

CNS Practical 11

Aim: The aim is to implement a basic checksum algorithm for data integrity verification.

Theory:

Checksum Concept:

The concept of a checksum is a technique used to ensure the integrity of data during transmission or storage. It involves generating a fixed-size value (checksum) from the original data in a way that any subsequent changes to the data can be detected. Checksums are commonly employed in network communication, file storage, and data verification processes.

Checksums play a pivotal role in both error detection and correction, ensuring the integrity of data during transmission or storage. When data is transmitted over a network, it is susceptible to errors caused by factors like noise or packet loss. In this context, checksums serve as a mechanism for error detection. The sender calculates a checksum based on the data and transmits it alongside the actual data. At the receiving end, the checksum is recalculated, and any discrepancy between the transmitted and calculated checksums indicates the presence of errors in the data.

One key advantage of checksums is their ability to quickly identify errors. Instead of comparing the entire content of the data, a simple checksum comparison provides an efficient means of determining data integrity. Additionally, checksums can be applied to smaller blocks of data, allowing for the localization of errors within the dataset.

Most commonly used checksum algorithms in computer science and information technology to ensure data integrity are-

- Internet Checksum:
- Cyclic Redundancy Check (CRC):
- Adler-32:
- Checksum in TCP/UDP:
- MD5 (Message Digest Algorithm 5):
- SHA-1 (Secure Hash Algorithm 1):
- SHA-256 and SHA-3:

Algorithm:

Simple Checksum Algorithm:

1. Input:

Data to be Transmitted or Stored: A sequence of bytes or characters.

2. Algorithm Steps:

a. Conversion to Numerical Values:

Convert each byte in the data to its numerical value. For simplicity, you can use the ASCII code of each character.

b. Checksum Calculation:

Sum all the numerical values obtained in the previous step.

c. Modulo Operation:

Take the sum modulo a chosen prime number to obtain the checksum.

3. Mathematical Representation:

a. Let p be a prime number.

b. If $x_1, x_2, x_3, \dots, x_n$ are the numerical values of individual bytes in the data, the checksum C is calculated as:

$$C = (x_1 + x_2 + x_3 + \dots + x_n) \bmod p$$

4. Checksum Verification:

At the receiving end, the same algorithm is applied to the received data. The calculated checksum is compared with the transmitted checksum. A match indicates that the data is likely intact, while a mismatch suggests potential errors.

Code:

```
def calculate_checksum(data, prime_number):
    checksum = sum(byte for byte in data) % prime_number
    return checksum

original_data = input("Enter the text: ").encode('utf-8')
chosen_prime = int(input("Enter a Prime No. = ")) # Choose a prime number for the modulo operation
checksum_result = calculate_checksum(original_data, chosen_prime)

print(f"Original Data: {original_data}")
print(f"Checksum: {checksum_result}")
```

Output:

A screenshot of a Windows command prompt window. The title bar shows 'Python' and standard window controls. The terminal text shows the execution of a Python script 'prime.py'. The user enters 'Hello World' and a prime number '79'. The script outputs the original data as a byte string and a checksum of 25. The status bar at the bottom indicates 'Ln 10, Col 38', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python 3.11.6 64-bit', and icons for search and notifications.

```
PS C:\Users\Administrator> & C:/Users/Administrator/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Administrator/Desktop/prime.py
Enter the text: Hello World
Enter a Prime No. = 79
Original Data: b'Hello World'
Checksum: 25
PS C:\Users\Administrator>
```

Viva Questions:

1. What is the purpose of a checksum in data communication?

Ans.1) The purpose of a checksum in data communication is to ensure the integrity of transmitted data. It serves as a simple and effective method for detecting errors that may occur during the transmission process.

2. What are the steps of implementing checksum algorithm.

Ans.2) The steps of implementing Checksum Algorithm are -

- i. Define the Data
- ii. Choose a Prime Number
- iii. Convert Data to Numerical Values
- iv. Sum Numerical Values
- v. Apply Modulo Operation
- vi. Transmit or Store Data with Checksum
- vii. Checksum Verification (Receiving End)
- viii. Error Handling (Optional)

3. How does the choice of the prime number in the modulo operation affect the checksum?

Ans.3) The choice of the prime number in the modulo operation affects the uniqueness, distribution, error detection sensitivity, and collision avoidance of checksums, influencing the overall robustness and efficiency of the checksum algorithm.