

Programmazione I

Lezione 18
10/11/22

Il sistema dei tipi

- Tipi “base” in GO

- `bool`
- `string`
- `int` `int8` `int16` `int32` `int64`
- `uint` `uint8` `uint16` `uint32` `uint64` `uintptr`
- `byte` // alias for `uint8`
- `rune` // alias for `int32`, represents a Unicode code point
- `float32` `float64`
- `complex64` `complex128`

- ASCII: ogni carattere usa 1 byte → 128 caratteri
- UNICODE → contiene
 - attualmente 137.994 “grafemi” (codepoints)
 - fino a 1.114.112 (17 planes, 2^{16} c.p per plane)
 - I primi 128 sono gli stessi dell’ASCII
 - Necessari fino a 4 byte (pensate a Maxint per 32 bit)
- Rappresentare UNICODE:
 - Standard di codifica fissa →
UTF-32 (sempre 4 byte), UTF-16 (2 byte, parziale)
 - Standard di codifica variabile → **UTF-8** (da 1 a 6 byte)
 - primi bit del primo byte: flag che indicano quanti byte sono usati (tanti “1” quanti sono i byte usati, poi “0”)
 - primi 2 bit di ogni altro byte: 10 se continua
 - Esempio: A → 65 → 01000001 → 1 byte (come tutto ASCII)
 - Esempio: “point of interest” → 8984 → 1000 1100 011000 → 3 byte

UTF-8: l'esempio di "Point of Interest"

- // Codice GO HEX 2318, DEC 8984, BIN 0010 0011 0001 1000

myc = '⌘'

fmt.Printf("%c %5d %032b %x\n",myc,myc,myc,myc)

→ ⌘ 8984 0000000000000000000000000010001100011000 2318

- UTF-8
 - 1110 (preambolo, tre byte)
0010 (completa primo byte)
 - 10 (preambolo secondo byte)
001100 (completa secondo byte)
 - 10 (preambolo terzo byte)
011000 (completa terzo byte)
11100010 10001100 10011000

“Catena” della codifica di un carattere GO

