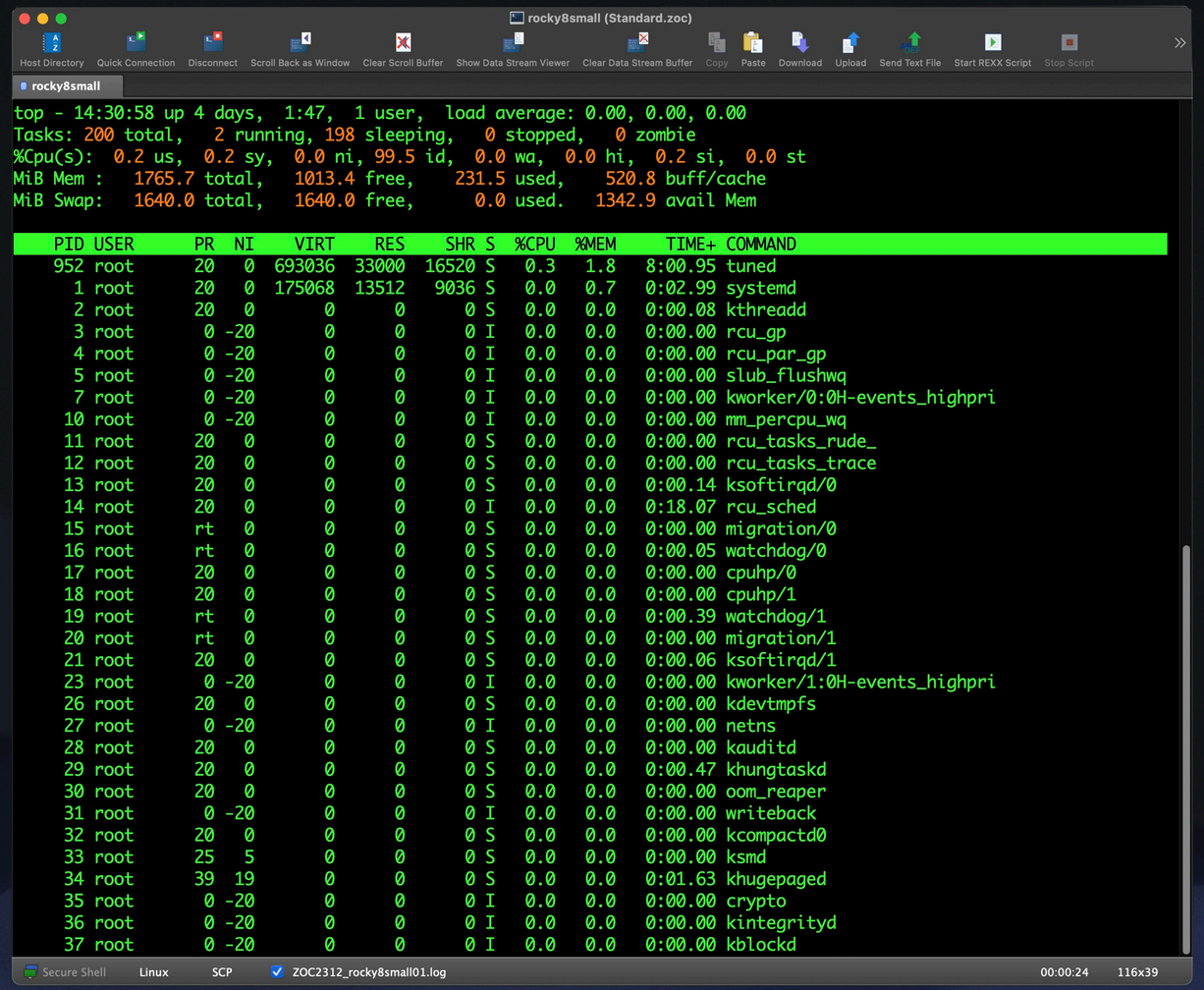

EXPLORING LINUX SHELL HISTORY AND HISTCONTROL
YOUTUBE.COM/@INFORMATIONTECHWITHDAVE

The Primary UI: The Shell

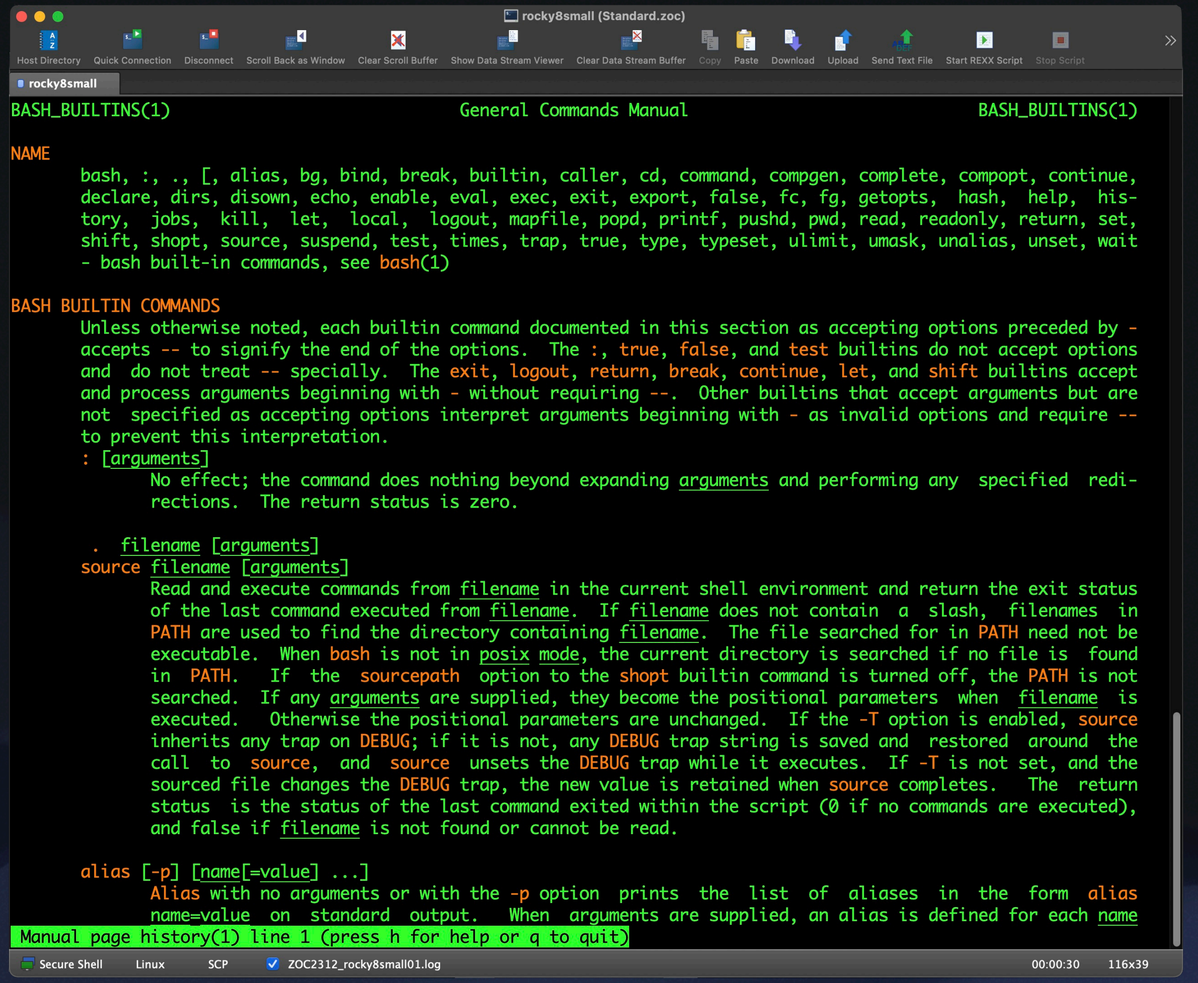
- On Linux the primary user interface is the shell, most commonly bash.
- Via the shell users can navigate the filesystem, manipulate files, and execute programs.
- A very useful feature of the shell is its history, which retains a list of commands the user ran, even between login sessions and surviving reboots.
- Understanding shell history is crucial for efficiently using the shell.
- The history is valuable for troubleshooting and investigating security breeches.

YouTube.com/@InformationTechWithDave



Evolution of Shell History

- The shell history feature dates back to the early days of UNIX operating systems.
- Originally, it was found in ~/.history.
- As bash and other more advanced shells became popular that facilitated users navigating through, searching, and manipulating the history.
- Nowadays, it's most commonly found in ~/.bash_history.
- Documentation for the history command is found in the bash man page.



YouTube.com/@InformationTechWithDave

HISTCONTROL Values

- The HISTCONTROL variable enhances control and customization of the bash history. Common values include:
  - ignorespace: Commands starting with a space character are not saved to the history. This is useful for avoiding the storage of sensitive information like passwords. It can also be used by bad actors to cover their tracks!
  - ignoredups: This prevents duplicate commands from being stored in the history.
  - ignoreboth: This is a combination of ignorespace and ignoredups. It ignores commands that start with a space and eliminates duplicates.
  - erasedups: Consecutive duplicate commands are removed from the history.
  - ignoreerr: Commands that fail (exit with a non-zero status) are not saved.

Configuring HISTCONTROL

- To set the HISTCONTROL variable use the export command, like so:
  - $ export HISTCONTROL=ignoreboth
- That will ignore commands starting with a space and duplicate commands.
- To have the change persist across login sessions and reboots, add this line to either your ~/.bashrc or ~/.bash_profile using a text editor.

Navigating History

- To display the history, do:
  - $ history
- To recall previous items use the up and down arrows.
- Other options:
  - $!n: Re-executes the command with the specified number 'n' from the history.
  - $!!: Re-executes the last command.
  - $!string: Re-executes the most recent command that starts with 'string'.
  - $ Ctrl+R: Initiates a reverse search through the history, allowing you to search for and execute a command that matches the entered string.
- By combining these features, users can make their command-line sessions more productive and secure.

YouTube.com/@InformationTechWithDave

Conclusion

- Understanding the history mechanism and utilizing features like HISTCONTROL in the Linux shell can significantly enhance productivity and streamline command-line workflows.
- Both experienced Linux users and n00bs will find that mastering these tools will make your interactions with the command line more efficient and enjoyable.
- Experiment with different settings, explore history navigation techniques, and tailor your shell experience to suit your preferences and requirements.

Thank you for watching!

If you liked this video please hit the like button and subscribe. Subscribe

Don’t forget to click the notification button. Subscribed

Please share with anyone who would find this helpful. Share

@InformationTechWithDave