Linux Lab Report – Kate Sofia Petersen

Email: kontakt@katepetersen.se

Part 1 – System and User Information **Objective:** Display system and user information including kernel version, current user, and date/time. --- ### Comment: These commands provide essential system and user identity information useful for system administration. --- ### Commands * `uname -a` → shows system information including kernel version. * `whoami` → shows the current user. * `date` → shows the current date and time. --- ## Part 2 – Commands with Flags and Arguments **Objective:** Use commands with flags and arguments to customize output. --- ### Comment: Flags and arguments modify command behavior and output. These examples demonstrate listing files with details and searching text ignoring case. --- ### Commands * `ls -l /home` → lists files with details in /home. * `grep -i "test" file.txt` → searches for "test" in file.txt, ignoring case. --- ## Part 3 – Navigating Between Directories **Objective:** Move confidently between directories using relative and absolute paths. --- ### Comment: Navigating directories is essential for file management. These commands show how to display the current directory and move to others. --- ### Commands * `pwd` → shows the current directory. * `cd /etc` → changes to the /etc directory. * `cd ~` → returns to the home directory. --- ## Part 4 – Creating and Removing Directories **Objective:** Manage directories by creating and removing them safely. --- ### Comment: Creating and removing directories helps organize files. Use mkdir to create and rmdir to remove empty directories. --- ### Commands * `mkdir dir1 dir2` → creates two directories. * `rmdir dir1` → removes an empty directory. --- ## Part 5 – Listing Files with Detailed Information **Objective:** View detailed file information including permissions and timestamps. --- ### Comment: Detailed listings provide insights into file permissions, ownership, size, and modification times. --- ### Commands * `ls -l` → shows file permissions, owner, size, and timestamps. --- ## Part 6 – Viewing and Changing File Permissions **Objective:** Understand and modify file permissions and ownership. --- ### Comment: File permissions control access. Use ls -l to view and chmod to change permissions. --- ### Commands * `ls -l file.txt` → shows permissions and ownership. * `chmod 644 file.txt` → changes permissions (owner: read/write, others: read). --- ## Part 7 – File Management **Objective:** Create, copy, move, and delete files safely. --- ### Comment: Managing files involves creating, copying, moving, and deleting. These commands demonstrate each operation. --- ### Commands * `touch file1.txt` → creates a file. * `cp file1.txt copy.txt` → copies a file. * `mv copy.txt newfile.txt` → moves/renames a file. * `rm newfile.txt` → deletes a file. * `ls` → verifies with directory listing. --- ## Part 9 – System Administration **Objective:** Perform administrative tasks with elevated privileges. --- ### Comment: System administration requires elevated privileges. Use sudo for admin commands and su to switch users. --- ### Commands * `sudo apt update` → runs command as administrator. * `sudo shutdown -h now` → shuts down the system. * `su - other_user` → switches user. --- ## Part 11 – Export Flow and Logging **Objective:** Manage logging environment variables and write to log files. --- ### Comment: Logging is essential for tracking system events. Set environment variables and append messages to log files. --- ### Commands * `export LOGFILE=log.txt` → creates environment variable for log file. * `echo "Start logging" >> $LOGFILE` → writes to log file. --- ## Part 12 – Disk Space and Memory Usage **Objective:** Monitor disk space and memory usage effectively. --- ### Comment: Disk and memory monitoring help maintain system health. Use df and free commands for readable output. --- ### Commands * `df -h` → shows disk space in human-readable format. * `free -h` → shows memory usage. --- ## Part 13 – User Groups and Permissions (Optional) **Objective:** View user groups and permissions. --- ### Comment: Understanding user groups helps manage permissions. --- ### Commands * `groups` → shows groups for current user. * `id` → shows UID, GID, and groups. --- ## Part 14 – Scheduled Jobs with cron (Optional) **Objective:** Manage scheduled jobs using cron. --- ### Comment: Cron schedules repetitive tasks. Verify cron service status and list jobs. --- ### Commands * `crontab -l` → lists scheduled jobs. * `systemctl status cron` → verifies cron is running. --- ## Part 15 – Environment Variables and System Settings **Objective:** View and manage environment variables. --- ### Comment: Environment variables store system settings. Use printenv to view them. --- ### Commands * `printenv` → shows all environment variables. * `printenv PATH` → shows specific variable. --- ## Part 16 – Network Ports and Services (Optional) **Objective:** Monitor network ports and services. --- ### Comment: Network monitoring helps secure services. Use netstat or ss to view open ports. --- ### Commands * `netstat -tuln` → shows open ports and services. * `ss -tuln` → alternative to netstat. --- ## Part 17 – System Logs with tail and Filtering **Objective:** View and filter system logs effectively. --- ### Comment: System logs provide insights into system events. Use tail and grep to filter logs. --- ### Commands * `tail /var/log/syslog` → shows latest lines in system log. * `grep "error" /var/log/syslog` → filters for "error" in logs. --- ## Part 18 – Directory Structure with find **Objective:** Search files and directories efficiently. --- ### Comment: Find helps locate files and directories recursively. --- ### Commands * `find . -type f` → shows all files in current and subdirectories. * `find . -type d` → shows all directories. --- ## Part 19 – User History and Command Logs (Optional) **Objective:** Review user command history. --- ### Comment: Command history helps track past commands. --- ### Commands * `history` → shows previous commands. * `cat ~/.bash_history` → shows history file. --- ## Part 20 – Scheduled Jobs with crontab **Objective:** Edit and manage scheduled jobs. --- ### Comment: Crontab allows editing scheduled jobs. Example shows daily backup at 5 AM. --- ### Commands * `crontab -e` → edit scheduled jobs. * Example: ``` 0 5 * * *

/home/sofia/backup.sh ``` Runs backup.sh daily at 05:00. ---

Mentor Feedback

Mentor, please provide your comments or feedback below: