# Advanced Notes on Shell Scripting

## Introduction to Shell Scripting

* **Shell Scripting**: Writing and executing a series of commands in a plain text file for sequential processing by the computer.
* **Purpose**: Automates repetitive tasks (e.g., shutdown -h now), optimizing workflow and enhancing efficiency.
* **Interpreter**: Typically uses **Bash** to interpret and execute commands in the specified order.

## Fundamental Structure of a Shell Script

1. **Shebang (**``**)**
   * Initial line of the script specifying the interpreter.
   * Example:
   * #!/bin/bash
2. **Executable Commands**
   * Any command valid in a terminal can be included in a script.
   * Example:
   * echo "Hello World"
3. **Variables**
   * Used for storing and reusing values within the script.
   * Example:
   * name="Alex"  
     echo "Hi $name"
4. **Comments**
   * Lines beginning with # are ignored by the interpreter.
   * Essential for improving readability and future maintenance of code.
5. **Permissions**
   * Necessary to assign executable permissions for the script.
   * Command to set permissions:
   * chmod +x scriptname.sh

## Automating Tasks with Cron and Crontab

* **Cron**: Time-based job scheduling service in Unix-like operating systems.
* **Crontab File**: Defines the schedule for executing commands or scripts.
* **Editing Cron Jobs**:
  + Command:
  + sudo crontab -e
* **Scheduling Entry Example**:
* 0 22 \* \* \* /path/to/script.sh
  + Executes the specified script daily at **10:00 PM**.
  + Asterisks represent minute, hour, day, month, and weekday fields.
* **Utility**: Automates routine processes, such as scheduled shutdowns.

## Case Study: Automated Shutdown Script

* Example script for shutting down a Kali Linux laptop at **10:00 PM** daily.
* Features:
  + Broadcasts a warning message to all logged-in users.
  + Implements a five-minute delay before shutdown.
  + Logs the shutdown event.
  + Initiates the shutdown process.

### Example Script (shutdown\_at\_10pm.sh):

#!/bin/bash  
  
# Notify all users  
wall "⚠️ System will shut down in 5 minutes. Please save your work."  
  
# Wait for 5 minutes  
sleep 300  
  
# Log the shutdown event  
echo "Shutdown triggered at $(date)" >> /var/log/shutdown\_log.txt  
  
# Execute shutdown  
shutdown -h now

## Implementation Procedure

1. **Script Creation**
   * Open the script for editing using a text editor (e.g., nano):

* sudo nano /usr/local/bin/shutdown\_at\_10pm.sh

1. **Assign Executable Permissions**
   * Ensure the script is executable using the chmod command:

* chmod +x /usr/local/bin/shutdown\_at\_10pm.sh

1. **Schedule the Script with Cron**
   * Add the script to the crontab for daily execution at 10:00 PM:

* sudo crontab -e
  + Add the following line:
* 0 22 \* \* \* /usr/local/bin/shutdown\_at\_10pm.sh

## Conclusion

* Shell scripting provides a robust method for automating tasks and optimizing workflows.
* Understanding the fundamental components, including **shebang**, **variables**, **comments**, and **permissions**, is essential for effective script creation and execution.
* Tools like **Cron** enable automated scheduling, enhancing system management efficiency.