**Detailed Examples and Explanations for UNIX Commands and Shell Scripting**

**1. File Management Commands**

* **Commands**: mkdir, rmdir, ls, cp, rm, mv, tree
* **Examples**:
* mkdir test\_directory # Create a new directory
* rmdir test\_directory # Remove an empty directory
* ls -la # List files with detailed information
* cp file1.txt test\_directory/ # Copy a file into a directory
* rm file1.txt # Delete a file
* mv file1.txt renamed\_file.txt # Rename a file
* tree # Display directory structure as a tree
* **Explanation**: These commands help in creating, renaming, copying, and managing files and directories.

**2. File and Directory Permissions**

* **Commands**: chmod, chown, chgrp
* **Examples**:
* chmod 755 script.sh # Owner can read, write, execute; others read, execute
* chown user1 script.sh # Change ownership of the file
* chgrp group1 script.sh # Change the group ownership
* **Explanation**: Manage access rights to secure files from unauthorized use.

**3. UNIX Links**

* **Commands**: ln
* **Examples**:
* ln file.txt hardlink\_file.txt # Create a hard link
* ln -s file.txt symlink\_file.txt # Create a symbolic link
* **Explanation**: Links allow multiple references to a single file, saving storage and improving accessibility.

**4. Timestamps**

* **Commands**: ls, touch
* **Examples**:
* ls -lt # List files sorted by modification time
* touch -t 202401011200 file.txt # Set timestamp to Jan 1, 2024, 12:00
* **Explanation**: Timestamps help in tracking file changes.

**5. Aliases**

* **Command**: alias
* **Example**:
* alias ll='ls -la' # Create an alias for detailed listing
* **Explanation**: Simplifies repetitive commands.

**6. Redirections**

* **Operators**: >, <, |
* **Examples**:
* ls > output.txt # Save output to a file
* cat < input.txt # Read from a file
* ls | grep "test" # Pipe output to another command
* **Explanation**: Manage input and output streams effectively.

**7. Text Filters**

* **Commands**: cat, cut, sort, uniq, etc.
* **Examples**:
* cat file.txt # Display file content
* cut -d',' -f1 file.csv # Extract first field of a CSV
* sort file.txt # Sort file content
* uniq -c file.txt # Count unique lines
* **Explanation**: Process and manipulate text data efficiently.

**8. Special Variables**

* **Variables**: $1, $2, $#, $@
* **Example**:
* echo "First argument: $1, Total: $#" # Access script arguments
* **Explanation**: Dynamically handle arguments in scripts.

**9. Conditional Structures**

* **Structures**: if, for, while
* **Examples**:
* if [ -f file.txt ]; then echo "Exists"; fi
* for file in \*.txt; do echo "$file"; done
* **Explanation**: Automate decision-making and repetition in scripts.

**10. Reading Input**

* **Command**: read
* **Examples**:
* read name # Wait for user input
* echo "Hello, $name"
* **Explanation**: Accept user input interactively or from files.

**11. Arithmetic in Shell**

* **Command**: $(( ))
* **Examples**:
* sum=$((3 + 5))
* echo "Sum: $sum"
* **Explanation**: Perform mathematical calculations directly in shell.

**12. Case Statement**

* **Command**: case
* **Examples**:
* case $1 in
* start) echo "Starting...";;
* stop) echo "Stopping...";;
* esac
* **Explanation**: Efficiently handle multiple conditions.

**13. Menu with select**

* **Command**: select
* **Examples**:
* select opt in Start Stop Exit; do
* echo "You chose $opt"; break;
* done
* **Explanation**: Create interactive menus in scripts.

**14. Regular Expressions**

* **Commands**: grep, egrep
* **Examples**:
* grep "pattern" file.txt # Search for a pattern
* egrep "^[A-Z]" file.txt # Lines starting with uppercase letters
* **Explanation**: Match patterns for text processing.

**15. Advanced Commands: sed and awk**

* **Commands**: sed, awk
* **Examples**:
* sed 's/old/new/g' file.txt # Replace text
* awk '{print $1}' file.txt # Extract the first column
* **Explanation**: Advanced text manipulation.

**16. Processes and Signals**

* **Commands**: ps, kill, trap
* **Examples**:
* ps -e # List all processes
* kill -9 1234 # Terminate process with PID 1234
* trap 'echo "Interrupt received"' INT
* **Explanation**: Manage running processes and handle signals.

**17. File Search (find)**

* **Command**: find
* **Examples**:
* find /path -name "\*.txt" -size +1M # Find large text files
* **Explanation**: Locate files based on criteria.