

# implementing code coverage

with

## -toolexec

Ehden Sinai



# implementing code coverage

with

## -toolexec

Ehden Sinai



```
package main

import (
    "fmt"
    "log"
    "os"
    "strconv"
)

func main() {
    n, err := strconv.Atoi(os.Args[1])
    if err != nil {
        log.Fatal("💩")
    }

    for i := 1; i <= n; i++ {
        fmt.Println(fizzbuzz(i))
    }
}

func fizzbuzz(n int) string {
    switch {
    case n%15 == 0:
        return "fizzbuzz"
    case n%5 == 0:
        return "buzz"
    case n%3 == 0:
        return "fizz"
    default:
        return strconv.Itoa(n)
    }
}
```

```
package main

import (
    "fmt"
    "log"
    "os"
    "strconv"
)

func main() {
    n, err := strconv.Atoi(os.Args[1])
    if err != nil {
        log.Fatal("😡")
    }

    for i := 1; i <= n; i++ {
        fmt.Println(fizzbuzz(i))
    }
}

func fizzbuzz(n int) string {
    switch {
    case n%15 == 0:
        return "fizzbuzz"
    case n%5 == 0:
        return "buzz"
    case n%3 == 0:
        return "fizz"
    default:
        return strconv.Itoa(n)
    }
}
```

go build -cover

go run -cover

go test -cover

go list -export -cover

```
package main; import _ "runtime/coverage"

import (
    "fmt"
    "log"
    "os"
    "strconv"
)

func main() {goCover_4bac6d588efe__0[0] = 4 ; goCover_4bac6d588efe__0[1] = goCover_4bac6d588efe_P ; goCover_4bac6d588efe__0[2] = 0 ; goCover_4bac6d588efe__0[3] = 1;
    n, err := strconv.Atoi(os.Args[1])
    if err != nil {goCover_4bac6d588efe__0[5] = 1;
        log.Fatal("💩")
    }

    goCover_4bac6d588efe__0[4] = 1;for i := 1; i <= n; i++ {goCover_4bac6d588efe__0[6] = 1;
        fmt.Println(fizzbuzz(i))
    }
}

func fizzbuzz(n int) string {goCover_4bac6d588efe__1[0] = 5 ; goCover_4bac6d588efe__1[1] = goCover_4bac6d588efe_P ; goCover_4bac6d588efe__1[2] = 1 ; goCover_4bac6d588efe__1[3] = 1;
    switch {
    case n%15 == 0:goCover_4bac6d588efe__1[4] = 1;
        return "fizzbuzz"
    case n%5 == 0:goCover_4bac6d588efe__1[5] = 1;
        return "buzz"
    case n%3 == 0:goCover_4bac6d588efe__1[6] = 1;
        return "fizz"
    default:goCover_4bac6d588efe__1[7] = 1;
        return strconv.Itoa(n)
    }
}
```

```
package main; import _ "runtime/coverage"

import (
    "fmt"
    "log"
    "os"
    "strconv"
)

func main() {goCover_4bac6d588efe__0[0] = 4 ; goCover_4bac6d588efe__0[1] = goCover_4bac6d588efe_P ; goCover_4bac6d588efe__0[2] = 0 ; goCover_4bac6d588efe__0[3] = 1;
    n, err := strconv.Atoi(os.Args[1])
    if err != nil {goCover_4bac6d588efe__0[5] = 1;
        log.Fatal("💩")
    }

    goCover_4bac6d588efe__0[4] = 1;for i := 1; i <= n; i++ {goCover_4bac6d588efe__0[6] = 1;
        fmt.Println(fizzbuzz(i))
    }
}

func fizzbuzz(n int) string {goCover_4bac6d588efe__1[0] = 5 ; goCover_4bac6d588efe__1[1] = goCover_4bac6d588efe_P ; goCover_4bac6d588efe__1[2] = 1 ; goCover_4bac6d588efe__1[3] = 1;
    switch {
    case n%15 == 0:goCover_4bac6d588efe__1[4] = 1;
        return "fizzbuzz"
    case n%5 == 0:goCover_4bac6d588efe__1[5] = 1;
        return "buzz"
    case n%3 == 0:goCover_4bac6d588efe__1[6] = 1;
        return "fizz"
    default:goCover_4bac6d588efe__1[7] = 1;
        return strconv.Itoa(n)
    }
}
```

```
package main
```

```
var goCover_4bac6d588efe_P uint32
```

```
var goCover_4bac6d588efe__0 [7]uint32
```

```
var goCover_4bac6d588efe__1 [8]uint32
```

```
var goCover_4bac6d588efe_M = [...]byte{
```

```
    0xab, 0x0, 0x0, 0x0, 0x2, 0x0, 0x0, 0x0, 0x1,  
    0x0, 0x0, 0x0, 0x1, 0x0, 0x0, 0x0, 0xa6,  
    0xab, 0xad, 0xa7, 0xfd, 0x97, 0xae, 0xbb, 0x23,  
    0xee, 0x28, 0x5f, 0x36, 0x67, 0x91, 0x7f, 0x0,  
    0x0, 0x0, 0x0, 0x5, 0x0, 0x0, 0x0, 0x2,  
    0x0, 0x0, 0x0, 0x76, 0x0, 0x0, 0x0, 0x8e,  
    0x0, 0x0, 0x0, 0x5, 0x0, 0x12, 0x65, 0x68,  
    0x64, 0x65, 0x6e, 0x2e, 0x6e, 0x65, 0x74, 0x2f,  
    0x66, 0x69, 0x7a, 0x7a, 0x62, 0x75, 0x7a, 0x7a,  
    0x4, 0x6d, 0x61, 0x69, 0x6e, 0x1e, 0x65, 0x68,  
    0x64, 0x65, 0x6e, 0x2e, 0x6e, 0x65, 0x74, 0x2f,  
    0x66, 0x69, 0x7a, 0x7a, 0x62, 0x75, 0x7a, 0x7a,  
    0x2f, 0x66, 0x69, 0x7a, 0x7a, 0x62, 0x75, 0x7a,  
    0x7a, 0x2e, 0x67, 0x6f, 0x8, 0x66, 0x69, 0x7a,  
    0x7a, 0x62, 0x75, 0x7a, 0x7a, 0x4, 0x2, 0x3,  
    0xa, 0xd, 0xc, 0x10, 0x2, 0x10, 0x2, 0x10,  
    0x1a, 0x1, 0xc, 0x10, 0xe, 0x3, 0x1, 0x10,  
    0x1a, 0x12, 0x3, 0x1, 0x0, 0x5, 0x4, 0x3,  
    0x15, 0x1d, 0x16, 0x9, 0x1, 0x17, 0x11, 0x18,  
    0x14, 0x1, 0x19, 0x10, 0x1a, 0x10, 0x1, 0x1b,  
    0x10, 0x1c, 0x10, 0x1, 0x1d, 0xa, 0x1e, 0x19,  
    0x1, 0x0,}
```

```
package main
import runtime/coverage

func init() {
    coverage.initHook()
}

var goCover_4bac6d588efe_P uint32
var goCover_4bac6d588efe__0 [7]uint32
var goCover_4bac6d588efe__1 [8]uint32
var goCover_4bac6d588efe_M = [...]byte{
    0xab, 0x0, 0x0, 0x0, 0x2, 0x0, 0x0, 0x0, 0x1,
    0x0, 0x0, 0x0, 0x1, 0x0, 0x0, 0x0, 0xa6,
    0xab, 0xad, 0xa7, 0xfd, 0x97, 0xae, 0xbb, 0x23,
    0xee, 0x28, 0x5f, 0x36, 0x67, 0x91, 0x7f, 0x0,
    // ...
}
```

---

This doesn't technically exist in go code. The compiler has some special casing to inject this into its IR. If it did exist, it would kinda look like this.



```
mode: set
ehden.net/fizzbuzz/fizzbuzz.go:10.13,12.16 2 0
ehden.net/fizzbuzz/fizzbuzz.go:12.16,14.3 1 0
ehden.net/fizzbuzz/fizzbuzz.go:16.2,16.26 1 0
ehden.net/fizzbuzz/fizzbuzz.go:16.26,18.3 1 0
ehden.net/fizzbuzz/fizzbuzz.go:21.29,22.9 1 1
ehden.net/fizzbuzz/fizzbuzz.go:23.17,24.20 1 0
ehden.net/fizzbuzz/fizzbuzz.go:25.16,26.16 1 1
ehden.net/fizzbuzz/fizzbuzz.go:27.16,28.16 1 0
ehden.net/fizzbuzz/fizzbuzz.go:29.10,30.25 1 1
```

```
mode: set
```

```
ehden.net/fizzbuzz/fizzbuzz.go:10.13,12.16 2 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:12.16,14.3 1 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:16.2,16.26 1 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:16.26,18.3 1 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:21.29,22.9 1 1
```

```
ehden.net/fizzbuzz/fizzbuzz.go:23.17,24.20 1 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:25.16,26.16 1 1
```

```
ehden.net/fizzbuzz/fizzbuzz.go:27.16,28.16 1 0
```

```
ehden.net/fizzbuzz/fizzbuzz.go:29.10,30.25 1 1
```

```
'go tool cover -html=cover.out'
```

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "log"
```

```
    "os"
```

```
    "strconv"
```

```
)
```

```
func main() {
```

```
    n, err := strconv.Atoi(os.Args[1])
```

```
    if err != nil {
```

```
        log.Fatal("💩")
```

```
    }
```

```
    for i := 1; i <= n; i++ {
```

```
        fmt.Println(fizzbuzz(i))
```

```
    }
```

```
}
```

```
func fizzbuzz(n int) string {
```

```
    switch {
```

```
    case n%15 == 0:
```

```
        return "fizzbuzz"
```

```
    case n%5 == 0:
```

```
        return "buzz"
```

```
    case n%3 == 0:
```

```
        return "fizz"
```

```
    default:
```

```
        return strconv.Itoa(n)
```

```
    }
```

```
}
```

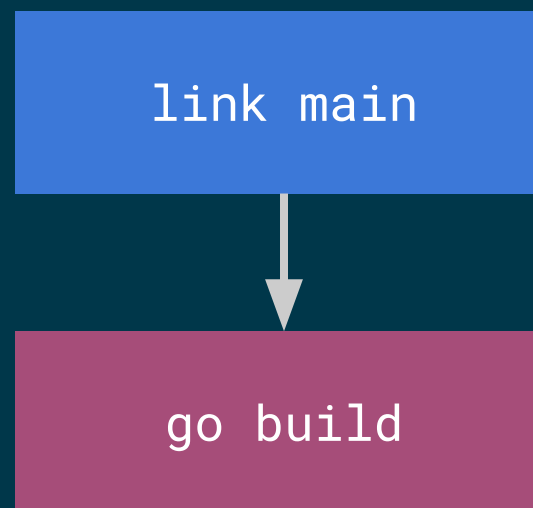
go build

link main

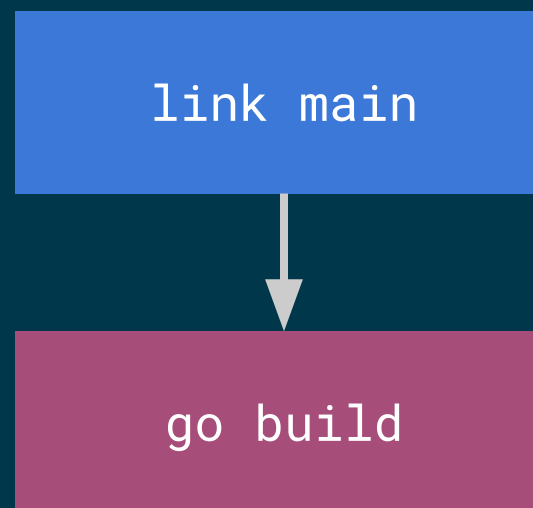


go build

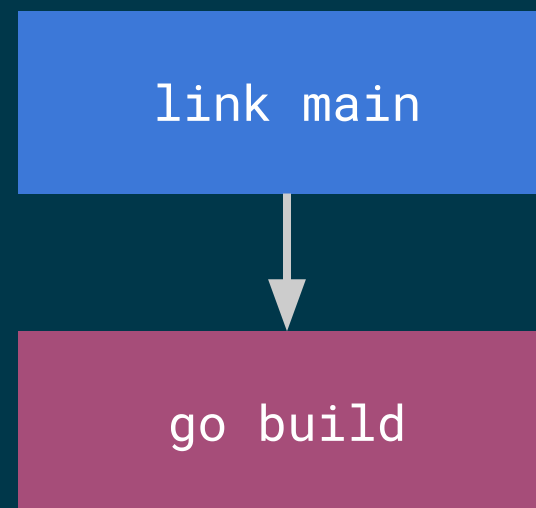
```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```



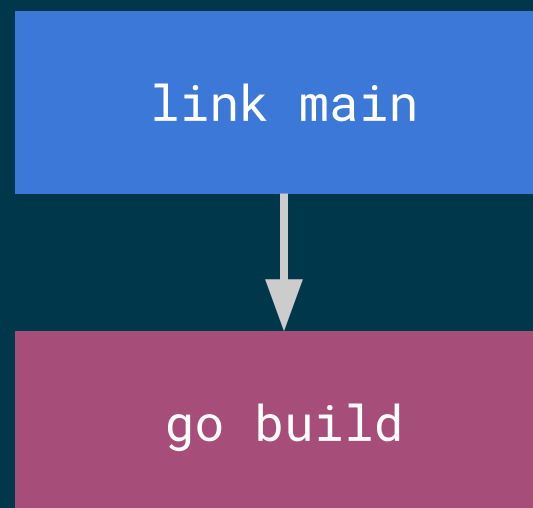
```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```



```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```



```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```





```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```

```
packagefile ehden.net/fizzbuzz=$TMPDIR/go-build3163368752/b001/_pkg_.a  
packagefile fmt=$TMPDIR/go-build3163368752/b002/_pkg_.a  
packagefile log=$TMPDIR/go-build3163368752/b046/_pkg_.a  
packagefile os=$TMPDIR/go-build3163368752/b035/_pkg_.a  
packagefile strconv=$TMPDIR/go-build3163368752/b025/_pkg_.a  
packagefile runtime=$TMPDIR/go-build3163368752/b009/_pkg_.a  
...
```

link main

go build

```
link -o $WORK/b001/exe/a.out  
-importcfg $WORK/b001/importcfg.link  
-buildmode=pie -buildid=<...> -extld=clang  
$WORK/b001/_pkg_.a
```

```
packagefile ehden.net/fizzbuzz=$TMPDIR/go-build3163368752/b001/_pkg_.a  
packagefile fmt=$TMPDIR/go-build3163368752/b002/_pkg_.a  
packagefile log=$TMPDIR/go-build3163368752/b046/_pkg_.a  
packagefile os=$TMPDIR/go-build3163368752/b035/_pkg_.a  
packagefile strconv=$TMPDIR/go-build3163368752/b025/_pkg_.a  
packagefile runtime=$TMPDIR/go-build3163368752/b009/_pkg_.a  
...
```

link main

go build

link main



go build

```
compile ehden.net/fizzbuzz
```

```
graph TD; A[compile ehden.net/fizzbuzz] --> B[link main]; B --> C[go build];
```

A vertical flowchart with three rectangular boxes. The top box is light teal and contains the text 'compile ehden.net/fizzbuzz'. A white arrow points down from the bottom center of this box to the top center of the middle box. The middle box is blue and contains the text 'link main'. Another white arrow points down from the bottom center of the middle box to the top center of the bottom box. The bottom box is maroon and contains the text 'go build'.

```
link main
```

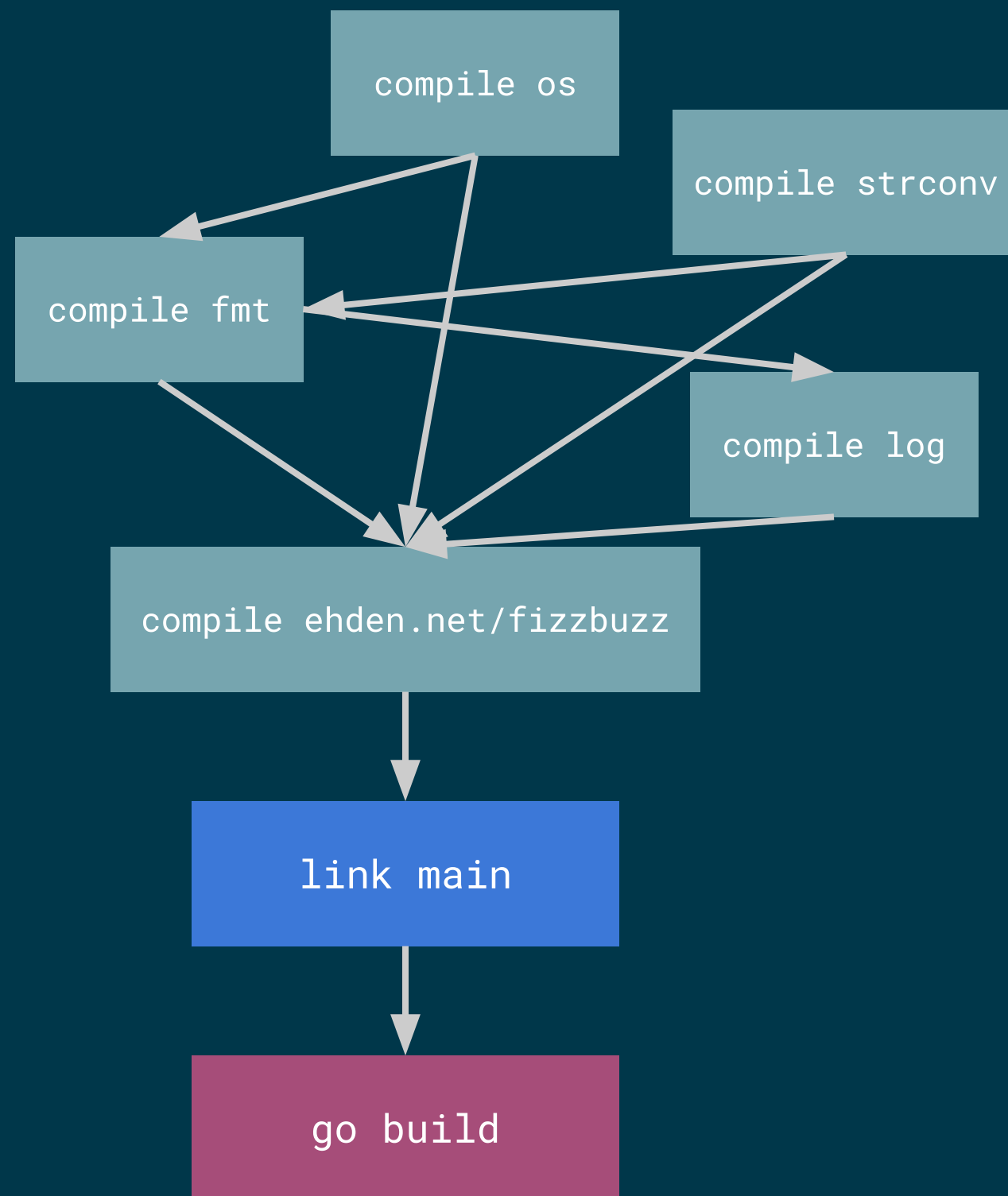
```
go build
```

```
compile -o $WORK/b001/_pkg_.a  
-importcfg $WORK/b001/importcfg  
-trimpath "$WORK/b001=>"  
-p main -lang=go1.22 -complete -buildid <...>  
-goversion go1.22.5 -c=4 -shared -nolocalimports -pack  
./fizzbuzz.go
```

compile ehden.net/fizzbuzz

link main

go build



```
compile ehden.net/fizzbuzz
```

```
graph TD; A[compile ehden.net/fizzbuzz] --> B[link main]; B --> C[go build];
```

A vertical flowchart with three rectangular boxes. The top box is light teal and contains the text 'compile ehden.net/fizzbuzz'. A white arrow points down from the bottom center of this box to the top center of the middle box. The middle box is blue and contains the text 'link main'. Another white arrow points down from the bottom center of the middle box to the top center of the bottom box. The bottom box is pink and contains the text 'go build'.

```
link main
```

```
go build
```

`cover fizzbuzz.go`



`compile ehden.net/fizzbuzz`



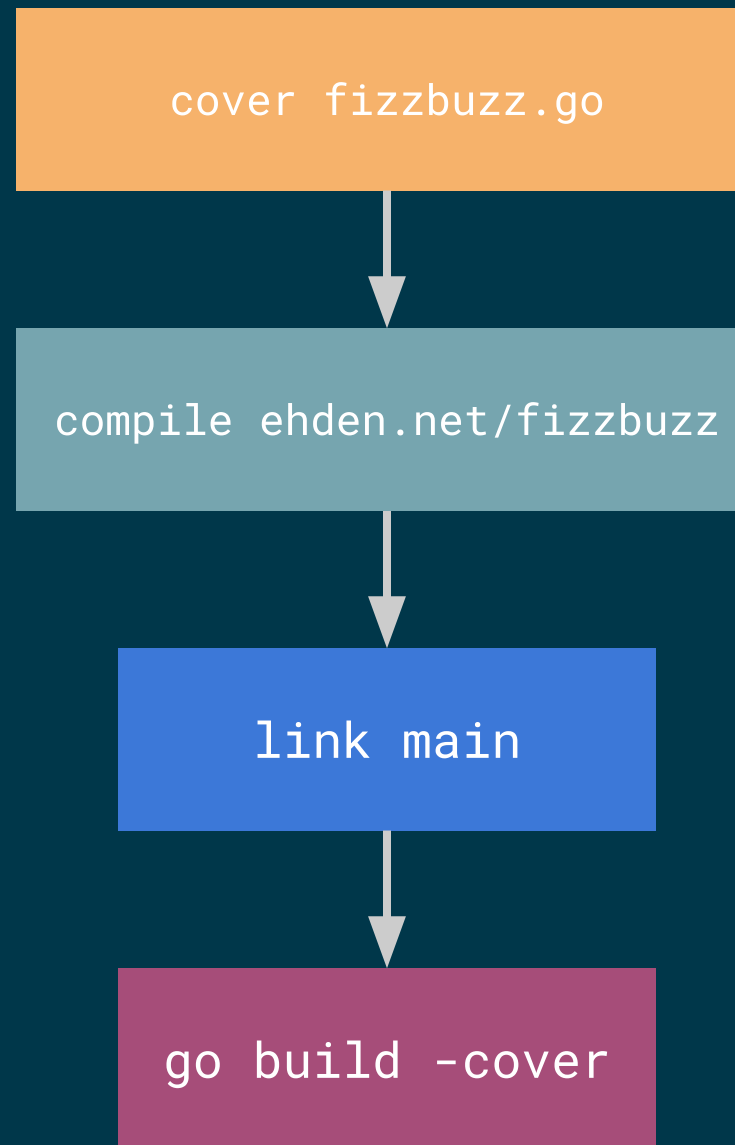
`link main`



`go build -cover`



```
cover -outfilelist $WORK/b001/coveroutfiles.txt  
-pkgcfg $WORK/b001/pkgcfg.txt -mode set  
-var goCover_4bac6d588efe_  
./fizzbuzz.go
```



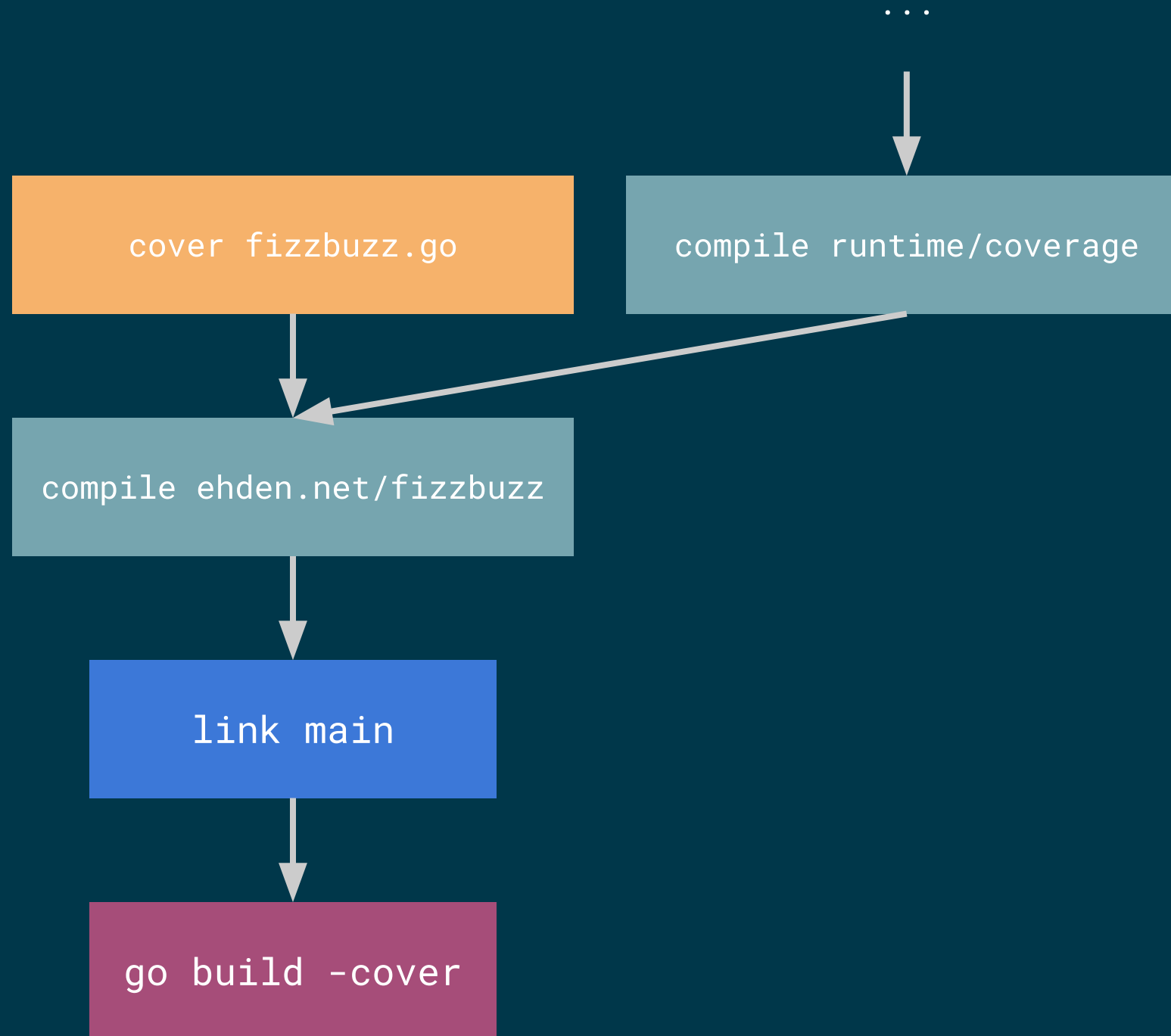
```
compile -o $WORK/b001/_pkg_.a  
-importcfg $WORK/b001/importcfg  
-coveragecfg=$WORK/b001/coveragecfg  
-trimpath "$WORK/b001=>"  
-p main -lang=go1.22 -complete -buildid <...>  
-goversion go1.22.5 -c=4 -shared -nolocalimports -pack  
$WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

cover fizzbuzz.go

compile ehden.net/fizzbuzz

link main

go build -cover



# doing this ourselves

`-toolexec 'cmd args'`

a program to use to invoke toolchain programs like vet and asm.

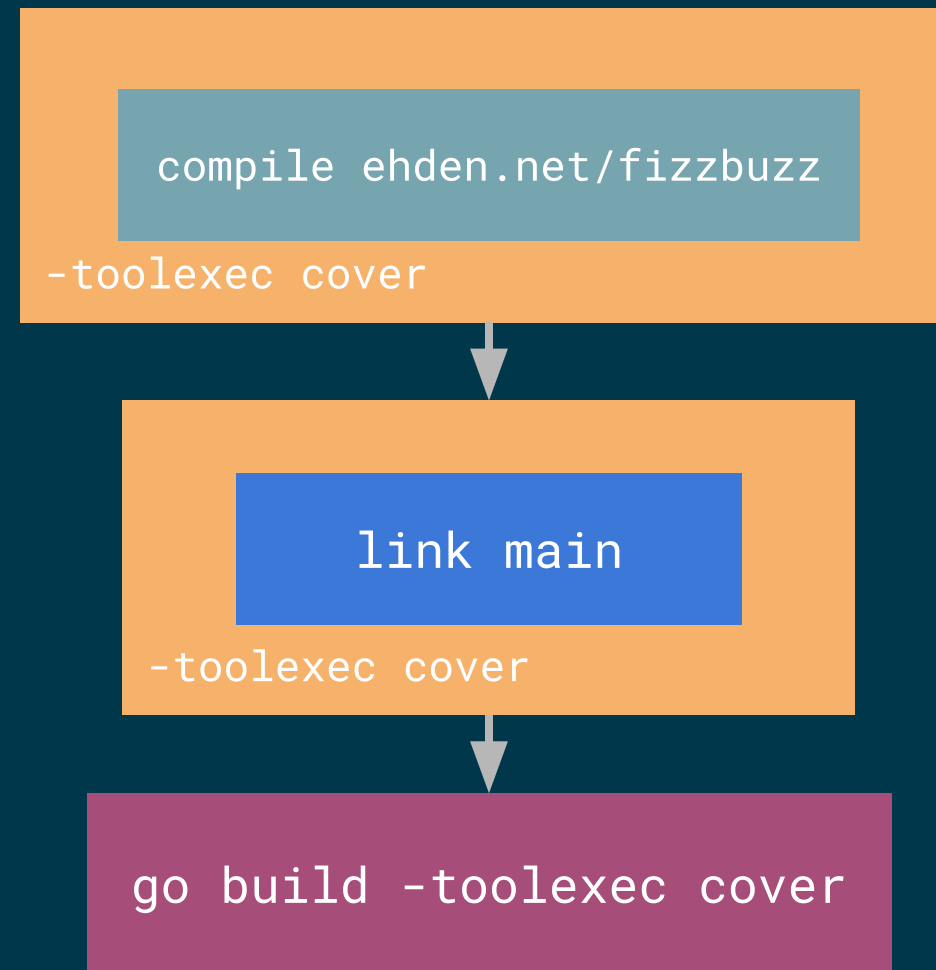
For example, instead of running asm, the go command will run

`'cmd args /path/to/asm <arguments for asm>'`.

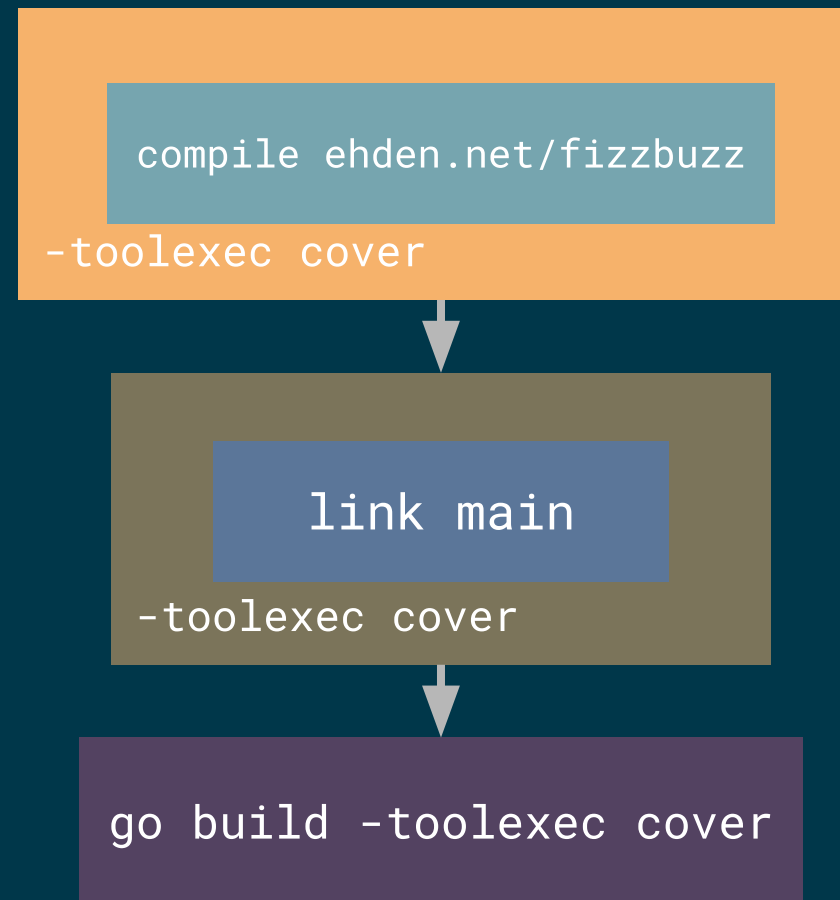
The `TOOLEXEC_IMPORTPATH` environment variable will be set,

matching `'go list -f {{.ImportPath}}'` for the package being built.

`'go help build'`



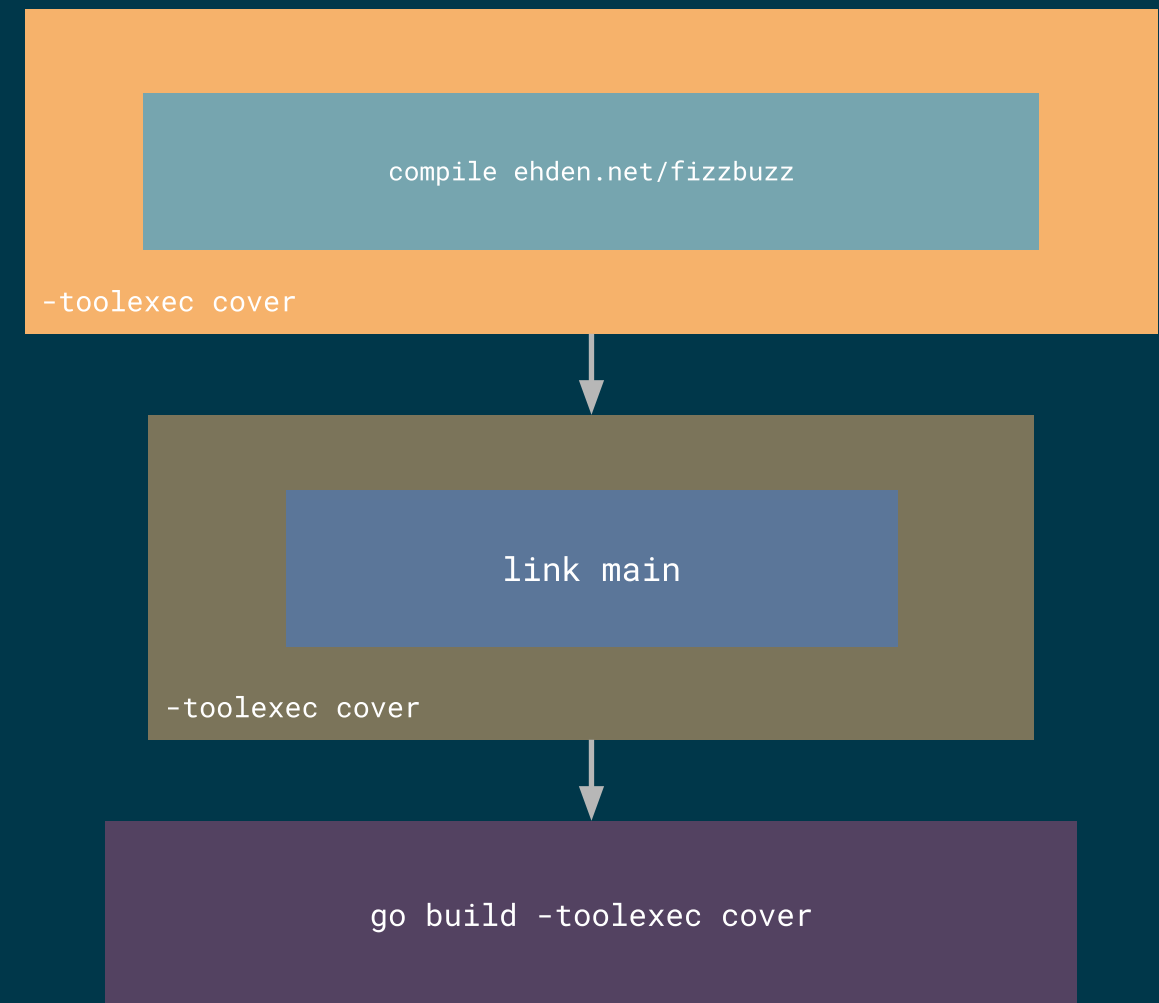
```
compile -o $WORK/b001/_pkg_.a  
-coveragecfg=$WORK/b001/coveragecfg  
-trimpath "$WORK/b001=>"  
-p main -lang=go1.22 -complete -buildid <...>  
-goversion go1.22.5 -c=4 -shared -nolocalimports -pack  
$WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```



```
compile -o $WORK/b001/_pkg_.a
  -coveragecfg=$WORK/b001/coveragecfg
  -trimpath "$WORK/b001=>"
  -p main -lang=go1.22 -complete -buildid <...>
  -goversion go1.22.5 -c=4 -shared -nolocalimports -pack
  $WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

```
func main() {defer _WriteCoverage();
  cover_73_107();n, err := strconv.Atoi(os.Args[1])
  if err != nil {
    cover_127_144();log.Fatal("😡")
  }
  for i := 1; i <= n; i++ {
    cover_178_202();fmt.Println(fizzbuzz(i))
  }
}
```

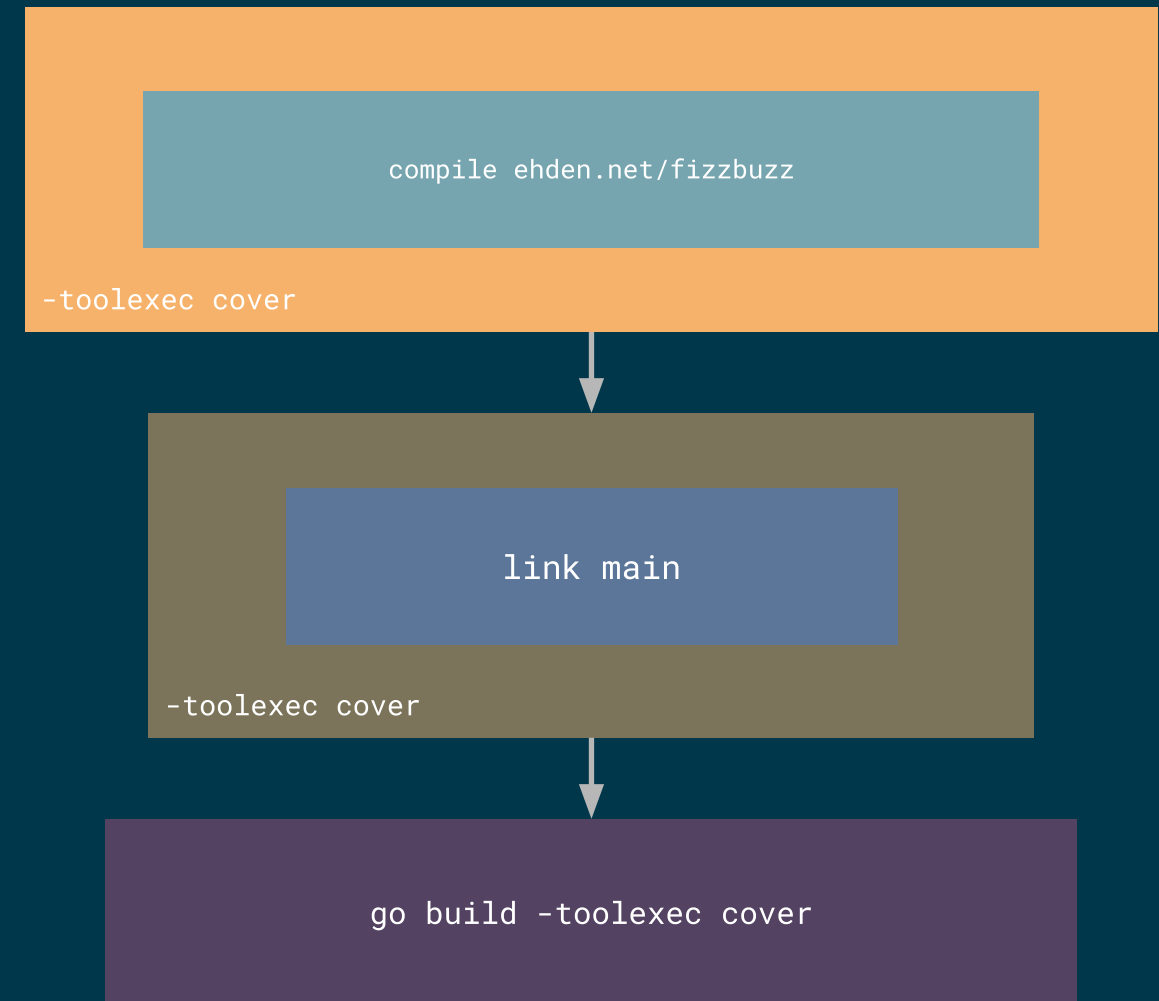
```
func fizzbuzz(n int) string {
  switch {
  case n%15 == 0:
    cover_268_285();return "fizzbuzz"
  case n%5 == 0:
    cover_304_317();return "buzz"
  case n%3 == 0:
    cover_336_349();return "fizz"
  default:
    cover_362_384();return strconv.Itoa(n)
  }
}
```



```
compile -o $WORK/b001/_pkg_.a
  -coveragecfg=$WORK/b001/coveragecfg
  -trimpath "$WORK/b001=>"
  -p main -lang=go1.22 -complete -buildid <...>
  -goversion go1.22.5 -c=4 -shared -nolocalimports -pack
  $WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

```
func main() {defer _WriteCoverage();
cover_73_107();n, err := strconv.Atoi(os.Args[1])
if err != nil {
    cover_127_144();log.Fatal("😡")
}
for i := 1; i <= n; i++ {
    cover_178_202();fmt.Println(fizzbuzz(i))
}
}
```

```
func fizzbuzz(n int) string {
switch {
case n%15 == 0:
    cover_268_285();return "fizzbuzz"
case n%5 == 0:
    cover_304_317();return "buzz"
case n%3 == 0:
    cover_336_349();return "fizz"
default:
    cover_362_384();return strconv.Itoa(n)
}
}
```

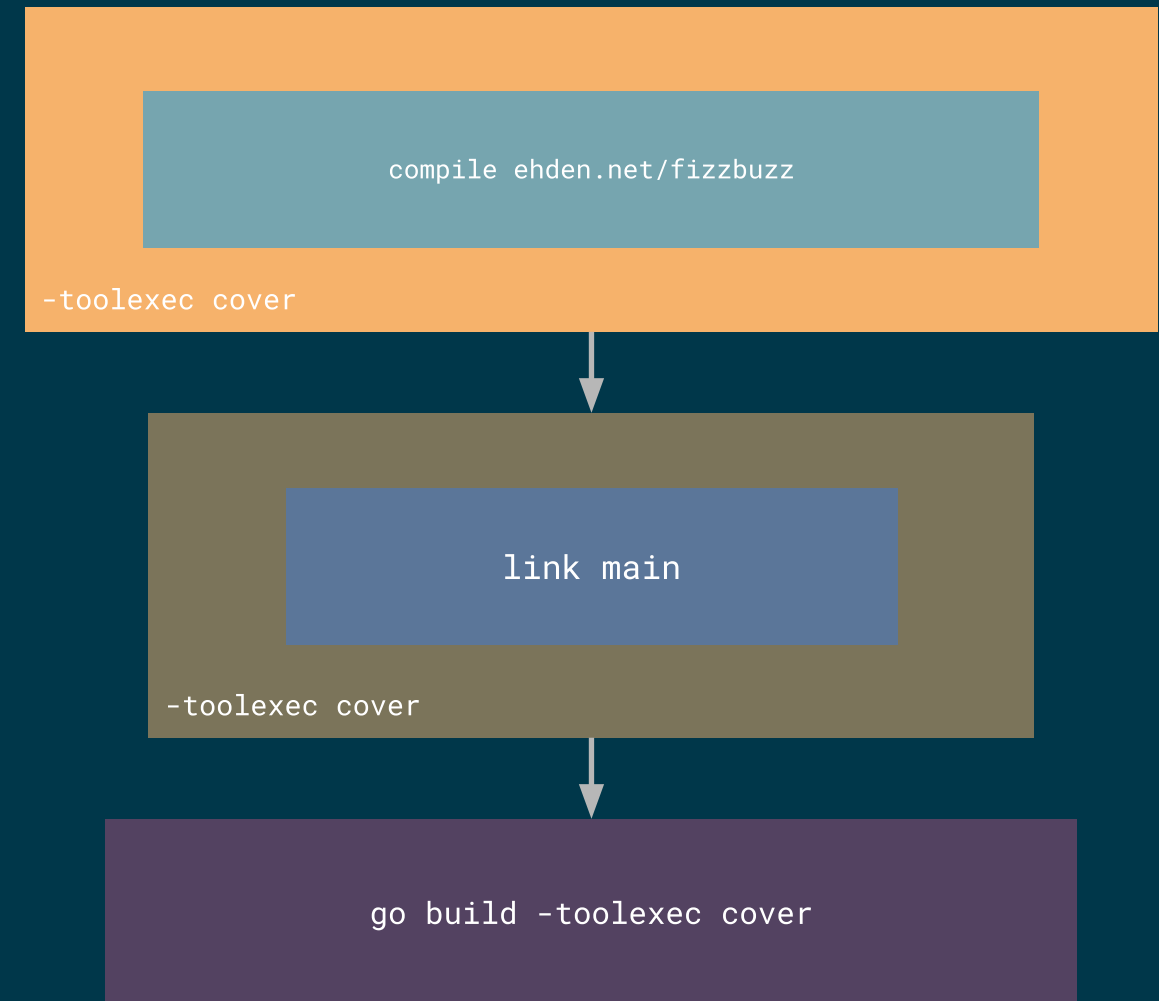




```
compile -o $WORK/b001/_pkg_.a
  -coveragecfg=$WORK/b001/coveragecfg
  -trimpath "$WORK/b001=>"
  -p main -lang=go1.22 -complete -builddid <...>
  -goversion go1.22.5 -c=4 -shared -nolocalimports -pack
  $WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

```
func main() {defer _WriteCoverage();
cover_73_107();n, err := strconv.Atoi(os.Args[1])
if err != nil {
    cover_127_144();log.Fatal("😡")
}
for i := 1; i <= n; i++ {
    cover_178_202();fmt.Println(fizzbuzz(i))
}
}
```

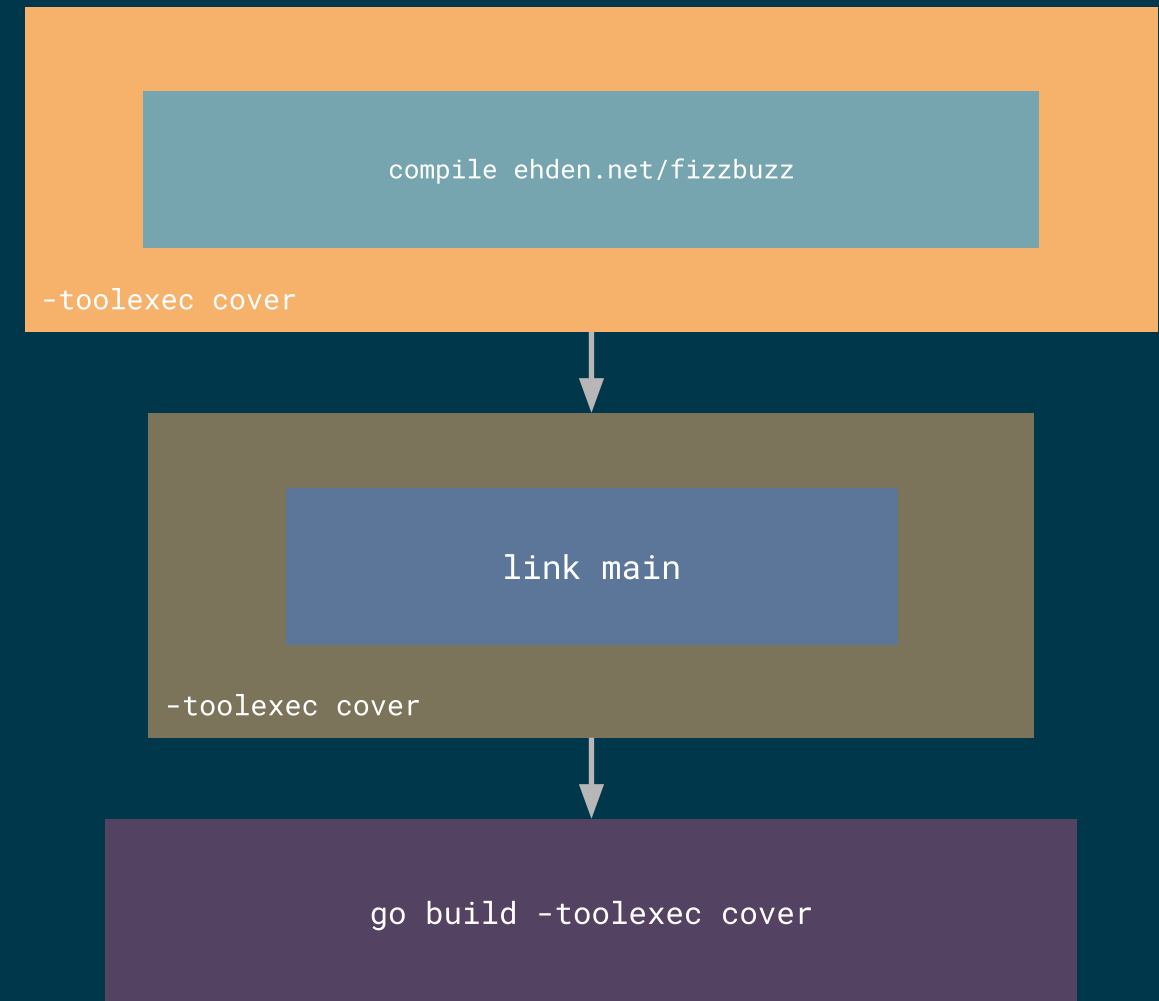
```
func fizzbuzz(n int) string {
switch {
case n%15 == 0:
    cover_268_285();return "fizzbuzz"
case n%5 == 0:
    cover_304_317();return "buzz"
case n%3 == 0:
    cover_336_349();return "fizz"
default:
    cover_362_384();return strconv.Itoa(n)
}
}
```



```
compile -o $WORK/b001/_pkg_.a
  -coveragecfg=$WORK/b001/coveragecfg
  -trimpath "$WORK/b001=>"
  -p main -lang=go1.22 -complete -builddid <...>
  -goversion go1.22.5 -c=4 -shared -nolocalimports -pack
  $WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

```
func main() {defer _WriteCoverage();
  cover_73_107();n, err := strconv.Atoi(os.Args[1])
  if err != nil {
    cover_127_144();log.Fatal("💩")
  }
  for i := 1; i <= n; i++ {
    cover_178_202();fmt.Println(fizzbuzz(i))
  }
}
```

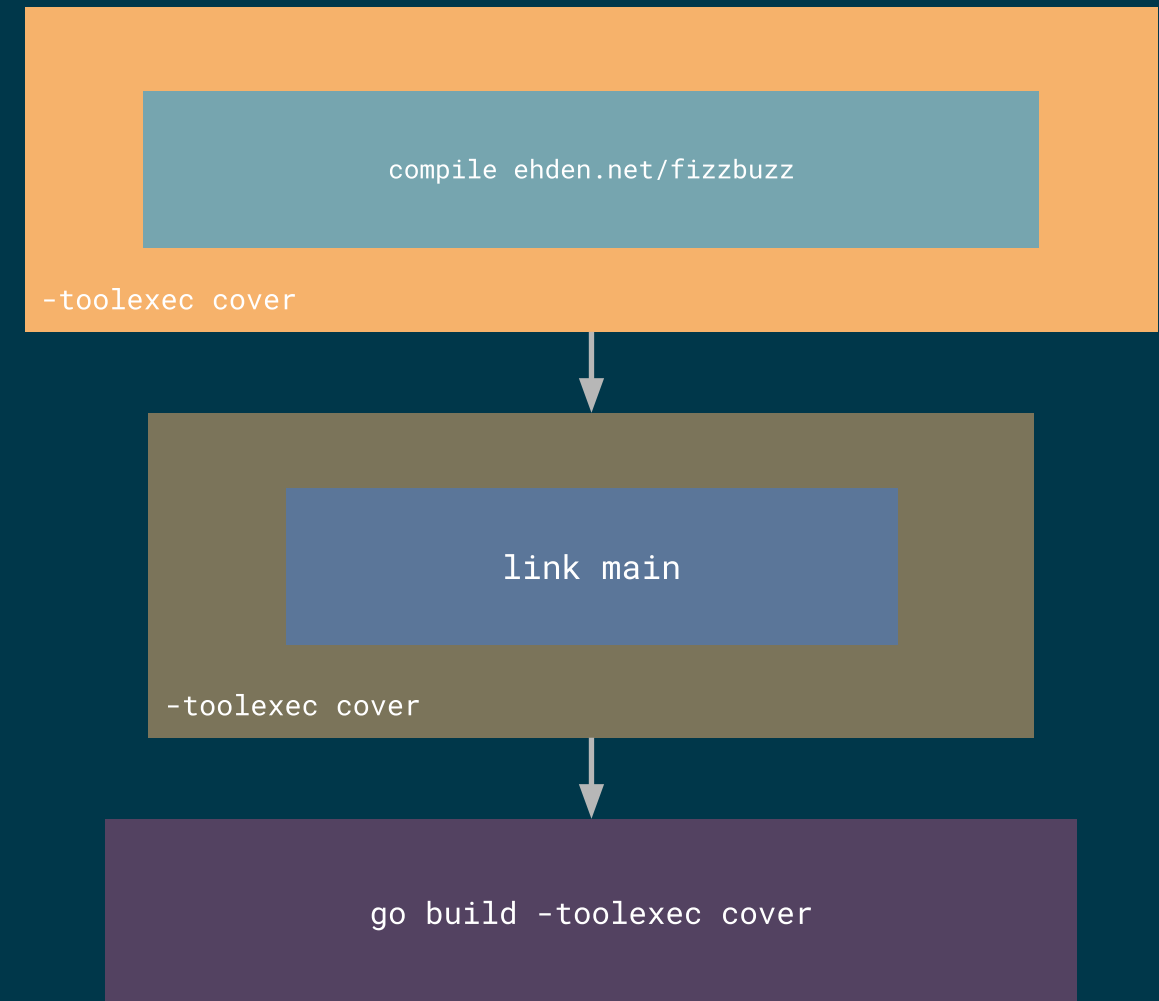
```
func fizzbuzz(n int) string {
  switch {
  case n%15 == 0:
    cover_268_285();return "fizzbuzz"
  case n%5 == 0:
    cover_304_317();return "buzz"
  case n%3 == 0:
    cover_336_349();return "fizz"
  default:
    cover_362_384();return strconv.Itoa(n)
  }
}
```



```
compile -o $WORK/b001/_pkg_.a
  -coveragecfg=$WORK/b001/coveragecfg
  -trimpath "$WORK/b001=>"
  -p main -lang=go1.22 -complete -builddid <...>
  -goversion go1.22.5 -c=4 -shared -nolocalimports -pack
  $WORK/b001/fizzbuzz.cover.go $WORK/b001/covervars.go
```

```
func main() {defer _WriteCoverage();
  cover_73_107();n, err := strconv.Atoi(os.Args[1])
  if err != nil {
    cover_127_144();log.Fatal("💩")
  }
  for i := 1; i <= n; i++ {
    cover_178_202();fmt.Println(fizzbuzz(i))
  }
}
```

```
func fizzbuzz(n int) string {
  switch {
  case n%15 == 0:
    cover_268_285();return "fizzbuzz"
  case n%5 == 0:
    cover_304_317();return "buzz"
  case n%3 == 0:
    cover_336_349();return "fizz"
  default:
    cover_362_384();return strconv.Itoa(n)
  }
}
```



`$WORK/b001/covervars.go`

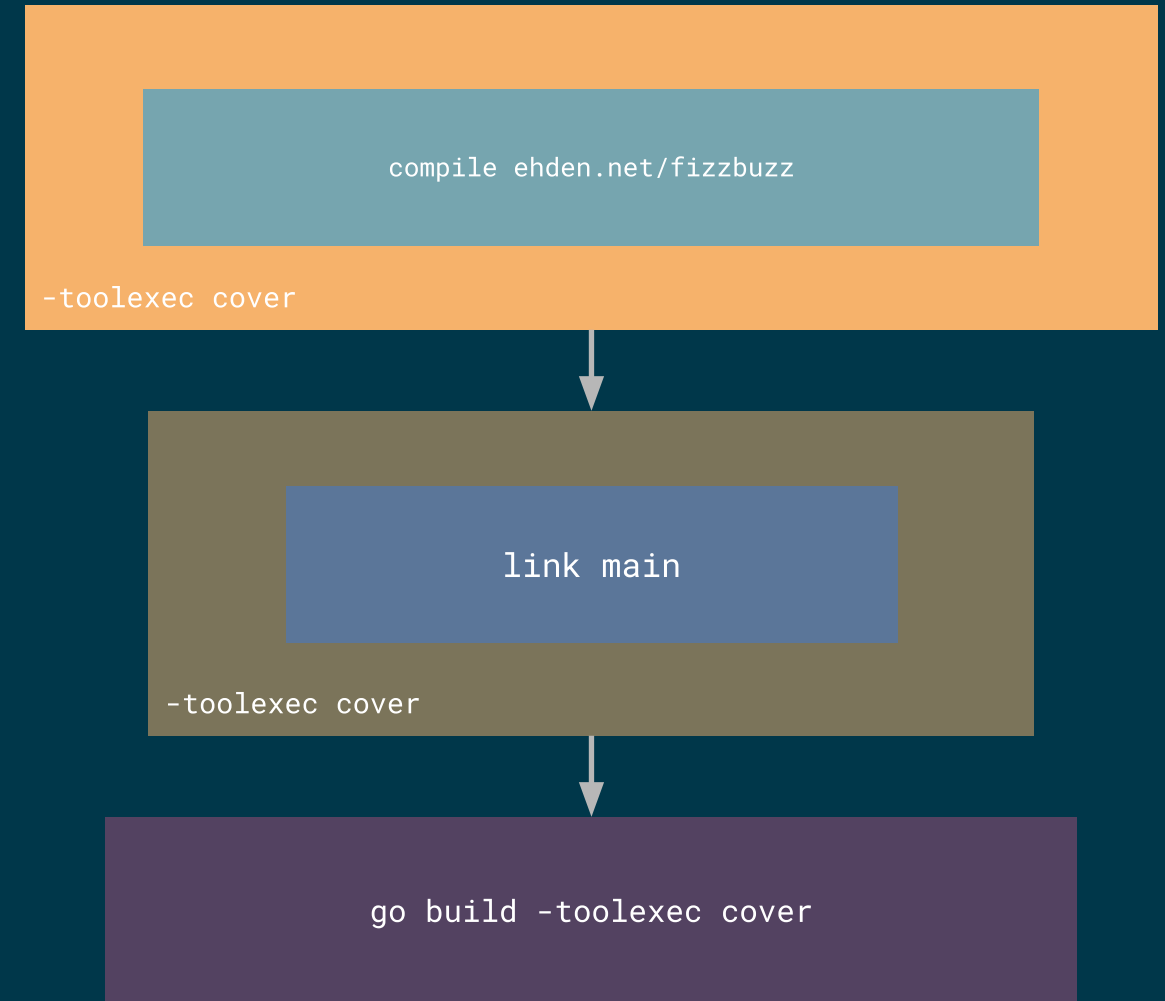
```
package main

import _ "unsafe"

//go:linkname cover_73_107 ehden.net/cover/vars.cover_73_107_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_127_144 ehden.net/cover/vars.cover_127_144_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_178_202 ehden.net/cover/vars.cover_178_202_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_268_285 ehden.net/cover/vars.cover_268_285_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_304_317 ehden.net/cover/vars.cover_304_317_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_336_349 ehden.net/cover/vars.cover_336_349_XRA0BjRi2VWNrDxFsPct
//go:linkname cover_362_384 ehden.net/cover/vars.cover_362_384_XRA0BjRi2VWNrDxFsPct

// ehden.net/fizzbuzz/fizzbuzz.go:11.2,11.36
func cover_73_107()
// ehden.net/fizzbuzz/fizzbuzz.go:13.3,13.20
func cover_127_144()
// ehden.net/fizzbuzz/fizzbuzz.go:17.3,17.27
func cover_178_202()
// ehden.net/fizzbuzz/fizzbuzz.go:24.3,24.20
func cover_268_285()
// ehden.net/fizzbuzz/fizzbuzz.go:26.3,26.16
func cover_304_317()
// ehden.net/fizzbuzz/fizzbuzz.go:28.3,28.16
func cover_336_349()
// ehden.net/fizzbuzz/fizzbuzz.go:30.3,30.25
func cover_362_384()

//go:linkname _WriteCoverage ehden.net/cover/vars.WriteCoverage
func _WriteCoverage()
```



```
package vars
```

```
import _ "unsafe"
```

```
import "os"
```

```
// ehden.net/fizzbuzz/fizzbuzz.go:11.2,11.36
```

```
var _cover_73_107_XRA0BjRi2VWnrDxFsPct uint8
```

```
func cover_73_107_XRA0BjRi2VWnrDxFsPct() { _cover_73_107_XRA0BjRi2VWnrDxFsPct = 1 }
```

```
var _cover_127_144_XRA0BjRi2VWnrDxFsPct uint8
```

```
func cover_127_144_XRA0BjRi2VWnrDxFsPct() { _cover_127_144_XRA0BjRi2VWnrDxFsPct = 1 }
```

```
var _cover_178_202_XRA0BjRi2VWnrDxFsPct uint8
```

```
func cover_178_202_XRA0BjRi2VWnrDxFsPct() { _cover_178_202_XRA0BjRi2VWnrDxFsPct = 1 }
```

```
// ...
```

```
func WriteCoverage() {
```

```
    outPath := "cover.out"
```

```
    f, _ := os.Create(outPath)
```

```
    defer f.Close()
```

```
    f.WriteString("mode: set\n")
```

```
    f.WriteString("ehden.net/fizzbuzz/fizzbuzz.go:11.2,11.36" + " 1 " + stringFor(_cover_73_107_XRA0BjRi2VWnrDxFsPct) + "\n")
```

```
    f.WriteString("ehden.net/fizzbuzz/fizzbuzz.go:13.3,13.20" + " 1 " + stringFor(_cover_127_144_XRA0BjRi2VWnrDxFsPct) + "\n")
```

```
    // ...
```

```
}
```

```
package vars
```

```
import _ "unsafe"
```

```
import "os"
```

```
// ehden.net/fizzbuzz/fizzbuzz.go:11.2,11.36
```

```
var _cover_73_107_XRA0BjRi2VWNRDxFsPct uint8
```

```
func cover_73_107_XRA0BjRi2VWNRDxFsPct() { _cover_73_107_XRA0BjRi2VWNRDxFsPct = 1 }
```

```
var _cover_127_144_XRA0BjRi2VWNRDxFsPct uint8
```

```
func cover_127_144_XRA0BjRi2VWNRDxFsPct() { _cover_127_144_XRA0BjRi2VWNRDxFsPct = 1 }
```

```
var _cover_178_202_XRA0BjRi2VWNRDxFsPct uint8
```

```
func cover_178_202_XRA0BjRi2VWNRDxFsPct() { _cover_178_202_XRA0BjRi2VWNRDxFsPct = 1 }
```

```
// ...
```

```
func WriteCoverage() {
```

```
    outPath := "cover.out"
```

```
    f, _ := os.Create(outPath)
```

```
    defer f.Close()
```

```
    f.WriteString("mode: set\n")
```

```
    f.WriteString("ehden.net/fizzbuzz/fizzbuzz.go:11.2,11.36" + " 1 " + stringFor(_cover_73_107_XRA0BjRi2VWNRDxFsPct) + "\n")
```

```
    f.WriteString("ehden.net/fizzbuzz/fizzbuzz.go:13.3,13.20" + " 1 " + stringFor(_cover_127_144_XRA0BjRi2VWNRDxFsPct) + "\n")
```

```
    // ...
```

```
}
```

# problem

- we need to compile this package somehow

# problem

- we need to compile this package somehow
  - ... into main

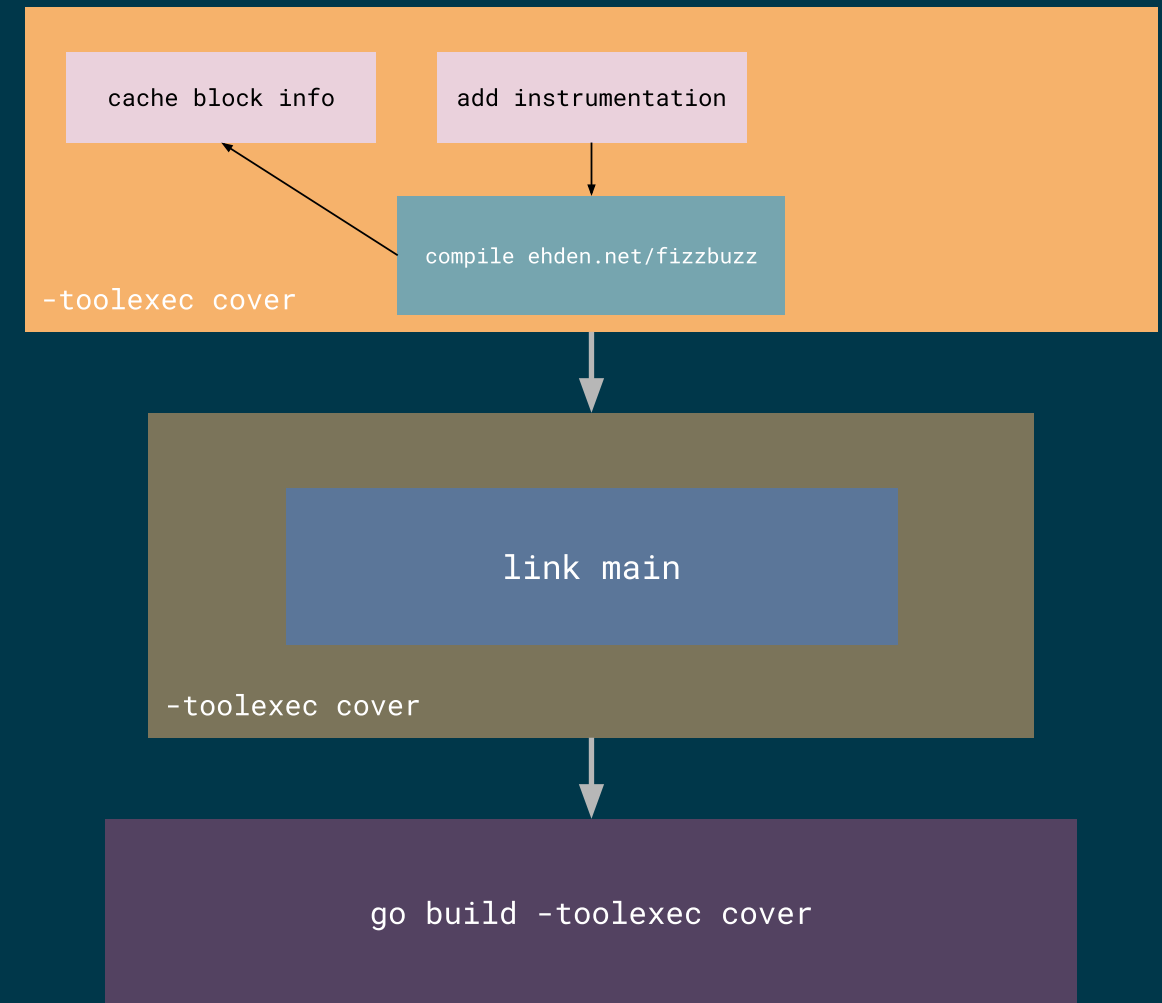


# problem

- we need to compile this package somehow
  - ... into main
    - but it needs to be compiled after main
  - ... woops

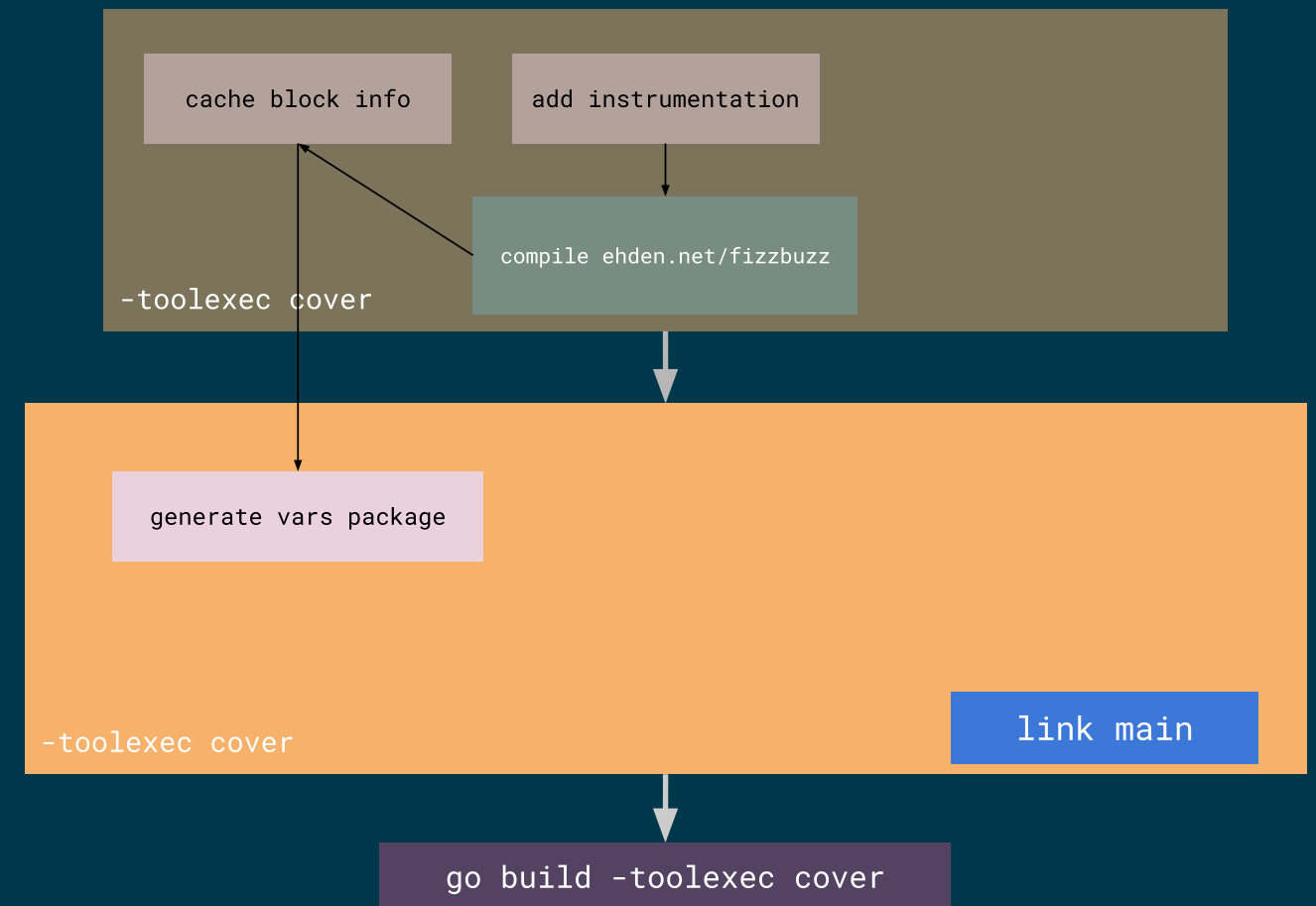
# ~~the plan~~ hold my beer

1. cache block info while we compile each package



# ~~the plan~~ hold my beer

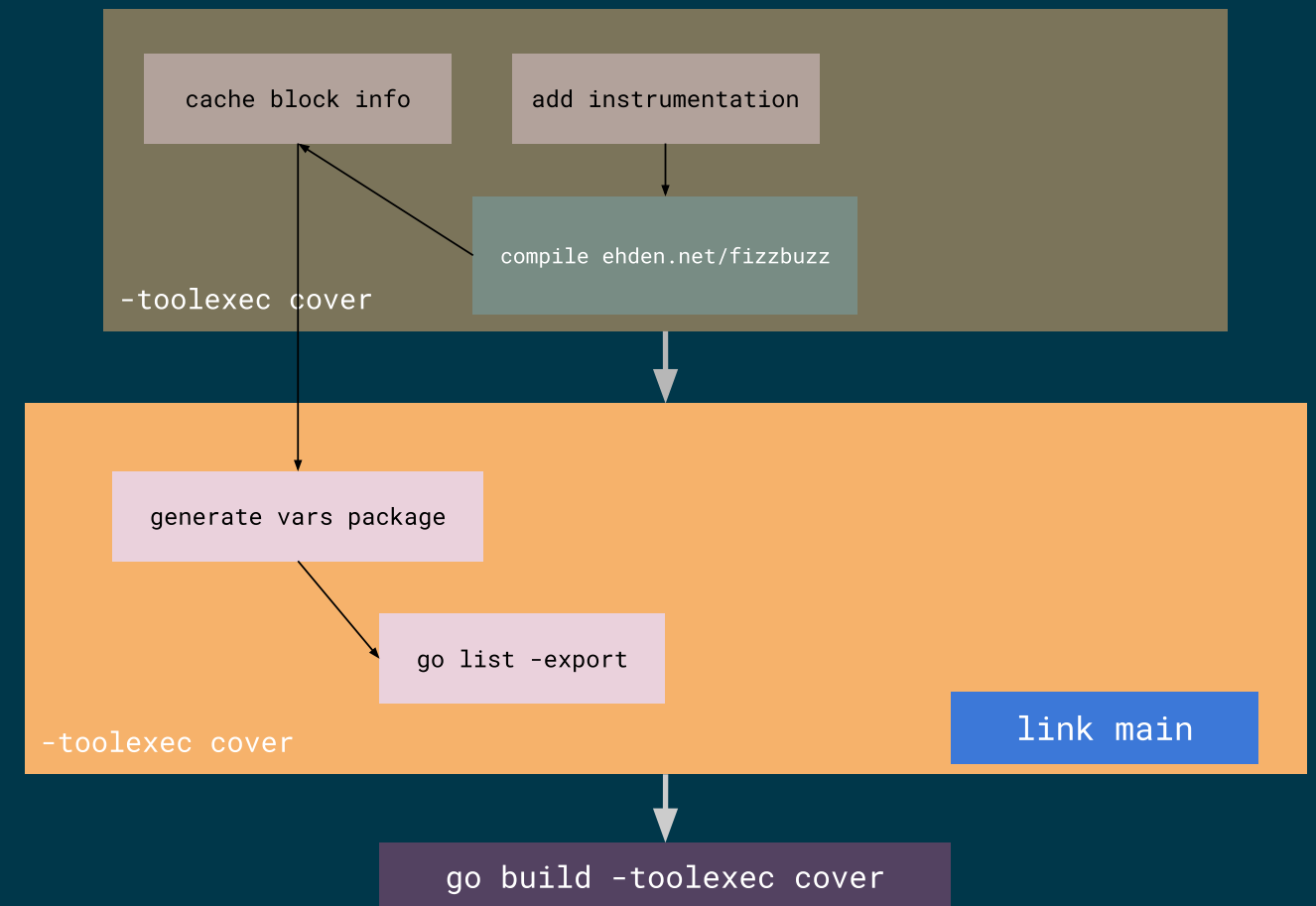
1. cache block info while we compile each package
2. generate package code with `importcfg.link`



# ~~the plan~~ hold my beer

1. cache block info while we compile each package
2. generate package code with `importcfg.link`
3. build our generated package

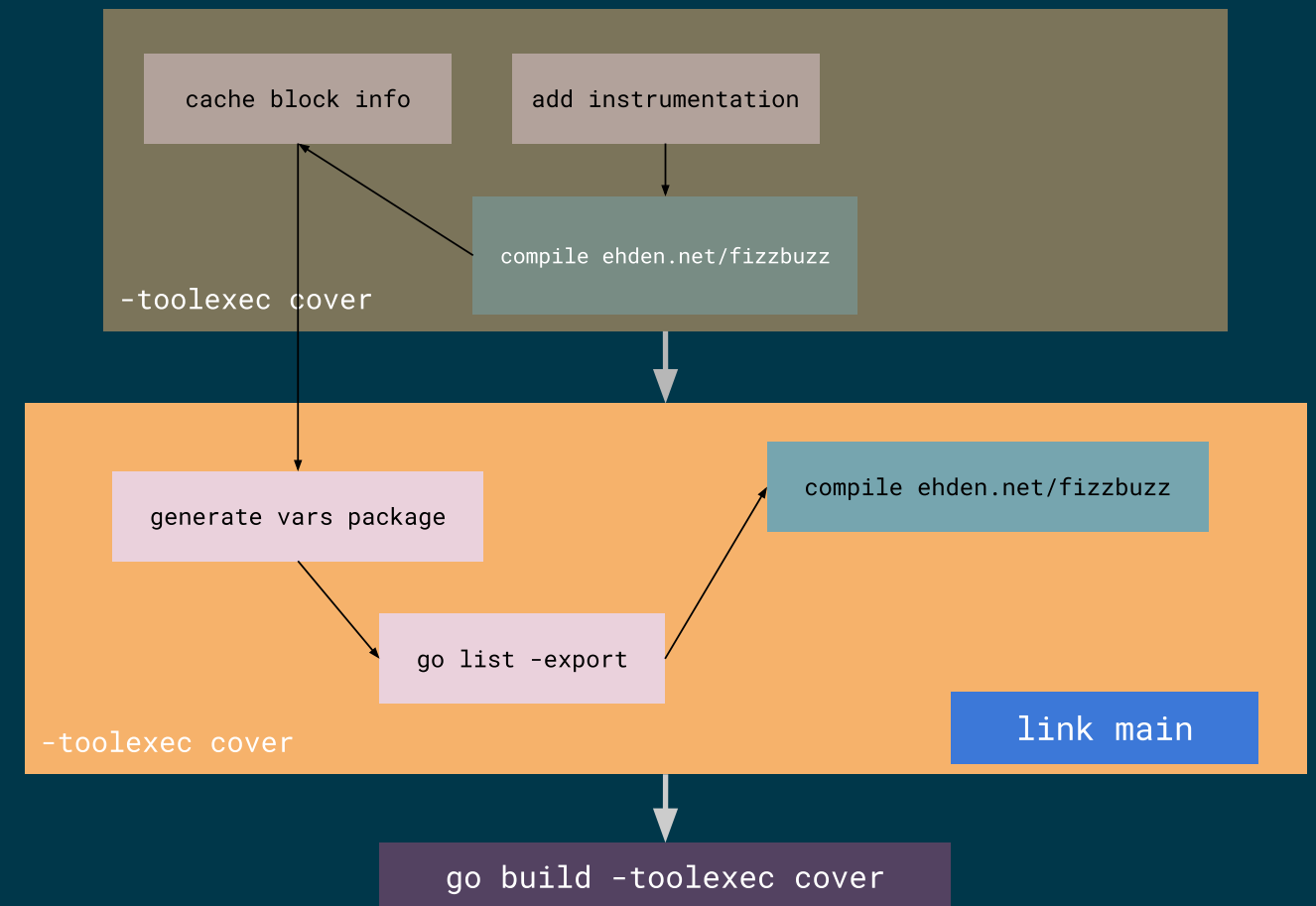
```
go list -toolexec cover -export -f {{ .Export }}
```



# ~~the plan~~ hold my beer

1. cache block info while we compile each package
2. generate package code with `importcfg.link`
3. build our generated package  

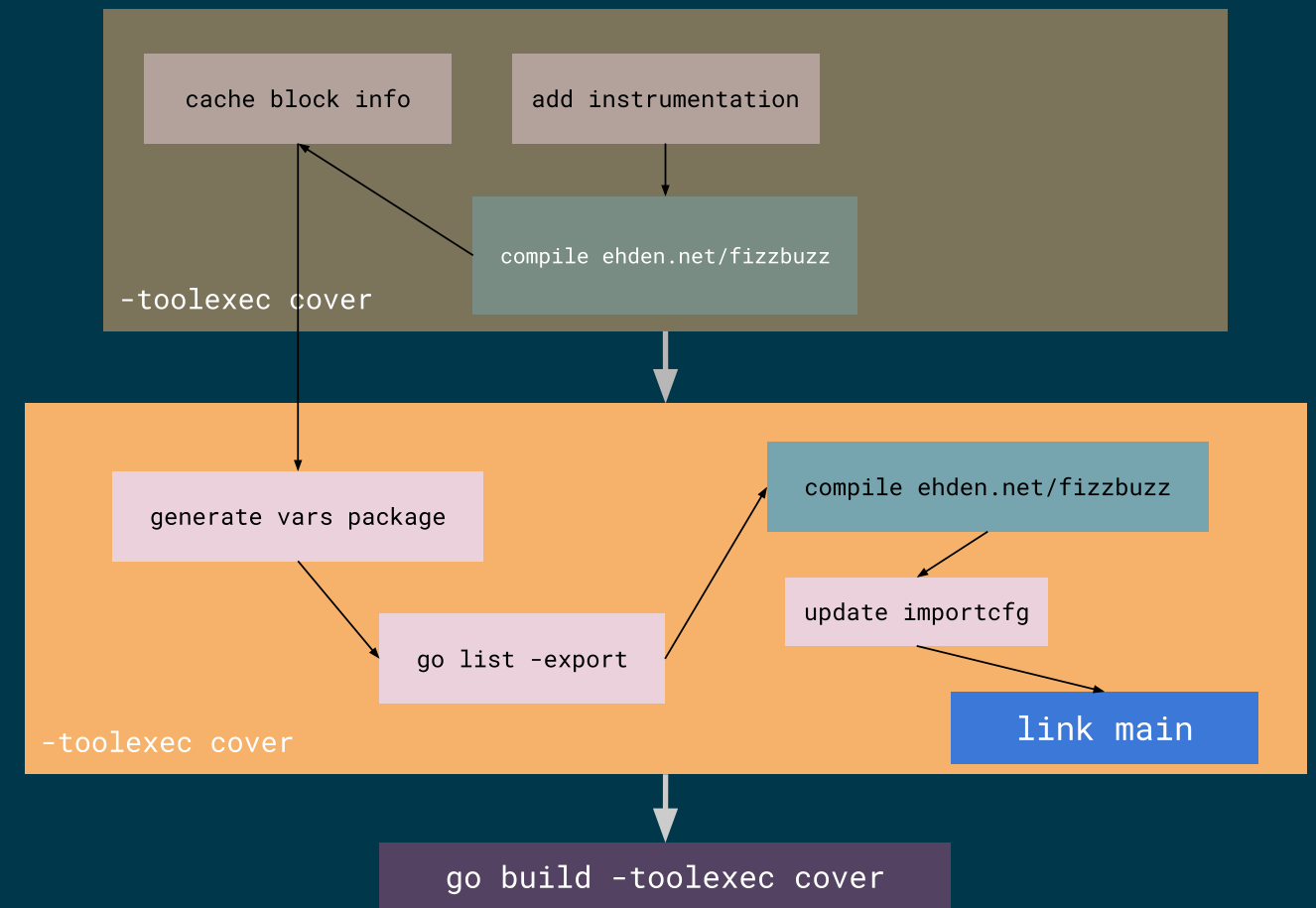
```
go list -toolexec cover -export -f {{ .Export }}
```
4. recompile main so it imports generated package



# ~~the plan~~ hold my beer

1. cache block info while we compile each package
2. generate package code with importcfg.link
3. build our generated package  

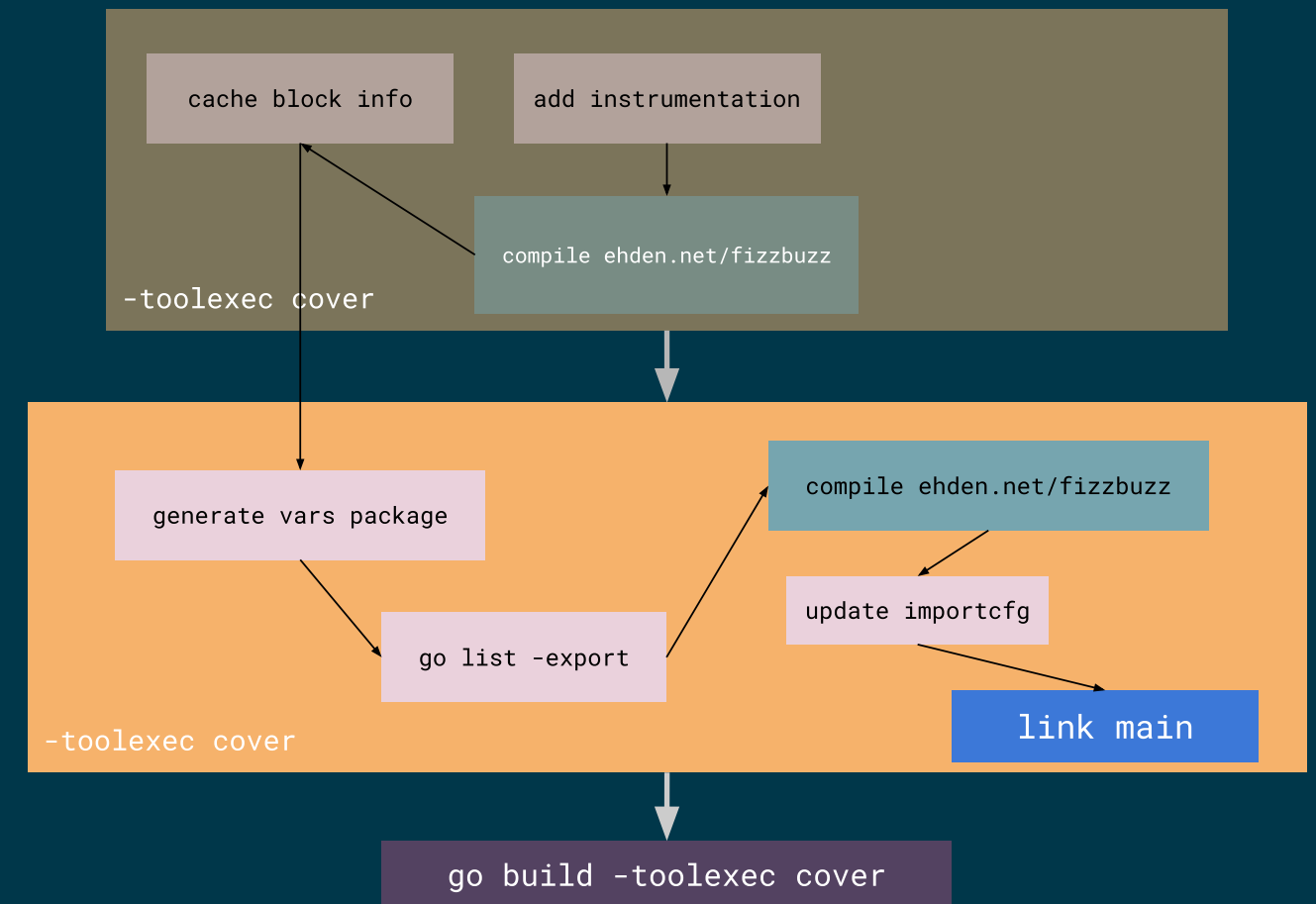
```
go list -toolexec cover -export -f {{ .Export }}
```
4. recompile main so it imports generated package
5. import linker's importcfg to include new packages



# ~~the plan~~ hold my beer

1. cache block info while we compile each package
2. generate package code with importcfg.link
3. build our generated package  

```
go list -toolexec cover -export -f {{ .Export }}
```
4. recompile main so it imports generated package
5. import linker's importcfg to include new packages
6. quiet contemplation



# please talk to me

*gophers slack:* @ehden

*email:* ehdens@gmail.com

*discord:* cixel



talk materials and code available at  
[github.com/cixel/gc2024](https://github.com/cixel/gc2024)