Systems Programming (CSCI 3000)

Shell Application Demos- Group Project

Professor: Ted Holmberg
Department of Computer Science
University: University of New Orleans

1 Project Description

In this project, students are required to form teams and select a popular shell application from the extensive list provided below. Each team will conduct a comprehensive analysis of their chosen application, including its features, usage, and potential benefits. The project will culminate in a presentation where teams demonstrate their application, discuss its functionalities, and field questions from the audience.

1.1 List of Shell Applications

The applications are categorized to assist in selection and provide a broad spectrum of utilities that are essential in systems programming and daily command-line usage.

1.1.1 Text Editors

- **Vim**: Highly configurable text editor built to enable efficient text editing.
- **Emacs**: An extensible, customizable, free/libre text editor — and more.
- **Nano**: Easy to use, lightweight command line text editor.

1.1.2 File Management

- **rsync**: Fast, versatile, remote (and local) file-copying tool.
- **tar**: Archive tool for creating and extracting .tar.gz files.
- **find**: Utility for searching files in a directory hierarchy.

1.1.3 Networking

- **ssh**: Secure Shell for safe remote login over unsecured networks.
- **curl**: Command line tool and library for transferring data with URLs.
- **wget**: Free utility for non-interactive download of files from the Web.

1.1.4 System Monitoring and Management

- **top**: Display Linux processes.
- **htop**: Interactive process viewer, a better alternative to 'top'.
- **tmux**: Terminal multiplexer, it enables a number of terminals to be accessed and controlled from a single terminal.

1.1.5 Version Control

- **git**: Free and open source distributed version control system.

1.1.6 Search Tools

- **grep**: Print lines matching a pattern.
- **awk**: Pattern scanning and processing language.
- **sed**: Stream editor for filtering and transforming text.

1.1.7 Networking and Security

- **nmap**: Network exploration tool and security / port scanner.
- **iptables**: User-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall.

1.1.8 Programming and Development

- **make**: Utility for directing compilation.
- **gcc**: GNU Compiler Collection, supports various programming languages.
- **docker**: Platform for developing, shipping, and running applications in containers.

1.1.9 Package Management

- **apt**: Advanced Package Tool, used on Debian and Ubuntu systems.

1.1.10 Data Transfer and Web Tools

- **scp**: Secure copy (remote file copy program).
- **rsync**: Utility for efficiently transferring and synchronizing files across computer systems.
- **curl**: Command line tool for transferring data with URL syntax.

1.1.11 Image and File Conversion

- **ImageMagick**: Software suite to create, edit, compose, or convert bitmap images.
- **ffmpeg**: A complete, cross-platform solution to record, convert and stream audio and video.

1.1.12 Miscellaneous

- **screen**: Full-screen window manager that multiplexes a physical terminal.
- **jq**: Lightweight and flexible command-line JSON processor.
- **tree**: Recursive directory listing command that produces a depth-indented listing of files.

This list is not exhaustive but provides a broad spectrum of the most popular and widely used shell applications, spanning various categories essential for systems programming, development, file management, and networking. Teams are encouraged to select an application that not only interests them but also presents a challenge and an opportunity for deep learning and analysis.

2 Objectives

1. To understand the practical usage of shell applications in systems programming.
2. To enhance research, collaboration, and presentation skills within a team environment.
3. To evaluate the significance of selected shell applications in solving real-world problems.

3 Project Requirements

Each team must:

- Select a shell application from the list provided by the instructor.
- Conduct a deep dive into the selected application, understanding its core functionalities, use cases, and impact.
- Prepare a 15-minute presentation showcasing a demo of the application, discussing its features, uses, and why it is beneficial.
- Submit a written report or presentation slides that includes an overview of the application, detailed analysis of its features, and a summary of the team's findings.
- Be prepared to answer questions following the presentation.

4 Presentation Date

The presentations will take place during the semester's lecture time. The exact date and time will be scheduled and announced.

5 Grading Rubric

5.1 Written Report / Presentation Slides (30%)

- Clarity and depth of analysis: 15%
- Organization and structure: 10%
- Grammar and spelling: 5%

5.2 Presentation (40%)

- Content accuracy and completeness: 20%
- Demonstration of the application: 10%
- Ability to engage the audience and answer questions: 10%

5.3 Team Collaboration and Participation (30%)

- Contributions to the project: 15%
- Peer evaluation: 15%

6 Submission Guidelines

The final report must be submitted in PDF format or PPT via email.

7 Conclusion

This project offers a unique opportunity to explore the practical applications of shell programming within a collaborative environment. By delving into the functionalities and uses of shell applications, students will gain valuable insights into their importance in the field of systems programming.