

Литература

Си

- [ISO/IEC 9899:1999: Programming languages - C](#)
- [ISO/IEC 9899:201x: Programming languages - C](#)
- [Wiki: C Programming/stdlib.h](#)
- [Yandex Music: Научно-технический рэп: Папа может в Си](#)

Обработка ошибок в Си

- [Carnegie Mellon University and IEEE: David Svoboda: Beyond errno. Error Handling in C](#)
- [GeeksForGeeks: Mazhar Imam Khan: Error Handling in C programs](#)
- [IBM: Errors: errno in UNIX programs](#)
- [Laurence Tratt: A Proposal for Error Handling](#)
- [Stack Overflow: Will Hartung: Any good idioms for error handling in straight C programs?](#)
- [Tutorialspoint: C Error Handling](#)
- [Tutorialspoint: Ankitha Reddy: Why should we avoid using global variables in C/C++?](#)
- [Wiki: C Programming/Error handling](#)
- [Wiki: errno.h](#)
- [Библиотека языка C GNU glibc: Сообщения об ошибках](#)
- [Список всех возможных значений errno в Linux \(kdave: errno.h for the masses\):](#)
 - [torvalds/linux/include/uapi/asm-generic/errno-base.h](#)
 - [torvalds/linux/include/uapi/asm-generic/errno.h](#)

- [Хабр: arielf: Обработка ошибок в C](#)

SEI CERT C Coding Standard

- [Wiki: CERT C Coding Standard](#)
- [SEI CERT C Coding Standard: Rule 12. Error Handling \(ERR\):](#)
 - [ERR32-C. Do not rely on indeterminate values of errno](#)
 - [ERR33-C. Detect and handle standard library errors](#)
 - и пр.
- [SEI CERT C Coding Standard: Rec. 12. Error Handling \(ERR\):](#)
 - [ERR00-C. Adopt and implement a consistent and comprehensive error-handling policy](#)
 - [ERR02-C. Avoid in-band error indicators](#)
 - [ERR05-C. Application-independent code should provide error detection without dictating error handling](#)
 - и пр.

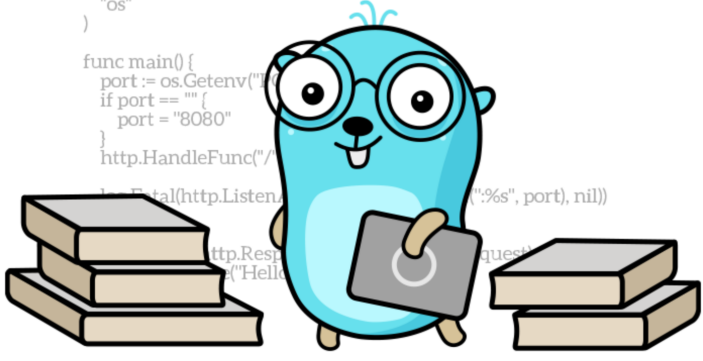
Прочее

- [Joshua Bloch: How to Design a Good API and Why it Matters \(JAVA\)](#)
- [Tutorialspoint: Who is who in Computer Science](#)

```
// This server can run on App Engine.
package main

import (
    "fmt"
    "log"
    "net/http"
    "os"
)

func main() {
    port := os.Getenv("PORT")
    if port == "" {
        port = "8080"
    }
    http.HandleFunc("/hello", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintln(w, "Hello, world!")
    })
    log.Fatal(http.ListenAndServe(":"+port, nil))
}
```



go learn()