

Table of Contents

| | |
|-----------------------|-----------|
| Graphic Bundle | 1 |
| Index | 22 |

Graphic Bundle

Chapter 1: Getting Started with Go and Unix Systems Programming

```
1Mac:code mtsouk$ godoc fmt.Println
use "godoc cmd/fmt" for documentation on the fmt command

func Println(a ...interface{}) (n int, err error)
Println formats using the default formats for its operands and writes to
standard output. Spaces are always added between operands and a newline
is appended. It returns the number of bytes written and any write error
encountered.

1Mac:code mtsouk$ godoc cmd/fmt
PACKAGE DOCUMENTATION

package fmt
import "."

Package fmt implements formatted I/O with functions analogous to C's
printf and scanf. The format 'verbs' are derived from C's but are
simpler.

Printing

The verbs:

General:

%v    the value in a default format
      when printing structs, the plus flag (%+v) adds field names
%#v   a Go-syntax representation of the value
%T    a Go-syntax representation of the type of the value
%%   a literal percent sign; consumes no value

Boolean:

%t    the word true or false

Integer:

%b    base 2
%c    the character represented by the corresponding Unicode code point
%d    base 10
%o    base 8
%q    a single-quoted character literal safely escaped with Go syntax.
%x    base 16, with lower-case letters for a-f
%X    base 16, with upper-case letters for A-F
%U    Unicode format: U+1234; same as "U+%04X"

Floating-point and complex constituents:

%b    decimalless scientific notation with exponent a power of two,
```


[Documents](#)
[Packages](#)
[The Project](#)
[Help](#)
[Blog](#)

Packages

- Standard library
- Other packages
 - Sub-repositories
 - Community

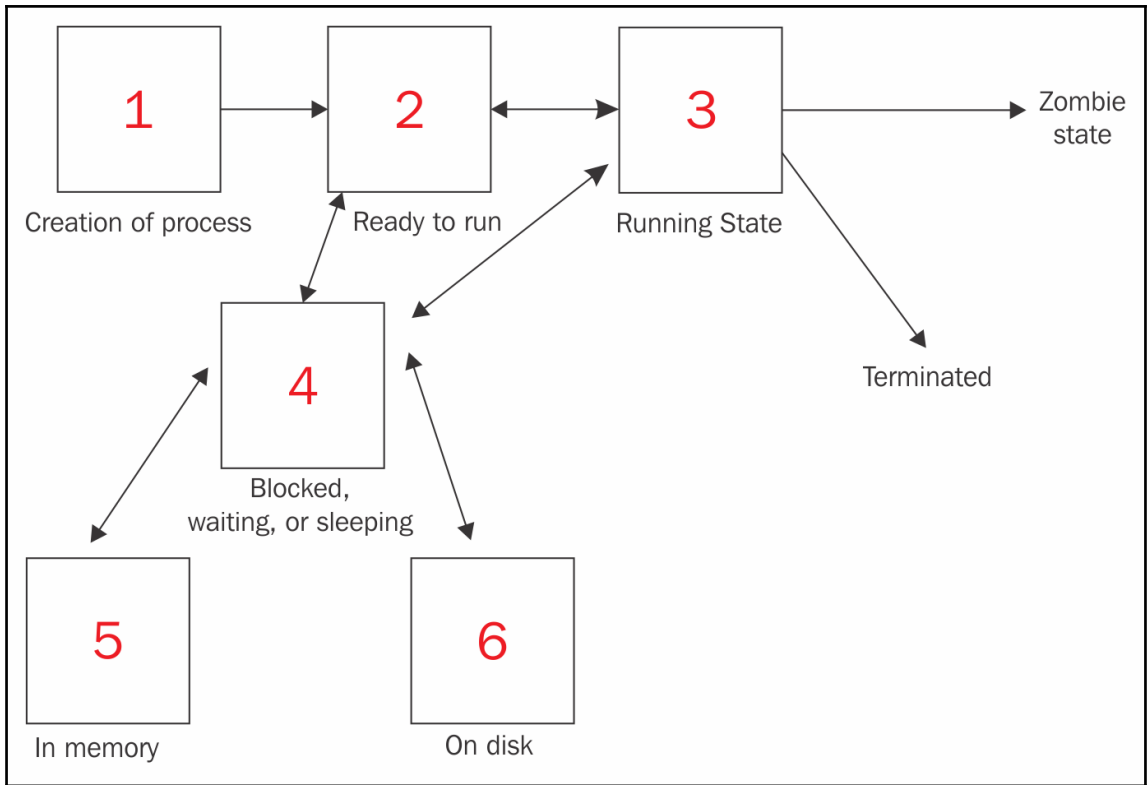
Standard library

| Name | Synopsis |
|-----------|---|
| archive | |
| tar | Package tar implements access to tar archives. |
| zip | Package zip provides support for reading and writing ZIP archives. |
| bufio | Package bufio implements buffered I/O. It wraps an io.Reader or io.Writer object, creating another object (Reader or Writer) that also implements the interface but provides buffering and some help for textual I/O. |
| builtin | Package builtin provides documentation for Go's predeclared identifiers. |
| bytes | Package bytes implements functions for the manipulation of byte slices. |
| compress | |
| bzip2 | Package bzip2 implements bzip2 decompression. |
| flate | Package flate implements the DEFLATE compressed data format, described in RFC 1951. |
| gzip | Package gzip implements reading and writing of gzip format compressed files, as specified in RFC 1952. |
| lz7 | Package lz7 implements the Lempel-Ziv-Welch compressed data format, described in T. A. Welch, "A Technique for High-Performance Data Compression", Computer, 17(6) (June 1984), pp 8-19. |
| zlib | Package zlib implements reading and writing of zlib format compressed data, as specified in RFC 1950. |
| container | |
| heap | Package heap provides heap operations for any type that implements heap.Interface. |
| list | Package list implements a doubly linked list. |
| ring | Package ring implements operations on circular lists. |
| context | Package context defines the Context type, which carries deadlines, cancelation signals, and other request-scoped values across API boundaries and between processes. |
| crypto | Package crypto collects common cryptographic constants. |
| aes | Package aes implements AES encryption (formerly Rijndael), as defined in U.S. Federal Information Processing Standards Publication 197. |
| cipher | Package cipher implements standard block cipher modes that can be wrapped around low-level block cipher implementations. |
| des | Package des implements the Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDEA) as defined in U.S. Federal Information Processing Standards Publication 46-3. |
| dsa | Package dsa implements the Digital Signature Algorithm, as defined in FIPS 186-3. |
| ecdsa | Package ecdsa implements the Elliptic Curve Digital Signature Algorithm, as defined in FIPS 186-3. |
| elliptic | Package elliptic implements several standard elliptic curves over prime fields. |
| hmac | Package hmac implements the Keyed-Hash Message Authentication Code (HMAC) as defined in U.S. Federal Information Processing Standards Publication 198. |



```
cp.go — code
1 package main
2
3 import (
4     "fmt"
5     "io/ioutil"
6     "os"
7 )
8
9 func main() {
10     if len(os.Args) != 3 {
11         fmt.Println("Please use two command line arguments!")
12         os.Exit(-1)
13     }
14
15     in := os.Args[1]
16     out := os.Args[2]
17
18     input, err := ioutil.ReadFile(in)
19     if err != nil {
20         fmt.Println("Error reading the input!")
21     }
22
23     err = ioutil.WriteFile(out, input, 0644)
24     if err != nil {
25         fmt.Println("Error creating the destination file!")
26     }
27
28 }
29
```

Line: 11:23 | Go | Tab Size: 4 | Settings



Chapter 2: Writing Programs in Go

```
2.mtsouk@mail: ~ (bash)
rMacBook:code mtsouk$ go help environment
The go command, and the tools it invokes, examine a few different
environment variables. For many of these, you can see the default
value of on your system by running 'go env NAME', where NAME is the
name of the variable.

General-purpose environment variables:

GCCGO      The gccgo command to run for 'go build -compiler=gccgo'.
GOARCH     The architecture, or processor, for which to compile code.
           Examples are amd64, 386, arm, ppc64.
GOBIN      The directory where 'go install' will install a command.
GOOS       The operating system for which to compile code.
           Examples are linux, darwin, windows, netbsd.
GOPATH     For more details see: 'go help gopath'.
GORACE     Options for the race detector.
           See https://golang.org/doc/articles/race\_detector.html.
GOROOT     The root of the go tree.

Environment variables for use with cgo:

CC         The command to use to compile C code.
CGO_ENABLED Whether the cgo command is supported. Either 0 or 1.
CGO_FLAGS  Flags that cgo will pass to the compiler when compiling
           C code.
CGO_CPPFLAGS Flags that cgo will pass to the compiler when compiling
           C or C++ code.
CGO_CXXFLAGS Flags that cgo will pass to the compiler when compiling
           C++ code.
CGO_FFLAGS
```

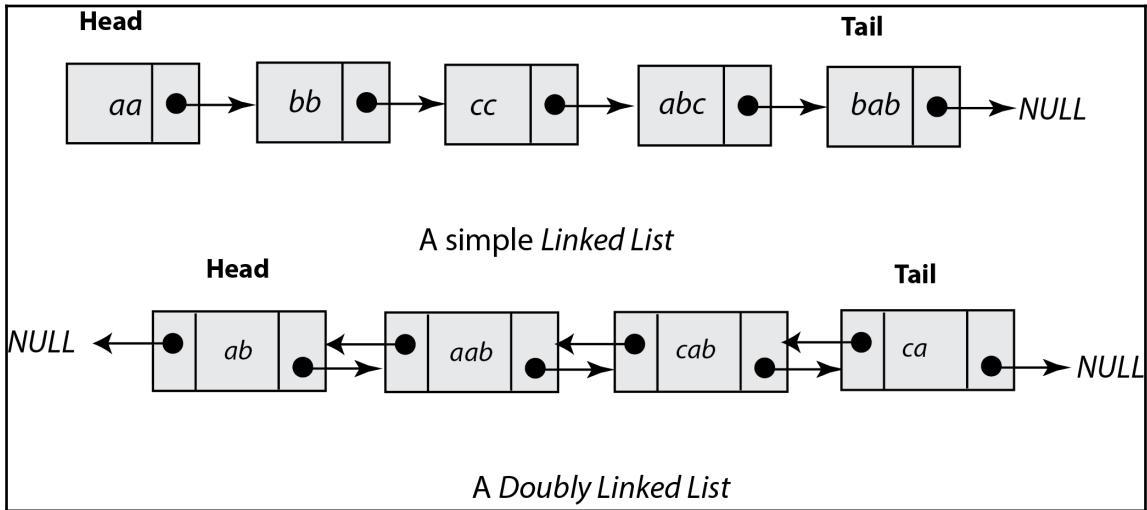
Chapter 3: Advanced Go Features

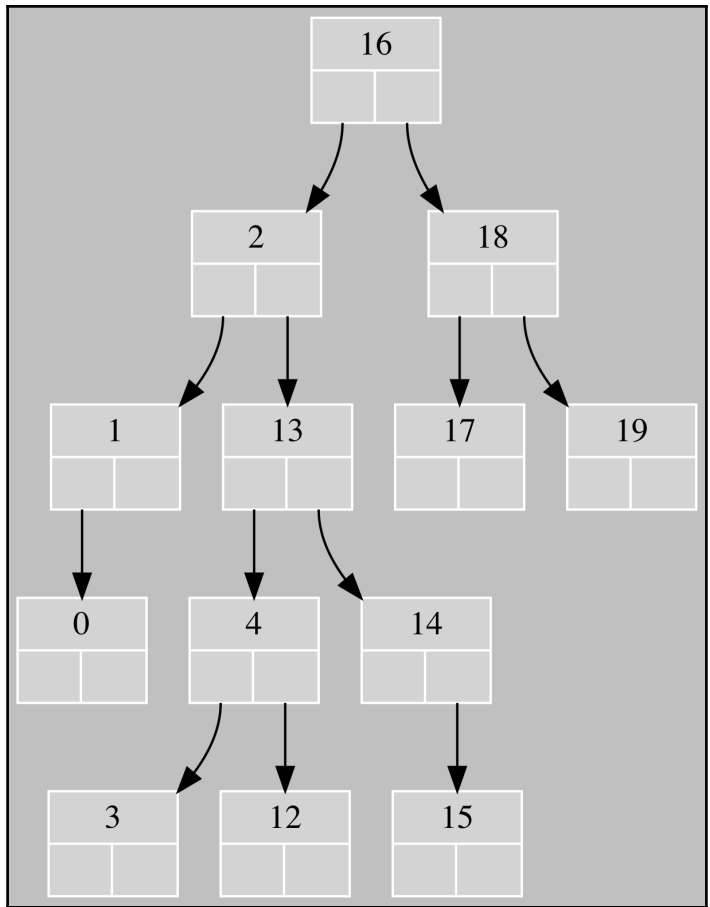
```
2. mtsouk@mail: ~/Desktop/goBook/ch/ch3/code (ssh)
code$ strace ./addCLAImproved 1 2
execve("./addCLAImproved", [ "./addCLAImproved", "1", "2"], [ /* 22 vars */ ]) = 0
arch_prctl(ARCH_SET_FS, 0x544990) = 0
sched_getaffinity(0, 128, {1}) = 8
mmap(0xc000000000, 65536, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xc000000000
munmap(0xc000000000, 65536) = 0
mmap(NULL, 262144, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f09f6cf6000
mmap(0xc208000000, 1048576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xc208000000
mmap(0xc207ff0000, 65536, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xc207ff0000
mmap(0xc000000000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xc000000000
mmap(NULL, 65536, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f09f6ce6000
mmap(NULL, 1439992, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f09f6b86000
mmap(NULL, 131072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f09f6b66000
sigaltstack({ss_sp=0xc208006000, ss_flags=0, ss_size=32768}, NULL) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
rt_sigaction(SIGHUP, NULL, {SIG_DFL, [], 0}, 8) = 0
rt_sigaction(SIGHUP, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGINT, NULL, {SIG_DFL, [], 0}, 8) = 0
rt_sigaction(SIGINT, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGQUIT, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGILL, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGTRAP, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGABRT, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGBUS, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGFPE, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGUSR1, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGSEGV, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGUSR2, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGPIPE, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGALRM, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGTERM, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
rt_sigaction(SIGSTKFLT, {0x428480, ~[], SA_RESTORER|SA_STACK|SA_RESTART|SA_SIGINFO, 0x4284f0}, NULL, 8) = 0
```

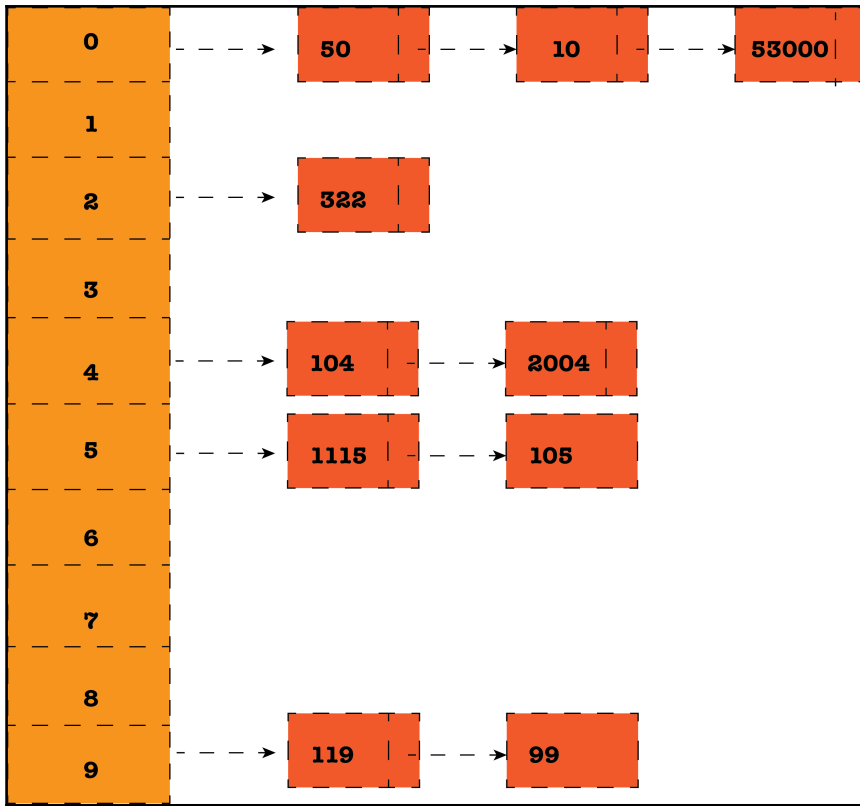
```
2. bash
rMacBook:code mtsouk$ sudo dtruss ./addCLAImproved 1
SYSALL(args) = return
fatal error: runtime: bsdthread_register error

runtime stack:
runtime.throw(0x10b6e6d, 0x21)
    /usr/local/Cellar/go/1.8/libexec/src/runtime/panic.go:596 +0x95 fp=0x7fff5fbffb20 sp=0x7fff5fbffb00
runtime.goenvs()
    /usr/local/Cellar/go/1.8/libexec/src/runtime/os_darwin.go:108 +0xa0 fp=0x7fff5fbffb50 sp=0x7fff5fbffb20
runtime.schedinit()
    /usr/local/Cellar/go/1.8/libexec/src/runtime/proc.go:486 +0xa1 fp=0x7fff5fbffb90 sp=0x7fff5fbffb50
runtime.r0_go(0x7fff5fbffbc0, 0x2, 0x7fff5fbffbc0, 0x0, 0x2, 0x7fff5fbffce0, 0x7fff5fbffc1, 0x0, 0x7fff5fbffc3, 0x7fff5fbffd01, ...)
    /usr/local/Cellar/go/1.8/libexec/src/runtime/asm_amd64.s:158 +0x183 fp=0x7fff5fbffb98 sp=0x7fff5fbffb90
stat64("/usr/lib/libc++.1.dylib\0", 0x7FFF5FBFDD48, 0x1) = 0 0
stat64("/usr/lib/libSystem.B.dylib\0", 0x7FFF5FBFDD48, 0x1) = 0 0
stat64("/usr/lib/libc++abi.dylib\0", 0x7FFF5FBFDC58, 0x1) = 0 0
stat64("/usr/lib/system/libcache.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libcommonCrypto.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libcompiler_rt.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libcopyfile.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libcorecrypto.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libdyld.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libkeymgr.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/liblaunch.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libmacho.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libquarantine.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libremovefile.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libsystem_asl.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libsystem_blocks.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libsystem_c.dylib\0", 0x7FFF5FBFD788, 0x1) = 0 0
stat64("/usr/lib/system/libsystem_configuration.dylib\0", 0x7FFF5FBFD788, 0x1) =
0 0
stat64("/usr/lib/system/libsystem_coreservices.dylib\0", 0x7FFF5FBFD788, 0x1) =
0 0
stat64("/usr/lib/system/libsystem_coretls.dylib\0", 0x7FFF5FBFD788, 0x1) =
0 0
```

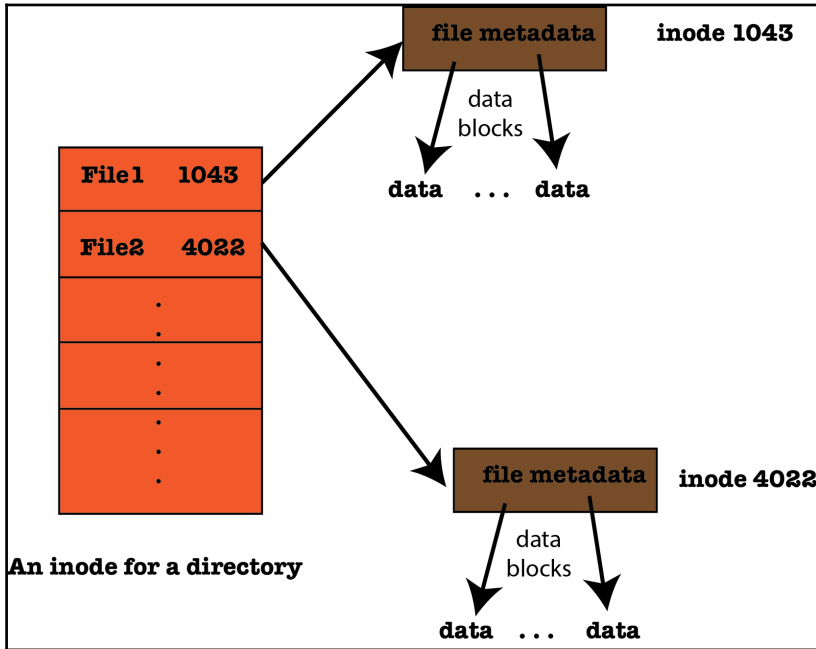

Chapter 4: Go Packages, Algorithms, and Data Structures

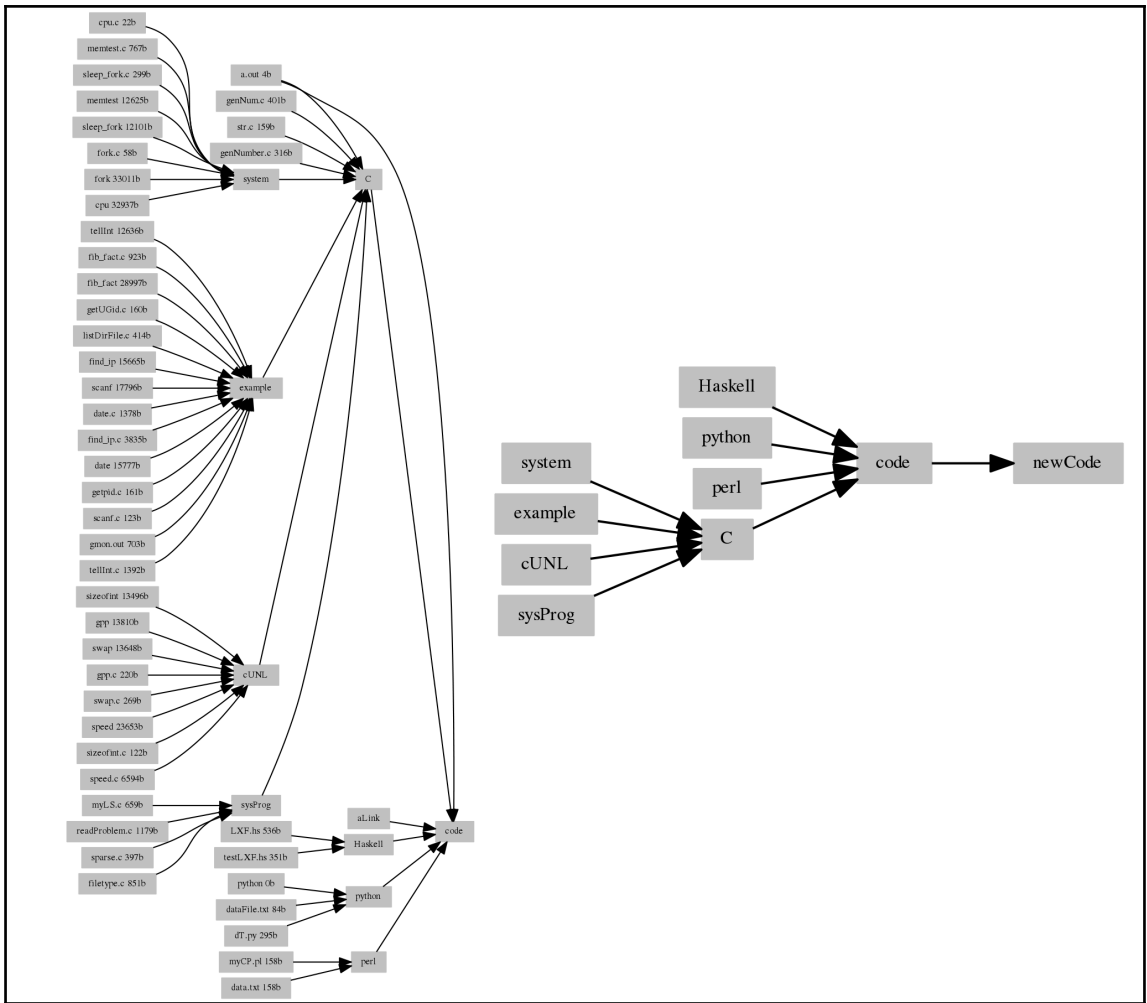




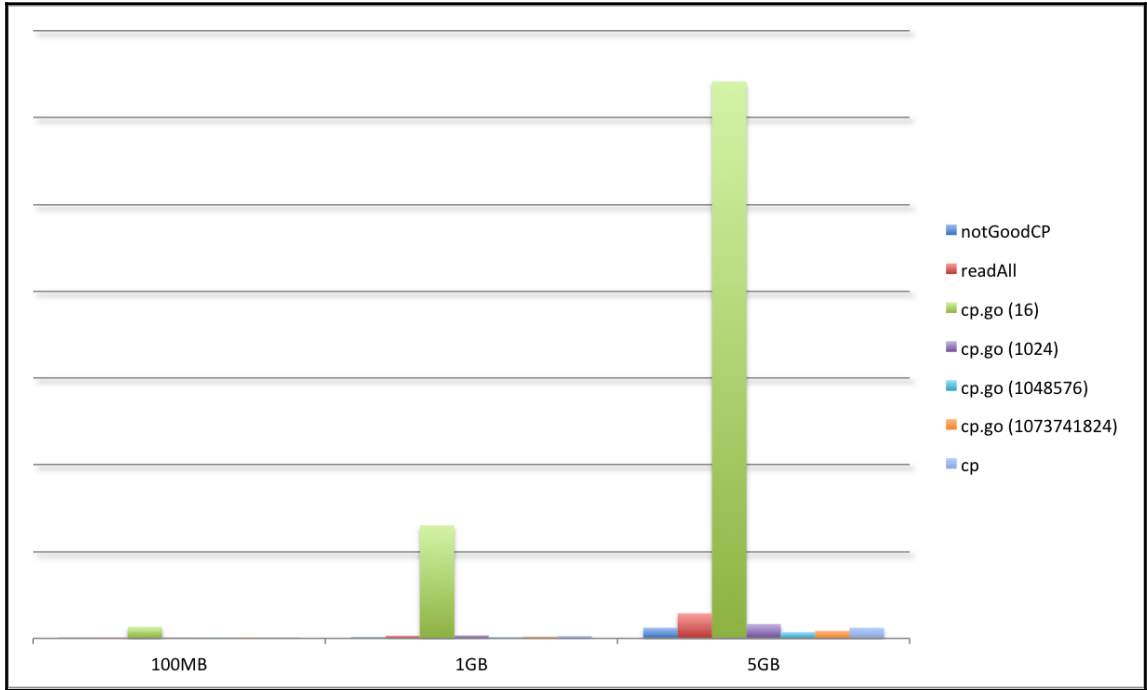


Chapter 5: Files and Directories

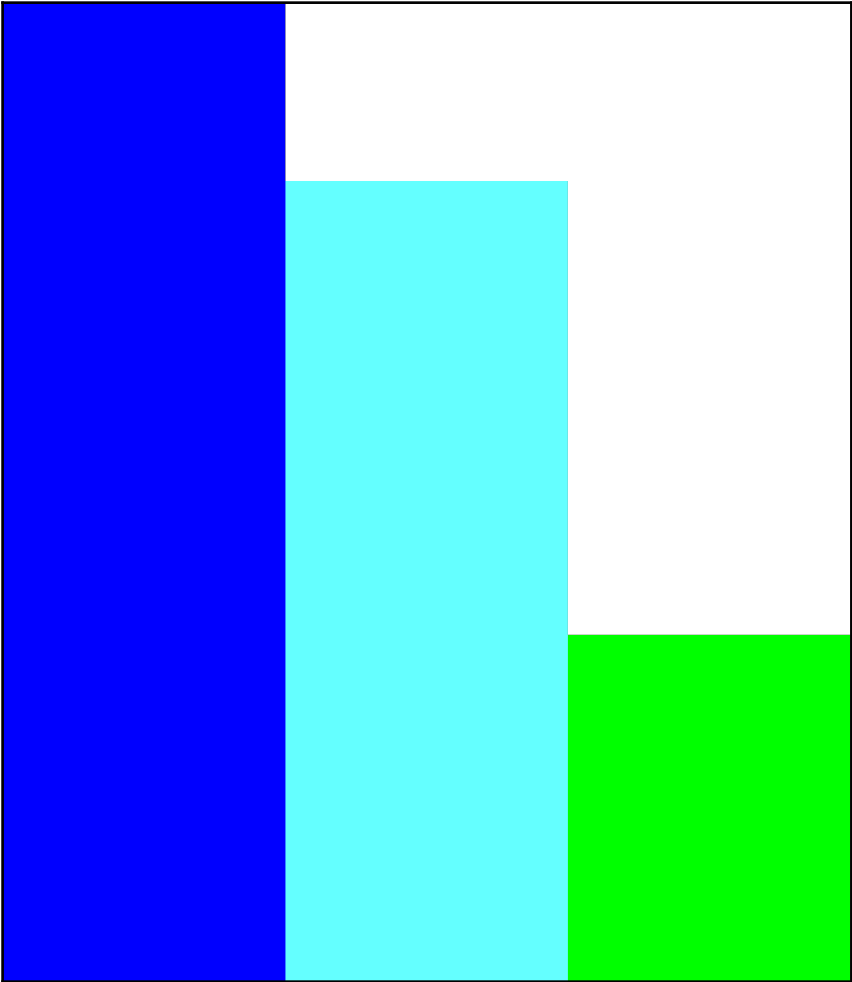


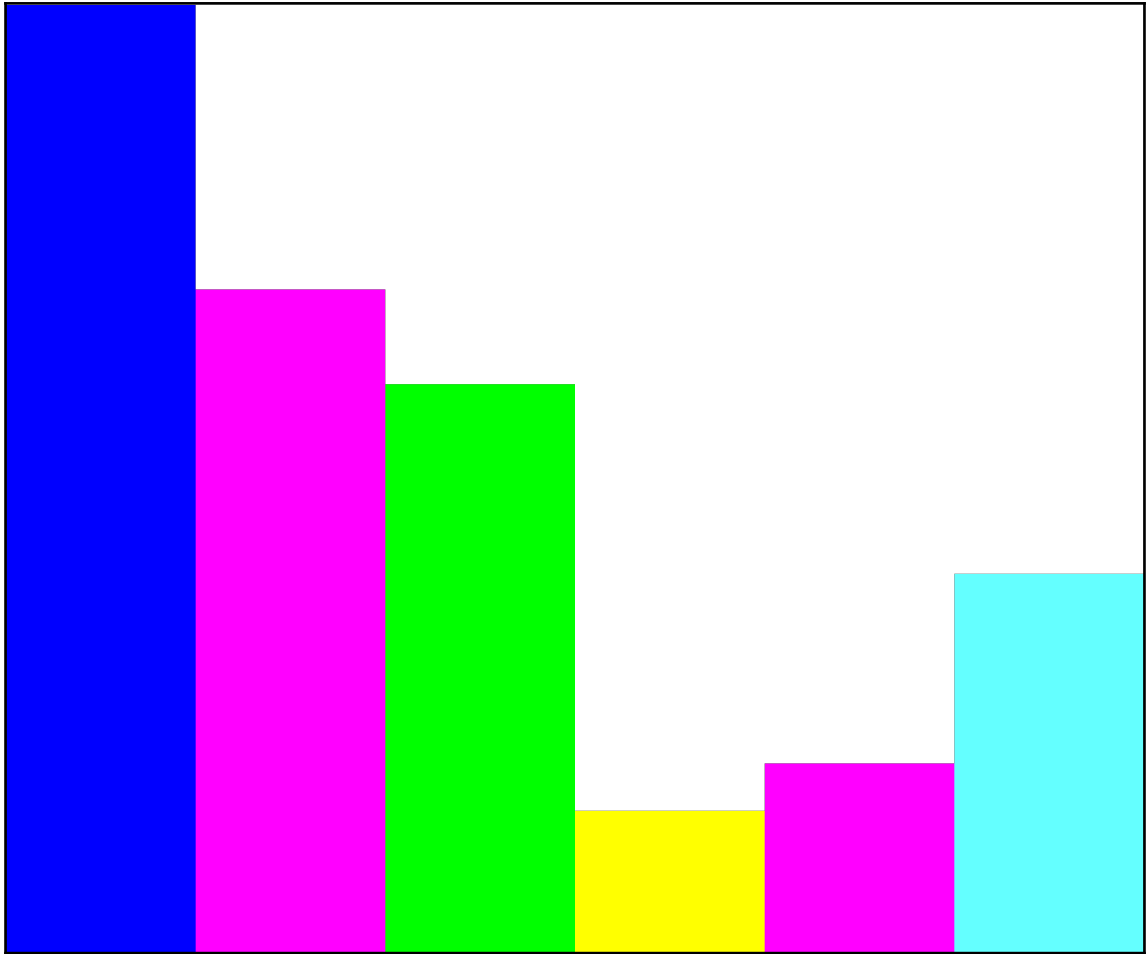


Chapter 6: File Input and Output

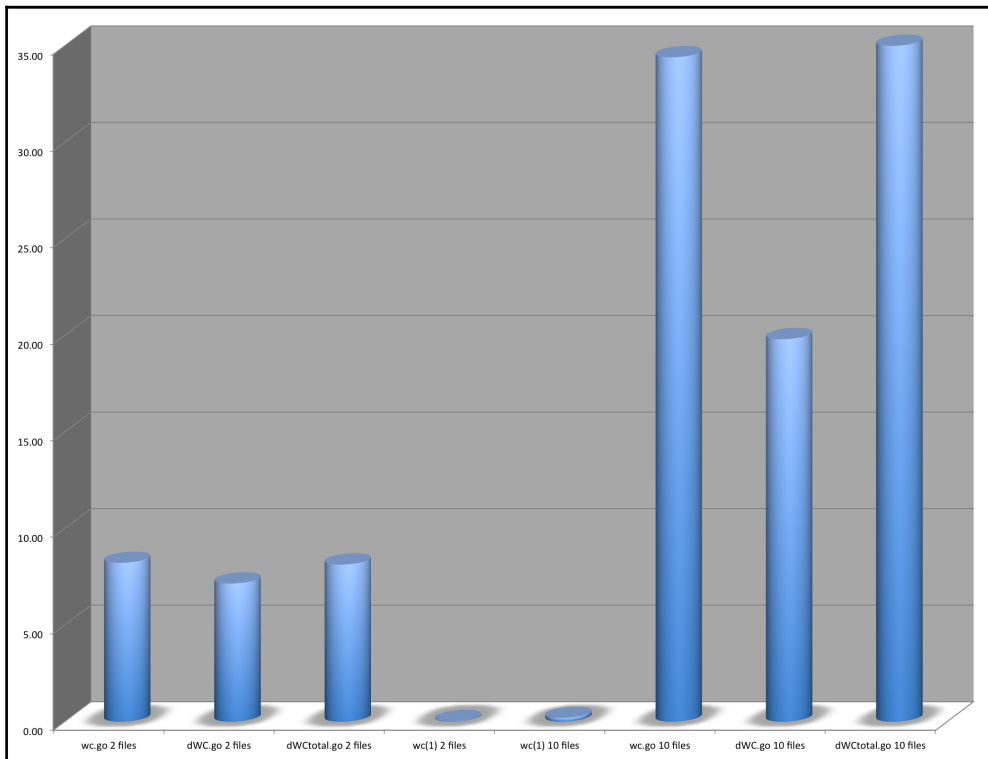
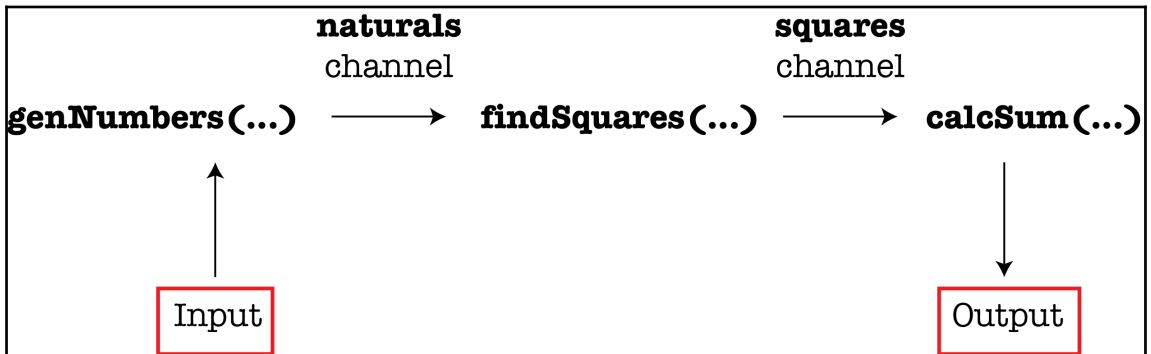


Chapter 8: Processes and Signals

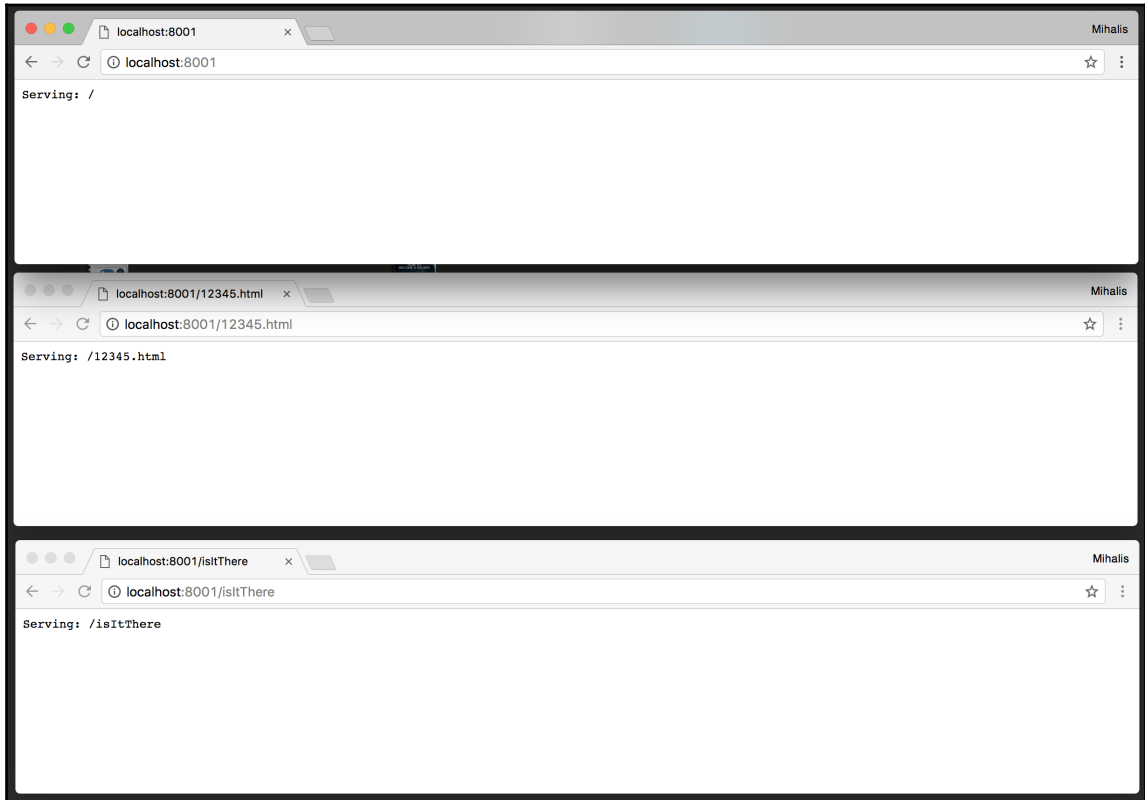


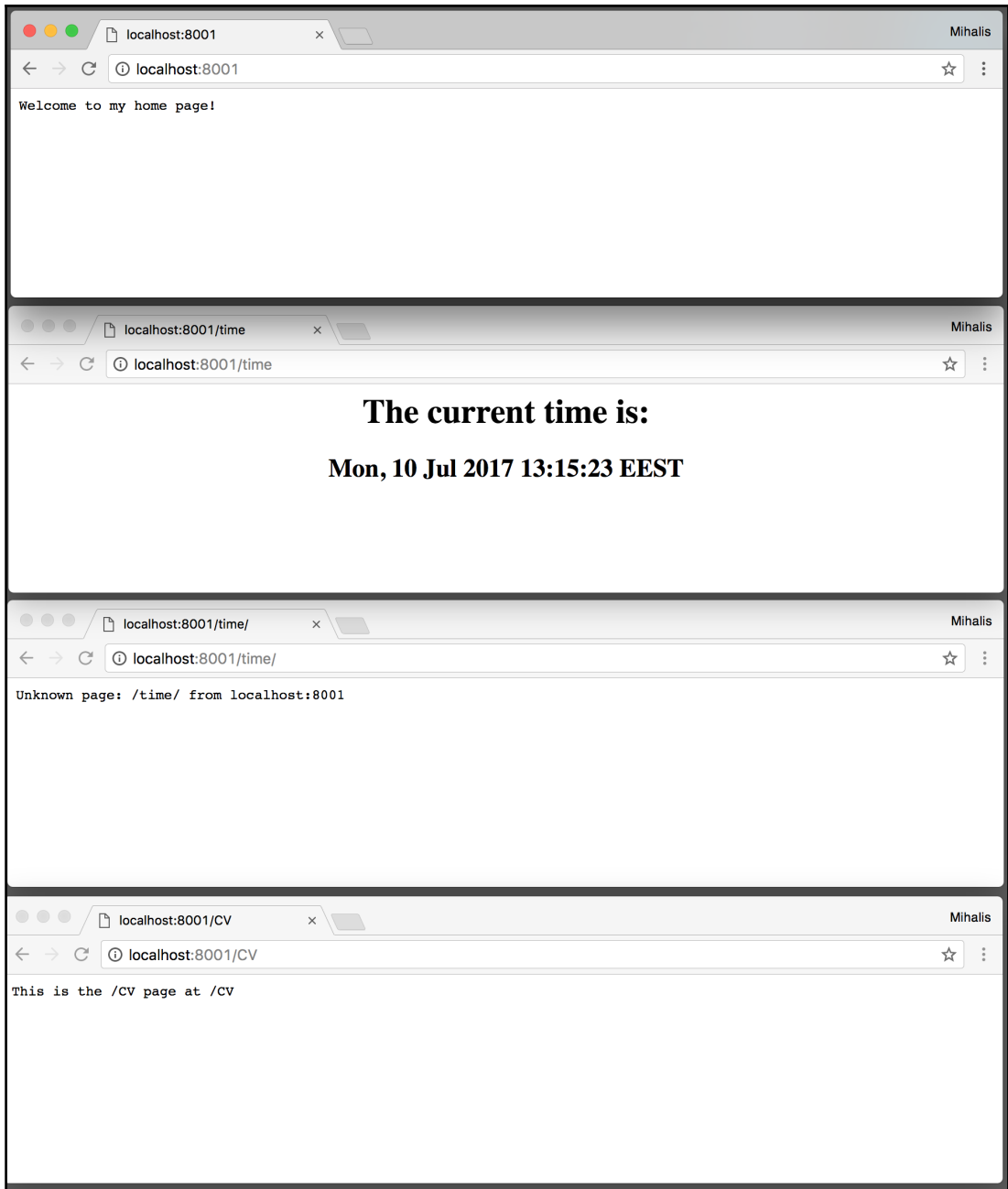


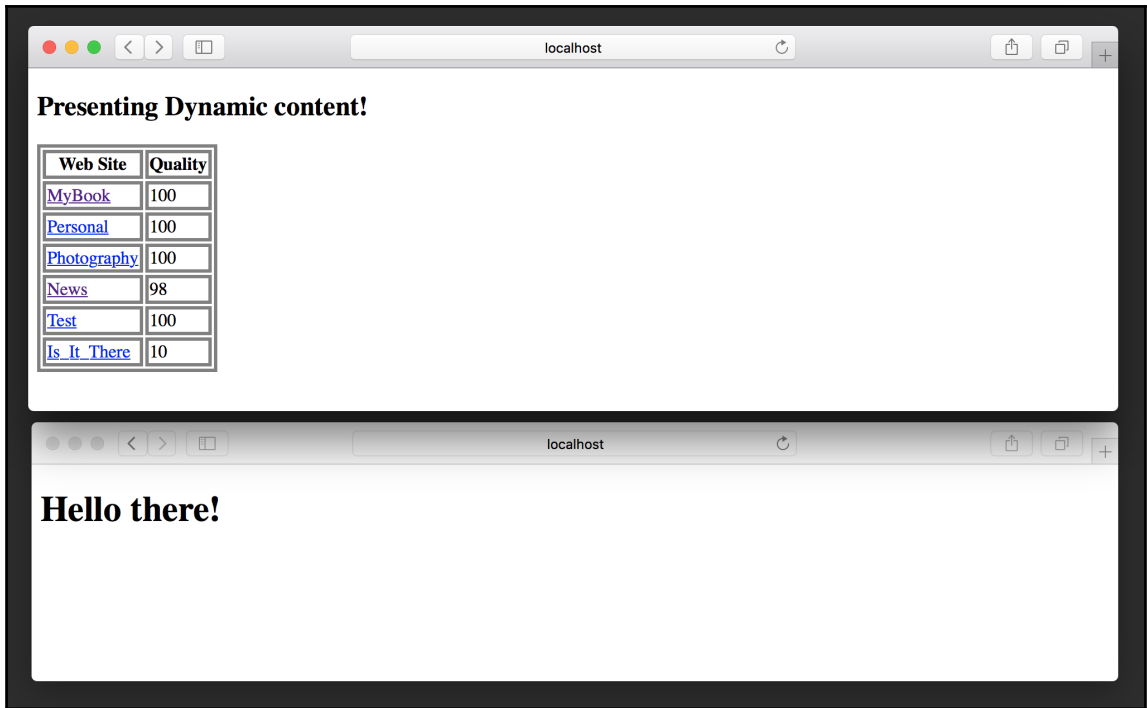
Chapter 9: Goroutines - Basic Features



Chapter 11: Writing Web Applications in Go







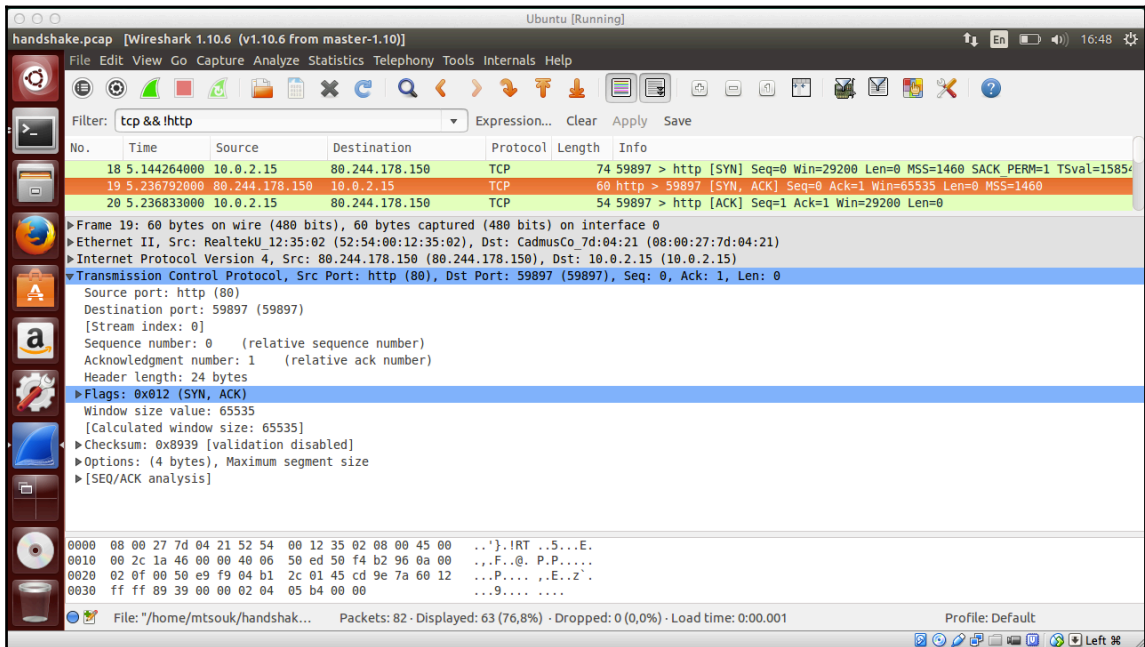
Displaying MongoDB data

localhost:8001

Displaying Dynamic MongoDB content!

| P 1 | P 2 | P 3 | P 4 | P 5 |
|-----|-----|-----|-------|-------|
| -10 | -20 | 100 | -1000 | 10000 |
| -9 | -18 | 81 | -729 | 6561 |
| -8 | -16 | 64 | -512 | 4096 |
| -7 | -14 | 49 | -343 | 2401 |
| -6 | -12 | 36 | -216 | 1296 |
| -5 | -10 | 25 | -125 | 625 |
| -4 | -8 | 16 | -64 | 256 |
| -3 | -6 | 9 | -27 | 81 |
| -2 | -4 | 4 | -8 | 16 |
| -1 | -2 | 1 | -1 | 1 |
| 1 | 2 | 1 | 1 | 1 |
| 2 | 4 | 4 | 8 | 16 |
| 3 | 6 | 9 | 27 | 81 |
| 4 | 8 | 16 | 64 | 256 |

Chapter 12: Network Programming



Index