Beautiful (and strange) I/O

Lightning Talk, Go and Cloud Native Leipzig

https://golangleipzig.space

Martin Czygan

@BasislagerCo, 2019-06-14, 19:00

Go Proverb

- The bigger the interface, the weaker the abstraction

> More of theses at https://go-proverbs.github.io/

Exemplified in package io

Generic I/O with `io.Reader` and `io.Writer` and a few other interfaces.

> https://golang.org/pkg/io/

	R	W	C	S
io.Reader	x			
io.Writer		x		
io.Closer			x	
io.Seeker				x
io.ReadWriter	x	x		
io.ReadCloser	x		x	
io.ReadSeeker	x			x
io.WriteCloser		x	x	
io.WriteSeeker		x		x
io.ReadWriteCloser	x	x	x	
io.ReadWriteSeeker	x	x		x

Missing things

Libraries might implement missing pieces, e.g.

- ReadSeekCloser, ReaderAtCloser

From: github.com/go4org/go4.

IO interface list

- `io.ReaderAt` (p, off)
- `io.ReaderFrom` (r)
- `io.WriterAt` (p, off)
- `io.WriterTo` (w)

Use cases | io.ReaderAt

- `io.ReaderAt`, `io.WriterAt` -- (parallel writes) with offset

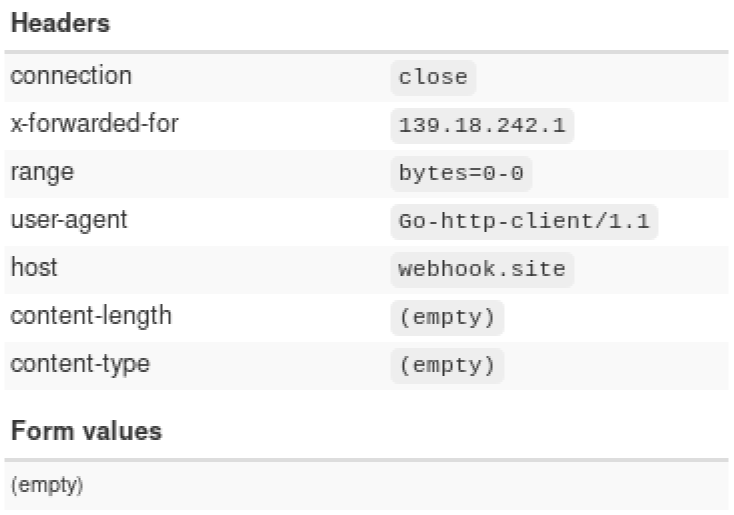
Sidenote: For filesystems, there is a pread(2) system call in Linux

> read from or write to a file descriptor at a given offset ...
> The pread() and pwrite() system calls are especially useful in **multithreaded applications**. They allow multiple threads to perform I/O on the same **same file descriptor** without being affected by changes to the file offset by other threads.

- HTTP range request example (on zip without download)
- see: `examples/rangerequest`

RFC 7233 HTTP Range Requests

> Likewise, devices with limited local storage might benefit from being able to request only a subset of a larger representation, such as a single page of a very large document, or the dimensions of an embedded image. -- https://tools.ietf.org/html/rfc7233#section-1



Headers

connection	`close`
x-forwarded-for	`139.18.242.1`
range	`bytes=0-0`
user-agent	`Go-http-client/1.1`
host	`webhook.site`
content-length	`(empty)`
content-type	`(empty)`

Form values

(empty)

Use cases | io.ReaderFrom

- Optimizing Copy

> To avoid using an intermediate buffer entirely, types can implement interfaces to read and write directly. When implemented, the Copy() function will avoid the intermediate buffer and use these implementations directly.

- maybe: io.ReaderFrom — a data structure, that know how to deserialize itself

Use cases | io.ReaderFrom

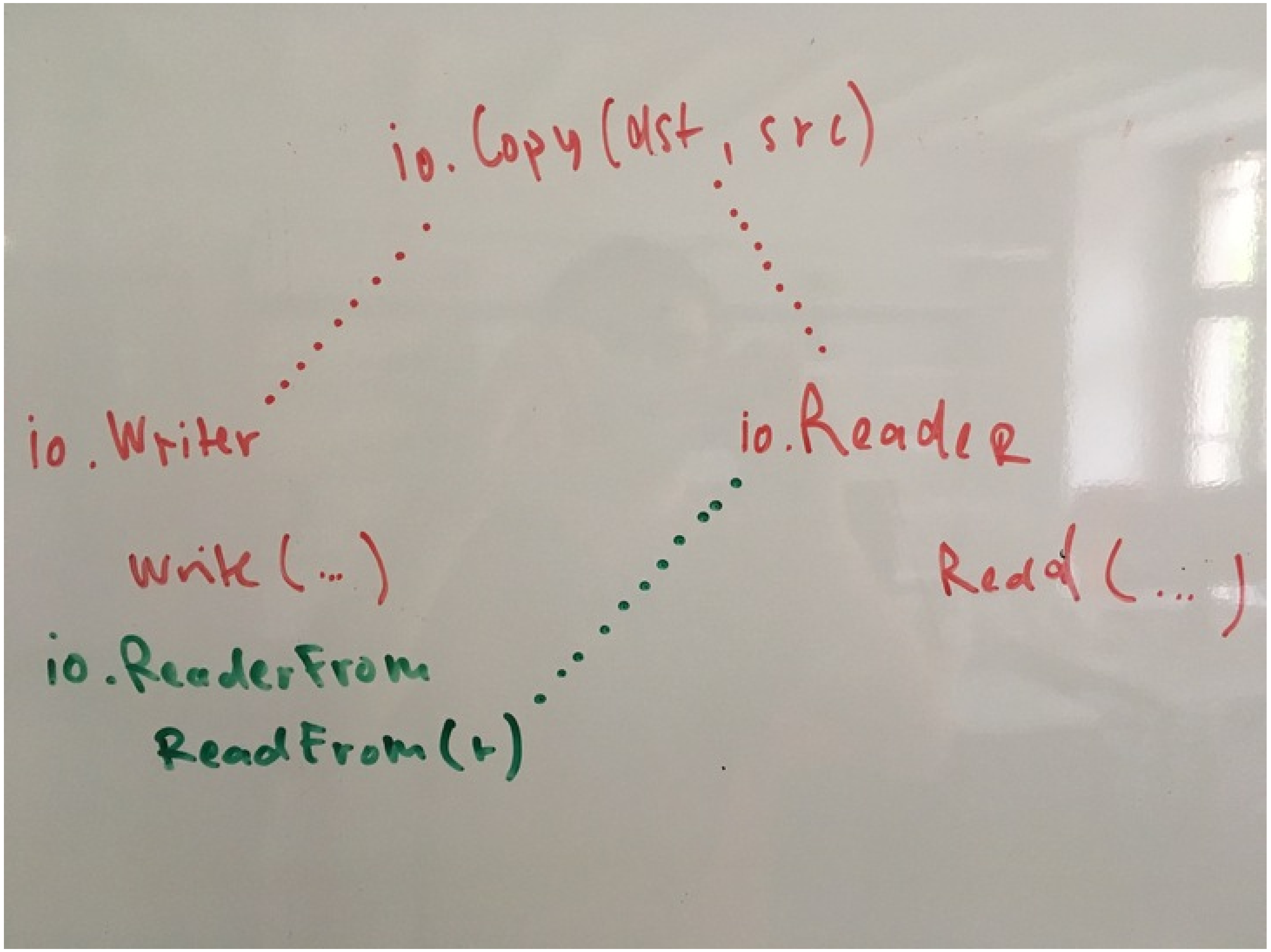
```go
// io.go, https://golang.org/src/io/io.go
// ...
// Similarly, if the writer has a ReadFrom method,
// use it to do the copy.

if rt, ok := dst.(ReaderFrom); ok {
        return rt.ReadFrom(src)
}
```

Also known as: interface upgrade.

> The zero-copy IO in Go is so elegant.

- https://news.ycombinator.com/item?id=8714051 (174, 2014)

Use cases | io.ReaderFrom



Use cases | Bad example (most likely)

> Example: different JSON API structs, but each of them implements io.ReaderFrom, so the data fetch can be separated --fetchLocation(location string, r io.ReaderFrom)

io.ReaderFrom is an optional interface

- Enabling optional optimizations/features

Readers for various types

Rune

- `io.RuneReader` (read a rune)
- `io.RuneScanner` (support for rewind)

Byte

- `io.ByteReader` (read a byte)
- `io.ByteScanner` (support for rewind)
- `io.ByteWriter`

String

- `io.StringWriter` (new in 1.12)

Who implements these interfaces?

- files, atomic files
- buffered io
- network connections
- response bodies
- compression algorithms
- hash sums
- images
- JSON and XML encoders and decoders
- utilities like counters, test data generators, stream splitters, mutli-readers, ... and more

A simple interface

Reader and Writer are single method interfaces.

```
type Reader interface {
    func Read(p []byte) (n int, err error)
}

type Writer interface {
    func Write(p []byte) (n int, err error)
}
```

Examples

Few examples for usage and custom implementations.

Empty reader and Discard

- Empty
- Discard

The standard library implementation of ioutil.Discard.

Example: multireader

Read from an arbitrary number of readers in sequence.

- MultiReader

Example: Embedding a reader

- Embedding a reader

This is also part of the Go Tour, currently in exercise methods/23.

Example: Endless stream

Generating test data.

- Endless stream

Example: Blackout

- Censoring reader

Example: stickyErrWriter

Allows to write multiple times without error checking, because the error sticks around.

- stickyErrWriter

From live hacking with Brad and Andrew.

I am a collector of implementations

If you happen to come across an interesting implementation, please let me know - E-Mail, via issue on exploreio, @cvvfj, ...

Links:

- https://golang.org/pkg/io/ (docs)
- https://www.datadoghq.com/blog/crossing-streams-love-letter-gos-io-reader/ (love letter)
- https://medium.com/go-walkthrough/go-walkthrough-io-package-8ac5e95a9fbd (walkthrough)
- https://www.youtube.com/watch?v=PAAkCSZUG1c (Go Proverbs, 2015)
- https://github.com/miku/exploreio (example implementations)