crc32 package standard library

Version: go1.20.1 Latest | Published: Feb 14, 2023 | License: BSD-3-Clause | Imports: 6 | Imported by: 11,162

Details Valid go.mod file Redistributable license Tagged version Stable version

Learn more

Repository cs.opensource.google/go/go

Links Report a Vulnerability

Documentation

Documentation

Rendered for linux/amd64

Overview

Package crc32 implements the 32-bit cyclic redundancy check, or CRC-32, checksum. See https://en.wikipedia.org/wiki/Cyclic_redundancy_check for information.

Polynomials are represented in LSB-first form also known as reversed representation.

See https://en.wikipedia.org/wiki/Mathematics_of_cyclic_redundancy_checks#Reversed_representations_and_reciprocal_polynomials for information.

Index

Examples

Constants

View Source

```
const (
    // IEEE is by far and away the most common CRC-32 polynomial.
    // Used by ethernet (IEEE 802.3), v.42, fddi, gzip, zip, png, ...
    IEEE = 0xedb88320

    // Castagnoli's polynomial, used in iSCSI.
    // Has better error detection characteristics than IEEE.
    // https://dx.doi.org/10.1109/26.231911
    Castagnoli = 0x82f63b78

    // Koopman's polynomial.
    // Also has better error detection characteristics than IEEE.
    // https://dx.doi.org/10.1109/DSN.2002.1028931
    Koopman = 0xeb31d82e
)
```

Predefined polynomials.

View Source

```
const Size = 4
```

The size of a CRC-32 checksum in bytes.

Variables

View Source

```
var IEEETable = simpleMakeTable(IEEE)
```

IEEETable is the table for the IEEE polynomial.

Functions

func Checksum

```
func Checksum(data []byte, tab *Table) uint32
```

Checksum returns the CRC-32 checksum of data using the polynomial represented by the Table.

func ChecksumIEEE

```
func ChecksumIEEE(data []byte) uint32
```

ChecksumIEEE returns the CRC-32 checksum of data using the IEEE polynomial.

func New

```
func New(tab *Table) hash.Hash32
```

New creates a new hash.Hash32 computing the CRC-32 checksum using the polynomial represented by the Table. Its Sum method will lay the value out in big-endian byte order. The returned Hash32 also implements encoding.BinaryMarshaler and encoding.BinaryUnmarshaler to marshal and unmarshal the internal state of the hash.

func NewIEEE

```
func NewIEEE() hash.Hash32
```

NewIEEE creates a new hash.Hash32 computing the CRC-32 checksum using the IEEE polynomial. Its Sum method will lay the value out in big-endian byte order. The returned Hash32 also implements encoding.BinaryMarshaler and encoding.BinaryUnmarshaler to marshal and unmarshal the internal state of the hash.

func Update

```
func Update(crc uint32, tab *Table, p []byte) uint32
```

Update returns the result of adding the bytes in p to the crc.

Types

type Table

```
type Table [256]uint32
```

Table is a 256-word table representing the polynomial for efficient processing.

func MakeTable

```
func MakeTable(poly uint32) *Table
```

MakeTable returns a Table constructed from the specified polynomial. The contents of this Table must not be modified.

▶ Example

Source Files

View all

crc32.go crc32_amd64.go crc32_generic.go