



**People’s Democratic Republic of Algeria**

**Ministry of Higher Education and Scientific Research**

**University of Algiers 1 Benyoucef BENKHEDDA**

**Faculty of Science**

**Department of Computer Science**

**Specialization: Network and Embedded Systems**

**1st year master**

(Teacher: Dr. **Abbas**)

**Raport**

**Theme**

|  |
| --- |
| Automated Backup and Restoration with Bash and C |

**By:**

Hicham SAIDI

BOUAICHA Rafik

2024/2025

# Used Libraries:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "backup.h"

#include "restore.h"

#include "option.h"

#include "basher.h"

## Basher Library:

Used by backup and restore libraries to simplify bash execution.

* **Commend executer:**

Take a bash command and execute it if it was structured right, and return an error message if not.

void execute\_command(const char\* command);

**x**

* **Bash launcher:**

Give to backup.sh the permission to be executed by running this command “chmod +x ./backup.sh”, like that we can use it in the command line and give it some parameters that are explain in the bash section of this report.

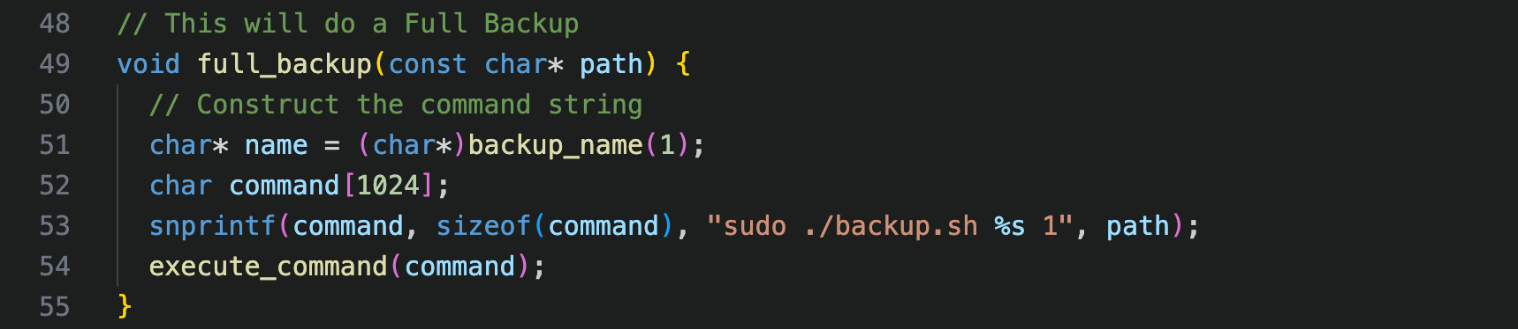
void bash\_lancher(const char\* command);

## Backup Library:

Contain all the backup functions:

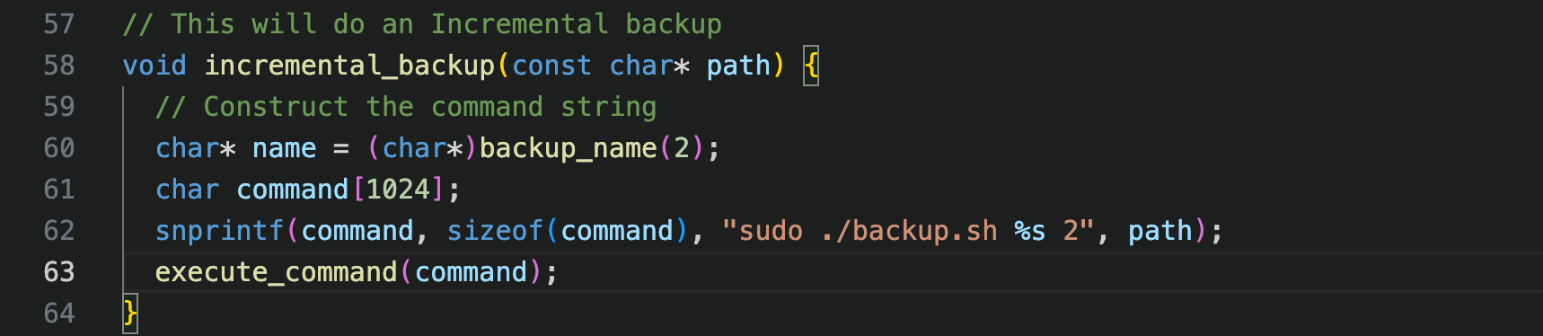
* **Full back up:**

The full\_backup function do a full back up, it accepts one parameter, path, which is a pointer to a character’s list that represent the path to the database or directory need to be backed up.



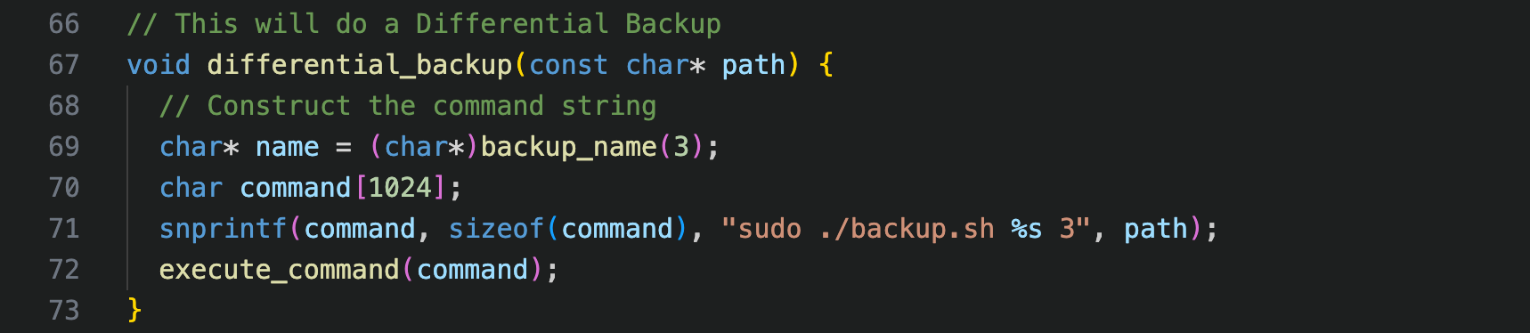
* **Incremental back up:**

The incremental\_backup function do an incremental back up, it accept one parameter, path, which is a pointer to a characters list that represent the path to the database or directory need to be backed up.



* **Differential back up:**

The differentail\_backup function does a differential back up, it accepts one parameter, path, which is a pointer to a characters list that represent the path to the database or directory need to be backed up.

****

## Option Library:

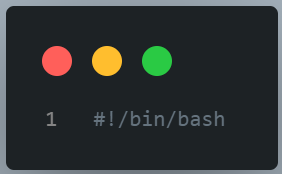
Contain all the functions that allow the user to control what to do a backup, restoration, exiting the program:

|  |  |
| --- | --- |
| void choose\_option(); | Allow the user to choose what option he need by selecting a number from 0 to 2:   1. For exiting 2. Back up 3. Restoration |
| char\* choose\_path(); | Take an input from the user and return a pointer of a list of characters that represent the path: |
| void choose\_backup(const char\* path); | Allow the user to choose backup type to use, change the path, or exiting by selecting a number from 0 to 4:   1. For exiting. 2. Full back up. 3. Incremental back up. 4. Differential back up. 5. Change path. |
| void choose\_restoration(const char\* path); |  |

**SCRIPT.SH :**

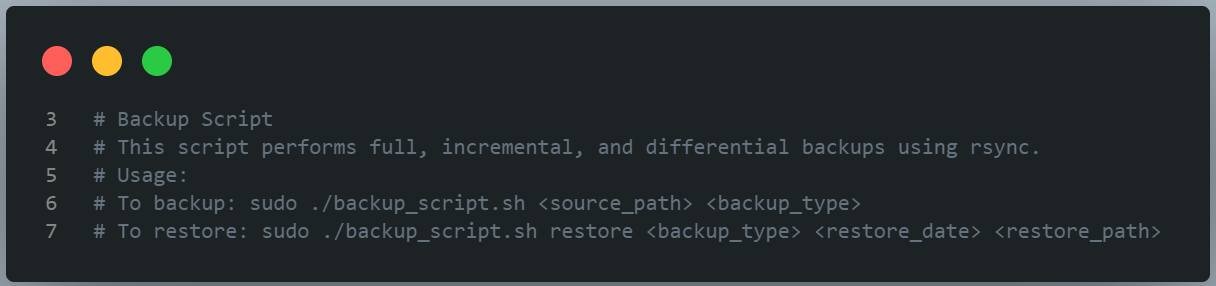
1. **Shebang :**

**Purpose**: This line tells the operating system that the script should be executed using the Bash shell. It is critical for script execution because it specifies the interpreter.

****

1. **Script Header Comment :**

A header comment explaining the purpose of the script, how to use it, and the expected parameters for backup and restoration operations.

****

1. **Destination Path :**

**Purpose**: This variable defines the base directory where all backups will be stored. Using /etc/backup suggests that this script is intended for a system-level backup, which typically requires elevated permissions (sudo).

****

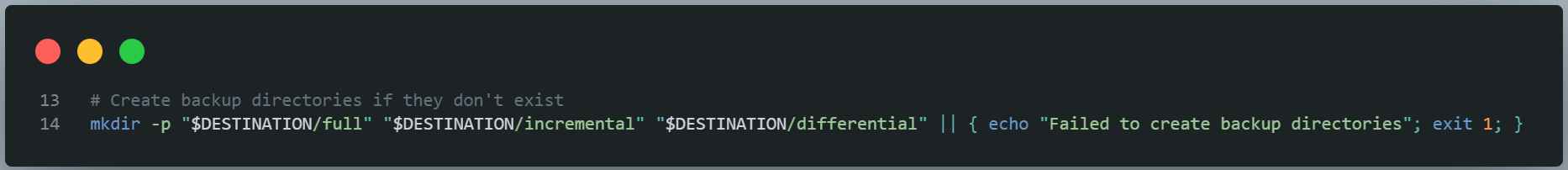
1. **Create Backup Directories :**

Purpose: This command creates three subdirectories:

* full: Stores full backups.
* incremental: Stores incremental backups.
* differential: Stores differential backups.

Explanation of Options:

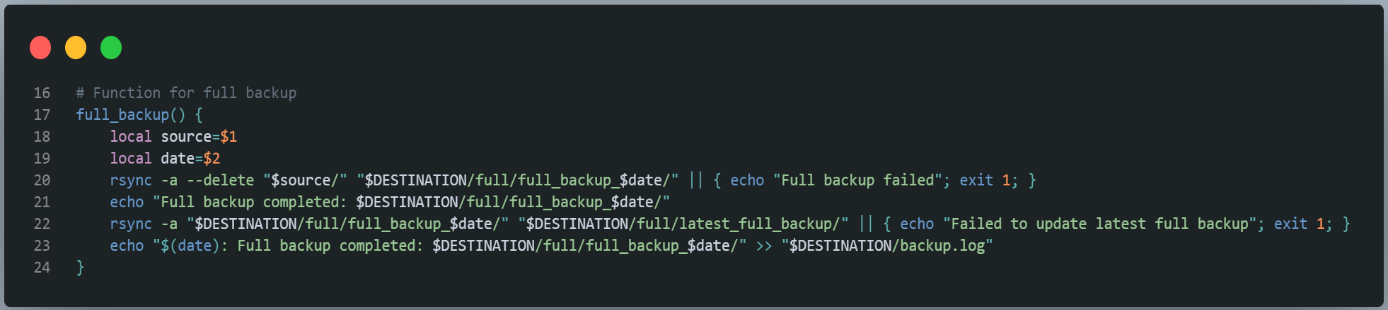
* -p : This option tells mkdir to create parent directories as needed, and it will not throw an error if the directories already exist.
* The || operator executes the command that follows it only if the preceding command fails (returns a non-zero exit status). In this case, if mkdir fails, an error message is printed, and the script exits.

****

1. **Function for Full Backup :**

* Parameters:
  + source: The directory you want to back up.
  + date: The current date/time to create a unique backup folder name.
* Detailed Steps:
  + rsync -a --delete "$source/" "$DESTINATION/full/full\_backup\_$date/":
    - rsync: A powerful file-copying tool, commonly used for backups.
    - -a: Archive mode; preserves permissions, timestamps, symbolic links, and directories.
    - --delete: Removes files from the destination that are no longer present in the source, maintaining an exact mirror.
  + Logging:
    - The success message is printed to the console.
    - The second rsync command updates a symlink for easy access to the latest full backup, copying the newly created full backup to a designated location named latest\_full\_backup.
    - The completion message, along with a timestamp, is logged into a backup.log file, providing a history of operations.

.

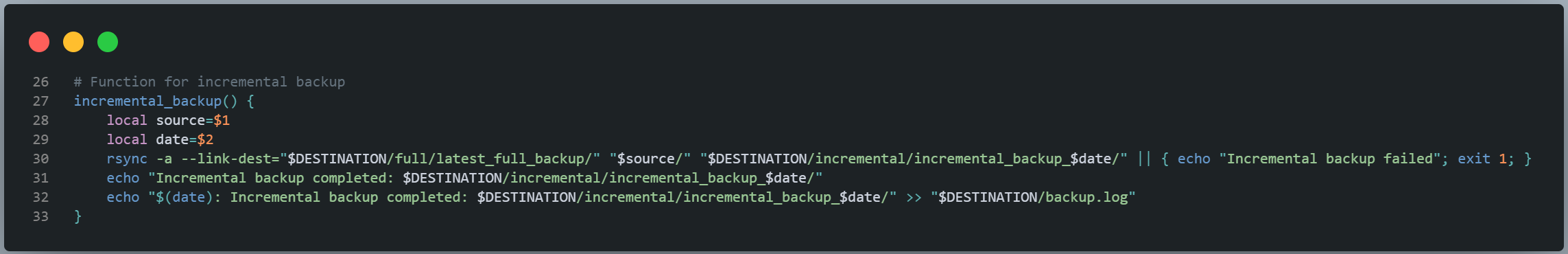
****

**6. Function for Incremental Backup :**

Function Definition: The incremental\_backup function takes the same parameters as the full backup function.

Backup Process:

* It uses rsync with --link-dest to perform an incremental backup:
  + This option allows rsync to create hard links for files that have not changed since the last full backup, significantly saving disk space.
* As with the full backup, it checks for errors and logs the completion message.

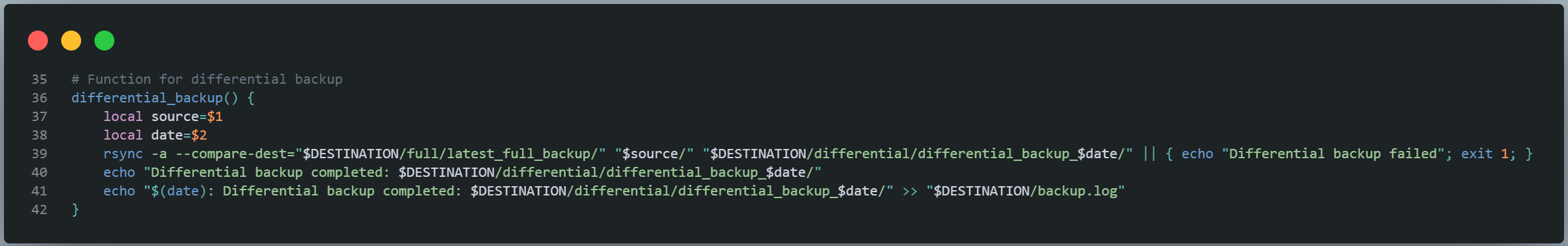
****

**7. Function for Differential Backup** :

Function Definition: Similar to the previous functions, differential\_backup also takes the source and date as parameters.

Backup Process:

* It uses rsync with --compare-dest:
  + This option compares files against the latest full backup, allowing it to copy only the files that have changed since that backup.
* It also includes error handling and logging.

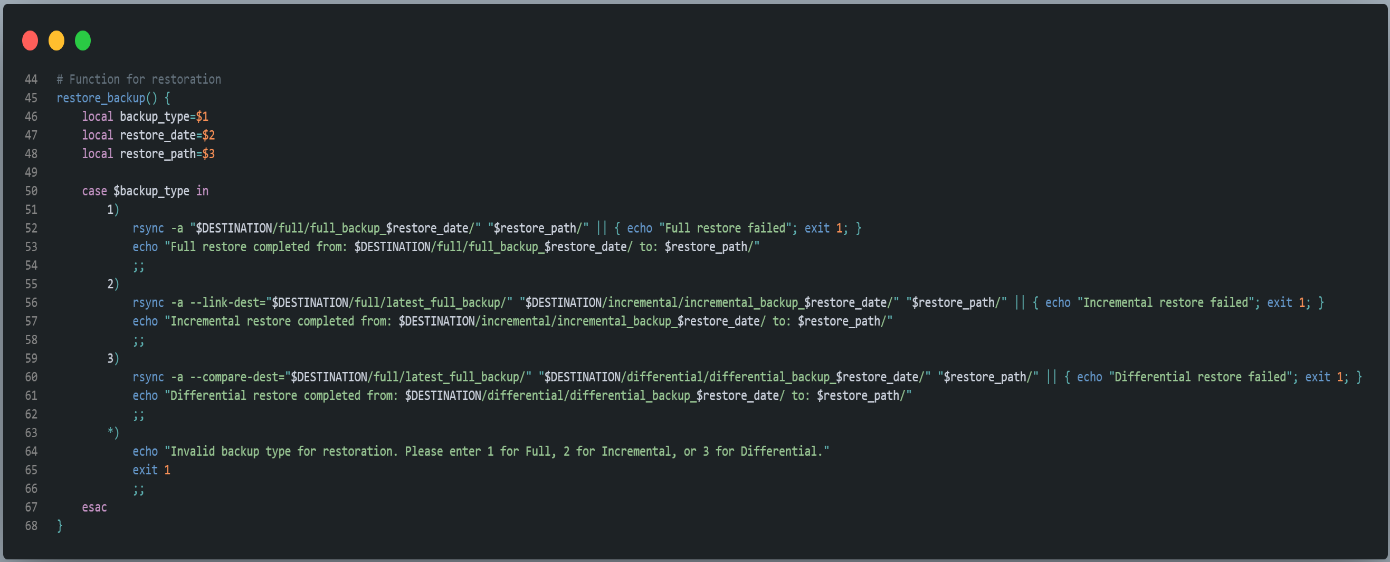
****

**8. Function for Restoration :**

Function Definition: The restore\_backup function handles the restoration of backups based on the specified type.

Restoration Process:

* The function uses a case statement to determine which type of backup to restore:
  + Full Restore: Directly copies from the full backup to the specified restore path.
  + Incremental Restore: Uses --link-dest to reference the latest full backup.
  + Differential Restore: Uses --compare-dest to copy files that changed since the last full backup.
* Error handling is included for each restoration type, along with completion messages.



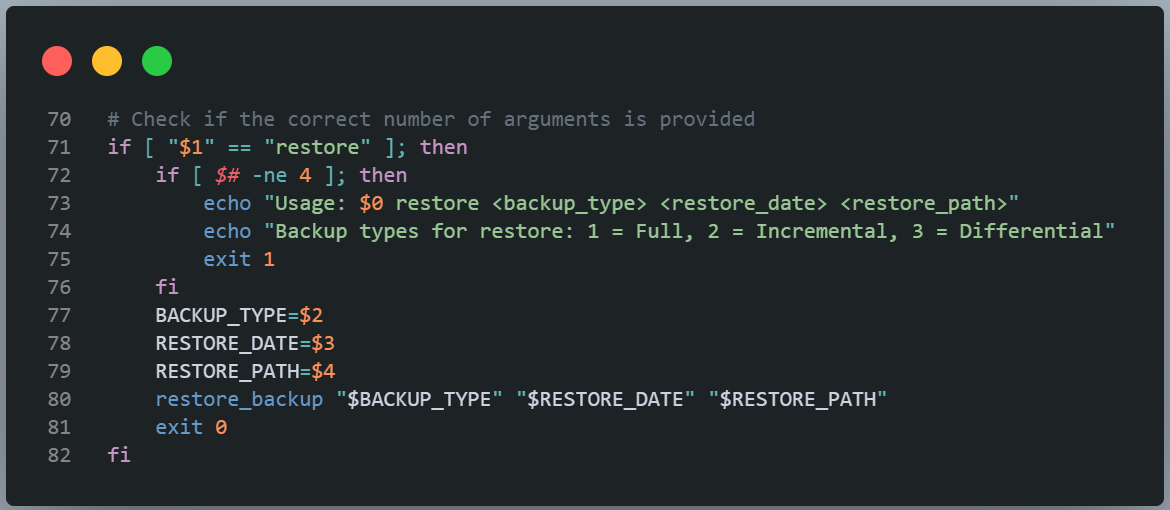
**9. Argument Parsing for Restore Command :**

This block checks if the first argument is "restore." If so, it expects four arguments:

* The command itself ($0), restore, the backup type, the restore date, and the restore path.

If the number of arguments is incorrect, it prints a usage message and exits.

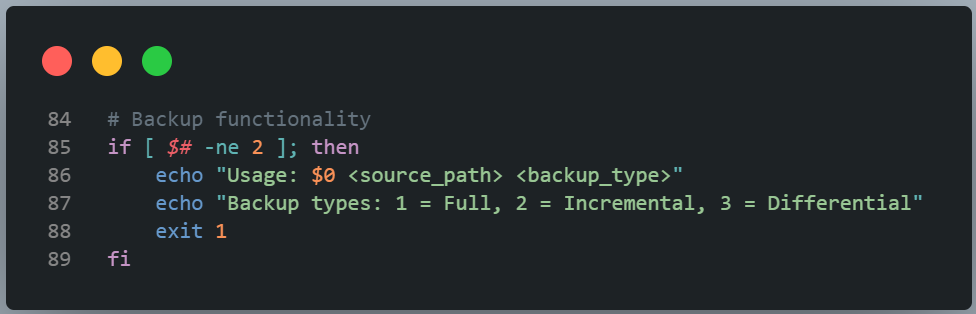
If everything is valid, it calls the restore\_backup function with the provided arguments.

****

**10. Argument Parsing for Backup Command :**

This checks that exactly two arguments are provided for backup operations (the source path and backup type).

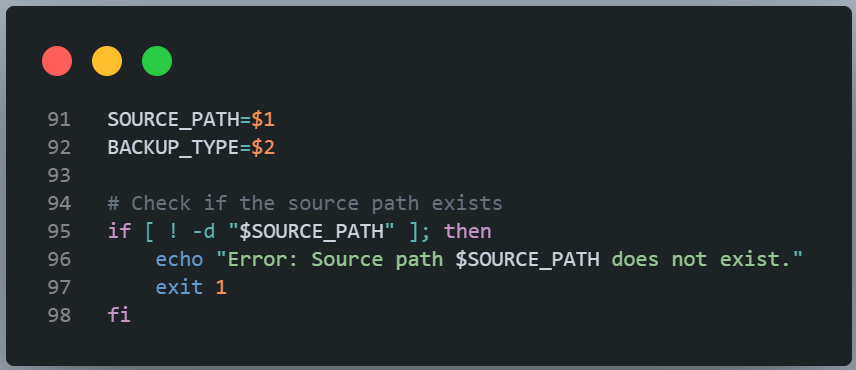
If the argument count is incorrect, it displays usage information and exits.



**11. Source Path Validation :**

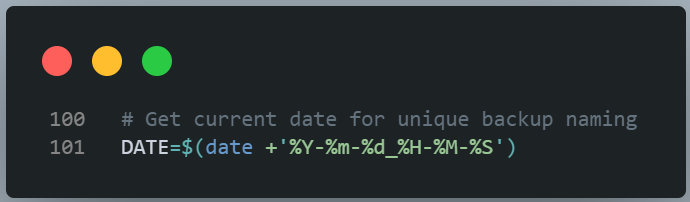
The source path is assigned from the first argument, and it checks if the specified source directory exists.

If the source path does not exist, an error message is displayed, and thescript exits**.**

****

**12. Get Current Date :**

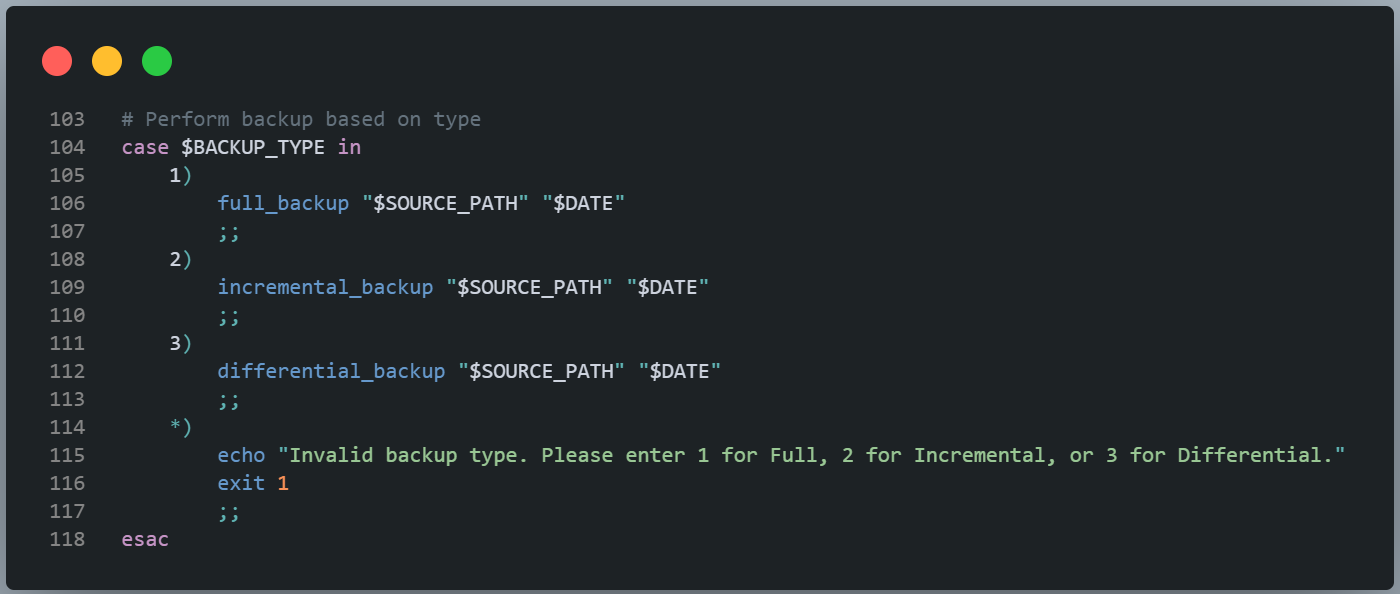
**Purpose**: Generates a timestamp to uniquely name the backup folders (e.g., 2024-11-01\_14-30-00), ensuring that each backup is easily identifiable..



**13. Perform Backup Based on Type :**

This case statement executes the appropriate backup function based on the specified backup type.

If an invalid backup type is entered, it displays an error message and exit



**14. Summary:**

This script provides a robust mechanism for creating backups of a specified directory in three different modes: full, incremental, and differential, with restoration capabilities. It also includes error handling and logging to keep track of completed operations, making it suitable for automated backup management tasks.