**UNIX Shell Programming and User Knowledge: Exam Topics 2024/25**

**1. Exam Format**

The exam consists of two parts:

* **Written Test**: 15-20 theoretical or practical questions.
* **Practical Exam**: Solve one task in 60 minutes using a computer (offline).

**2. Core Topics**

Below are detailed examples and explanations for the exam-relevant topics.

**1. UNIX Standardization (POSIX)**

* **Explanation**: POSIX defines a standard operating system interface and environment to ensure compatibility across UNIX systems.
* **Example**: APIs and shell command behaviors standardized by POSIX help in cross-platform operability.

**2. User and Group Management**

* **Commands**: adduser, passwd, groups
* **Examples**:
* sudo adduser john # Add a new user
* groups john # List groups of a user
* **Explanation**: Manage users and groups for system access control, ensuring secure and organized user administration.

**3. File System and Path Concepts**

* **Commands**: ls, pwd, cd, mkdir
* **Examples**:
* mkdir /tmp/my\_folder # Create a temporary directory
* cd /tmp/my\_folder # Change directory
* pwd # Print the working directory
* **Explanation**: Navigating and managing file system structures effectively.

**4. File Permissions**

* **Commands**: chmod, chown
* **Examples**:
* chmod 644 file.txt # Read/write for owner, read-only for others
* chown user1:group1 file.txt # Change file ownership
* **Explanation**: Ensures data security and access control for files and directories.

**5. Shell Patterns**

* **Patterns**: \*, ?, [ ]
* **Examples**:
* ls \*.txt # List all text files
* ls file?.txt # Match filenames like file1.txt, file2.txt
* **Explanation**: Simplify searches and file manipulations using wildcards.

**6. Redirections and Pipes**

* **Operators**: >, <, |
* **Examples**:
* echo "Hello World" > file.txt # Write output to a file
* cat file.txt | grep "Hello" # Filter file content
* **Explanation**: Efficiently handle input/output streams and command chaining.

**7. Processes and Signals**

* **Commands**: ps, kill, trap
* **Examples**:
* ps -e # List all processes
* kill -9 1234 # Terminate a process
* trap "echo Interrupted" INT # Handle interrupt signals
* **Explanation**: Monitor system processes and manage signals to ensure smooth execution.

**8. Scripting Structures**

* **Structures**: if, for, while, case
* **Examples**:
* if [ -f file.txt ]; then echo "File exists"; fi
* for i in 1 2 3; do echo $i; done
* **Explanation**: Automate repetitive tasks and implement logic-based operations.

**9. Advanced Commands**

* **Commands**: awk, sed, find
* **Examples**:
* awk '{print $1}' file.txt # Print the first column of a file
* sed 's/old/new/g' file.txt # Replace text in a file
* find / -name "\*.log" # Find log files
* **Explanation**: Streamline text processing and locate files quickly.

**10. Environmental Variables**

* **Variables**: PATH, LANG, LC\_TIME
* **Examples**:
* echo $PATH # Display the system PATH
* export LANG=en\_US.UTF-8 # Set the language environment
* **Explanation**: Configure system-wide or user-specific settings for optimal shell operation.

**11. Regular Expressions**

* **Commands**: grep, egrep
* **Examples**:
* grep "^A" file.txt # Find lines starting with 'A'
* egrep "[0-9]{3}-[0-9]{2}" file.txt # Match specific patterns
* **Explanation**: Validate and extract specific data patterns efficiently.

**12. Process and Time Management**

* **Commands**: at, crontab, date
* **Examples**:
* date +"%Y-%m-%d" # Display the current date
* crontab -e # Edit scheduled tasks
* **Explanation**: Schedule and manage timed tasks effectively for automation.

This document consolidates the exam topics into examples and explanations to aid preparation for both written and practical tests. Comprehensive coverage ensures readiness for all potential exam queries. Familiarity with manual pages (man, help, info) is highly recommended.