that. Therefore, our goal is to teach minimal Unix-like skills to first year students through a challenge-based learning approach. To this end, we will use a list of quests with ascendant difficulty levels, without no functionality to reinforce competition between students. Its interface is all web-based with a colored shell looking, so students can get used to Unix command line environment and slowly loose their distrust in a different operational system.*

1. Introduction

Nowadays, it is vital for a computer scientist to be prepared for different environments such as new frameworks, different programming paradigms and multiple operational systems (OS). Usually, people with low experience in the field carry a stigma that Unix-based systems are hard to use, specially when it comes to shell commands line interface that does not have buttons or the main use of mouse. Therefore, it is crucial that students loose their distrust when it comes to different OS's. Our university currently does not have any courses that teaches basic Unix although it is important to provide alternative ways for newcomers to learn the basics.

The method of using missions or achievements in the classroom is not new and can obtain good results in motivating and engaging students [Poffo et al. 2017, Santana et al. 2017]. Moreover, this method has been commonly studied to encourage elementary and high school to learn computational thinking and basic programming [Silva et al. 2017].

A few projects with the goal of teaching UNIX in a more active way were developed. One approach divides knowledge between students to encourage cooperation [Simón-Hurtado and Vivaracho-Pascual 2005] and other creates a treasure hunt using UNIX tools and bash commands to increase freshman's motivation [Moy 2011].

The goal of this project is to give students minimal knowledge of Unix-like shell. The tool environment is a colored bash command line, implemented without any graphic interface. In this way we can simulate the real environment encountered in Unix systems. We will use different colors to distinguish between shell messages and the tool messages. The challenges will be divided in levels, each one adding more commands and more combinations, such as pipes and redirects. An achievement system will also be used to increase student's motivation, but not encouraging a competitive environment.

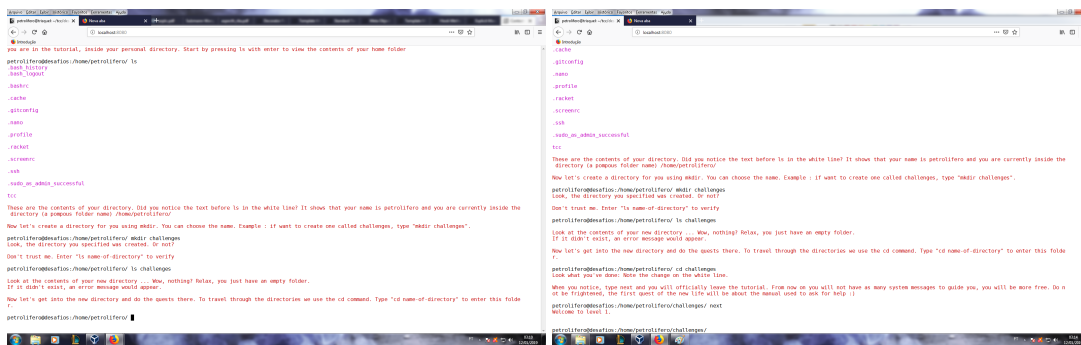


Figure 1. Screenshots of the tutorial initial phase.

2. Methodology

The epistemology of Piaget [Piaget 1970] states that knowledge is generated constructively, both from the subject itself and from her/his experiences. According to this theory, the construction of knowledge is active, i.e., the subject participates in her/his learning process. Active Learning Methods [Prince 2004] agrees with this cognitive theory, in which the student is active in her/his own learning. Differently to expository lectures, these approaches encourage students to interact with each other and with the teacher, allowing them to experience the content in different point of views.

With this in mind, we developed a tool that simulates a shell interface giving the student several challenges each one based on a different command line. The tool begins with a tutorial, as seen in Figure 1, teaching the basic of command line, such as the *cd* command for change directory. After the tutorial, students are given several challenges such as list all files in a folder or rename a file. Besides the tips and guides from the tool, the student may have other sources of research such as the internet, books or the help of colleagues.

After the tutorial, challenges start to be presented to the student. We may use as an example the beginner's quest related to the *rm* command. The student must remove a specific file only using this command. The system will block any attempt such as to remove other files or remove the correct one using other commands. Hence, the quest is only completed when the student has successfully removed the desired file using the proper command. Afterwards, a message in red will be displayed informing that the quest was completed. Any system message will still be presented in black. Every wrong attempt by the student will display a special message, with hints, guiding her/him to the right use of the command.

Each challenge represents an achievement and a colored scheme is used to terminate the student's level, i.e., the more challenges she/he solves, the greater the level she/he will be, with levels having different colors. In this way we intent to engage students without reinforcing competition.

3. Final Considerations

Our purpose with this project is to increase student's familiarity with more than one Operational System. Students usually have a distrust of Unix-like Operating Systems and we want to change this scenario, opening their minds to think about Operational Systems as

tools, each with its own advantages and disadvantages. We intent to conduct tests on the freshman of Computer Science of our university later next year. We also intent to monitor the system, generating logs of use, allowing an independent and personalized assistance to each student.

References

- Moy, M. (2011). Efficient and playful tools to teach unix to new students. In *Conference on Innovation and technology in computer science education*, pages 93–97. ACM.
- Piaget, J. (1970). *Science of Education and the Psychology of the Child*. Orion Press.
- Poffo, M., Thiry, M., Fernandes, A., and Heemann, C. (2017). Gamificação: Agente motivador na aprendizagem de engenharia de software. *Computer on the Beach*.
- Prince, M. (2004). Does active learning work? a review of the research. *J. Engr. Education*.
- Santana, A., Silva, T. R., and Aranha, E. (2017). Um modelo de tutor virtual para aulas baseadas em missões de programação de jogos digitais. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 1059.
- Silva, N. C., Lima, A. C., de Souza, N., and de Sousa, D. F. (2017). Raciocínio lógico nas escolas: Uma introdução ao ensino de algoritmos de programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 6, page 1011.
- Simón-Hurtado, M. A. and Vivaracho-Pascual, C. (2005). Learning unix in first year of computer engineering. *SIGCSE Bull*.