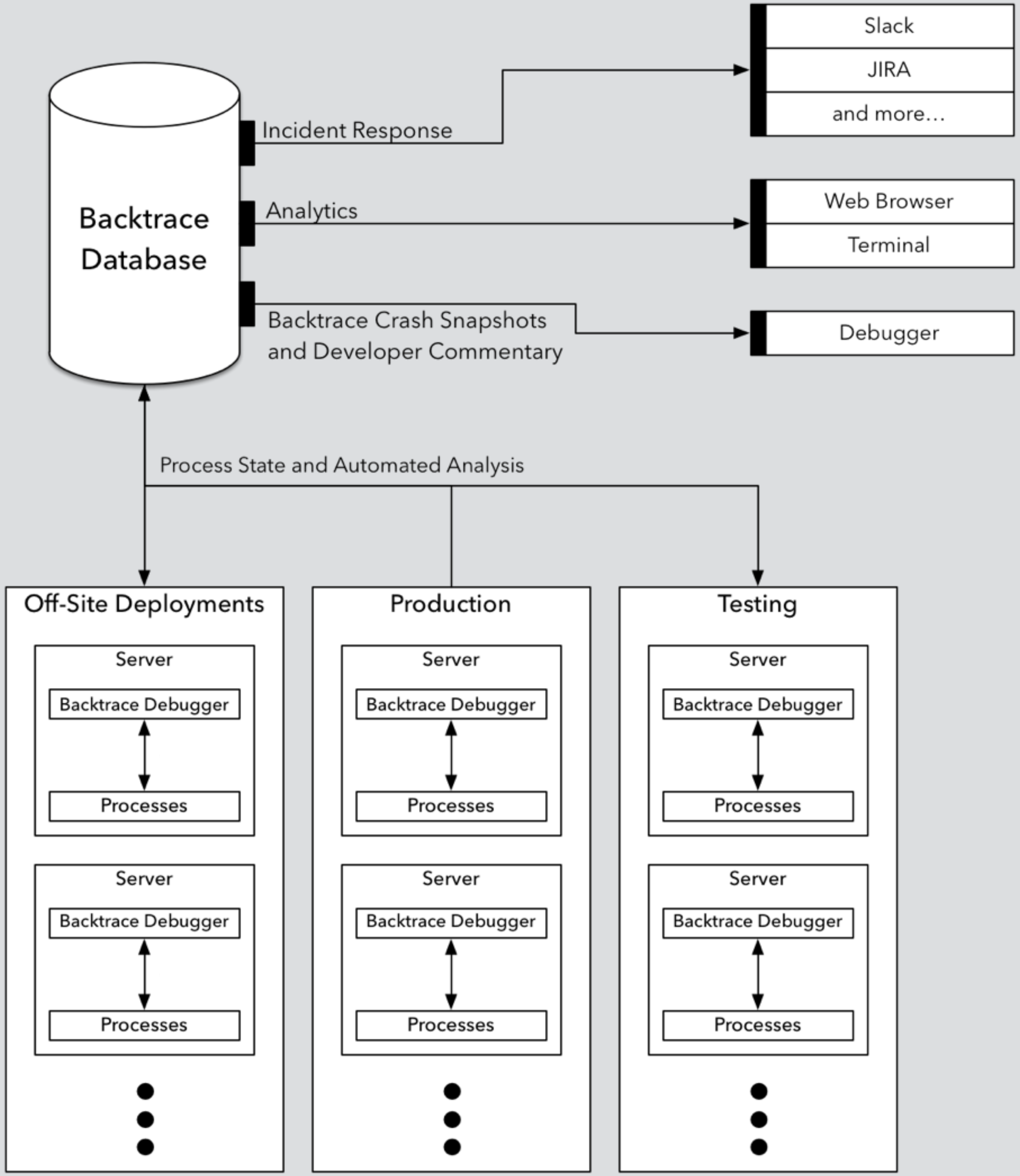
Building a Go Debugger

Abel Mathew
amathew@backtrace.io
@0xcd03

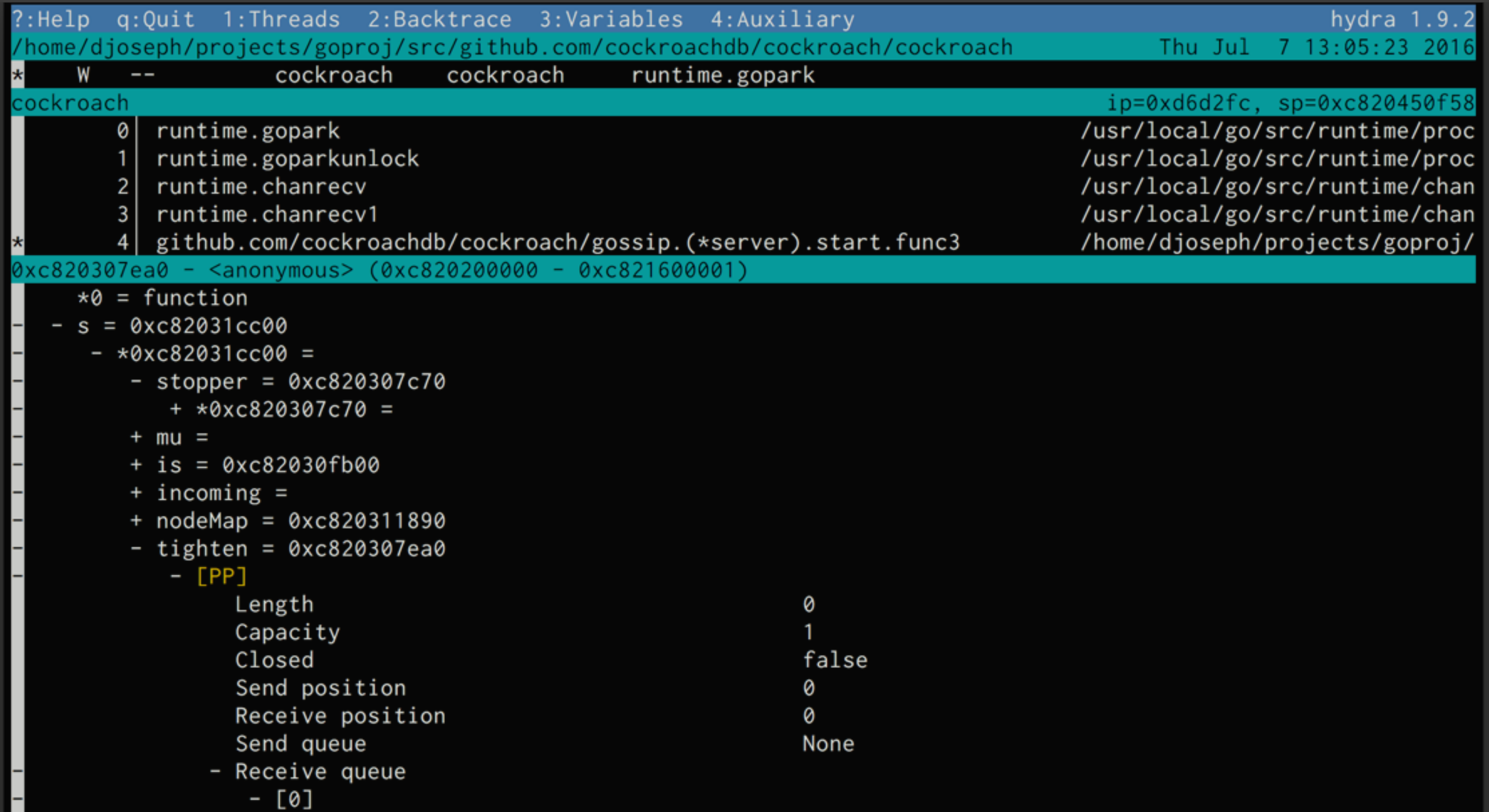
Backtrace


Backtrace Database
Incident Response
Slack
JIRA
and more…
Analytics
Web Browser
Terminal
Backtrace Crash Snapshots and Developer Commentary
Debugger
Process State and Automated Analysis
Off-Site Deployments
Production
Testing
Server
Backtrace Debugger
Processes
Server
Backtrace Debugger
Processes
Server
Backtrace Debugger
Processes
Server
Backtrace Debugger
Processes
Server
Backtrace Debugger
Processes
Server
Backtrace Debugger
Processes

Backtrace

Background

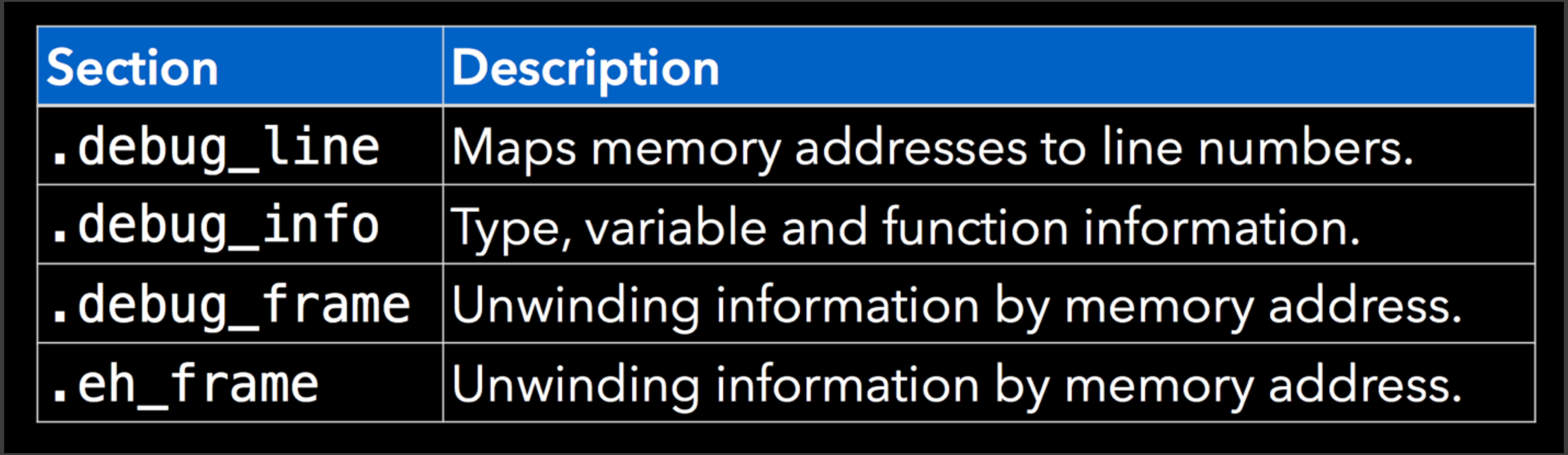
A symbolic debugger maps the state of registers and memory to a backtrace with variables and type information.



```
?:Help  q:Quit  1:Threads  2:Backtrace  3:Variables  4:Auxiliary                                    hydra 1.9.2
/home/djoseph/projects/goproj/src/github.com/cockroachdb/cockroach/cockroach           Thu Jul  7 13:05:23 2016
*    W   --         cockroach    cockroach     runtime.gopark
cockroach                                                                          ip=0xd6d2fc, sp=0xc820450f58
        0| runtime.gopark                                                       /usr/local/go/src/runtime/proc
        1| runtime.goparkunlock                                                 /usr/local/go/src/runtime/proc
        2| runtime.chanrecv                                                     /usr/local/go/src/runtime/chan
        3| runtime.chanrecv1                                                    /usr/local/go/src/runtime/chan
*       4| github.com/cockroachdb/cockroach/gossip.(*server).start.func3       /home/djoseph/projects/goproj/
0xc820307ea0 - <anonymous> (0xc820200000 - 0xc821600001)
     *0 = function
-  - s = 0xc82031cc00
-     - *0xc82031cc00 =
-        - stopper = 0xc820307c70
-           + *0xc820307c70 =
-        + mu =
-        + is = 0xc82030fb00
-        + incoming =
-        + nodeMap = 0xc820311890
-        - tighten = 0xc820307ea0
-           - [PP]
                 Length                                     0
                 Capacity                                   1
                 Closed                                     false
                 Send position                              0
                 Receive position                           0
                 Send queue                                 None
-              - Receive queue
                  - [0]
```

Backtrace

A symbolic debugger uses debug information to do this mapping.

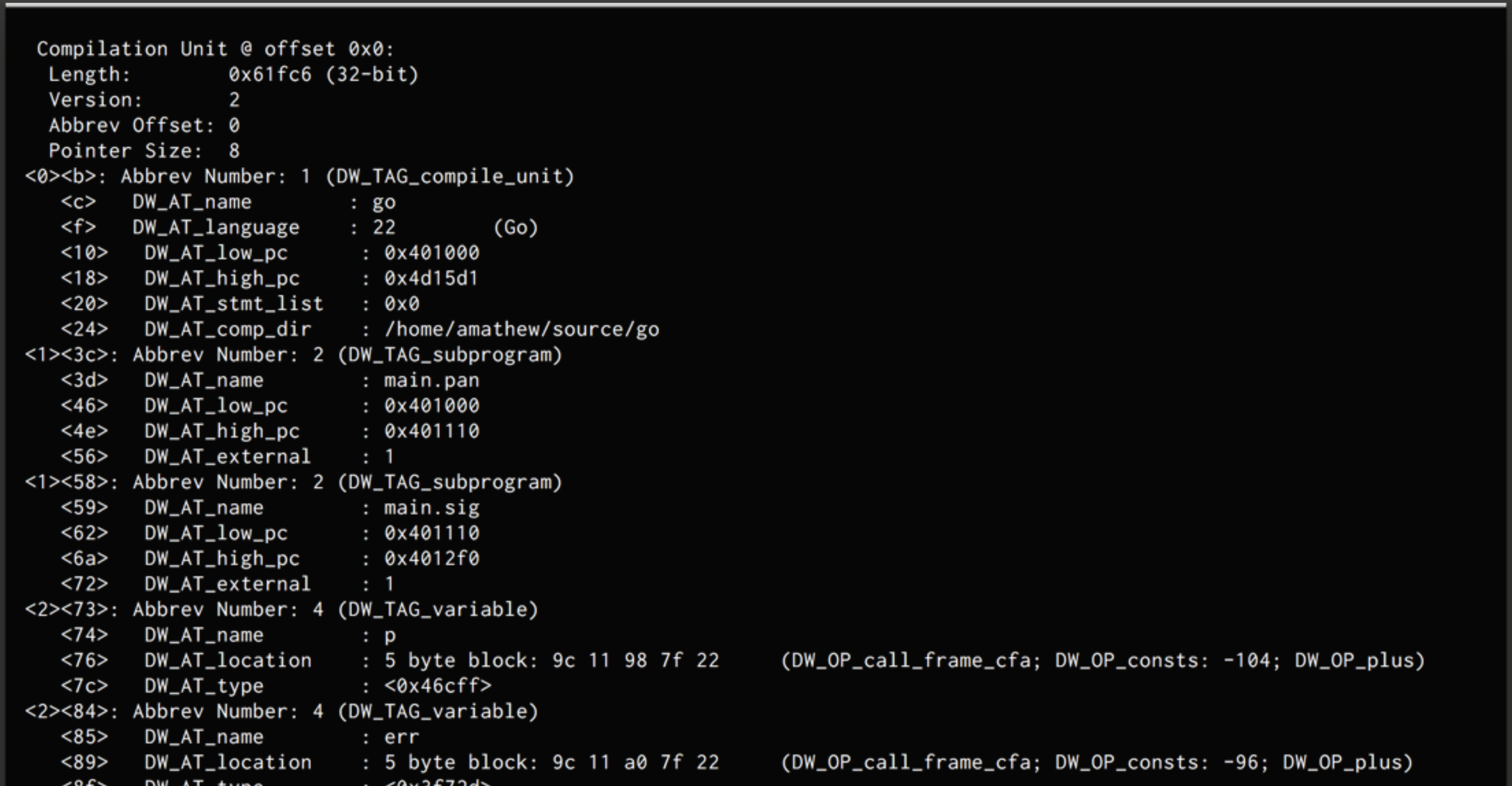


Section	Description
`.debug_line`	Maps memory addresses to line numbers.
`.debug_info`	Type, variable and function information.
`.debug_frame`	Unwinding information by memory address.
`.eh_frame`	Unwinding information by memory address.

DWARF is a popular format for debug information, used by many languages on some UNIX-like systems.

```
$ readelf --debug-dump=info main
Contents of the .debug_info section:

  Compilation Unit @ offset 0x0:
   Length:        0x61fc6 (32-bit)
   Version:       2
   Abbrev Offset: 0
   Pointer Size:  8

$ dwarfdump -i main
.debug_info
COMPILE_UNIT<header overall offset = 0x00000000>:
< 0><0x0000000b>  DW_TAG_compile_unit
                    DW_AT_name                  "go"
                    DW_AT_language              DW_LANG_Go
                    DW_AT_low_pc                0x00401000
                    DW_AT_high_pc               0x004d15d1
                    DW_AT_stmt_list             0x00000000
                    DW_AT_comp_dir              "/home/amathew/source/go"
```

There are variations in the DWARF generated by various compilers, including the Go compiler



```
 Compilation Unit @ offset 0x0:
  Length:        0x61fc6 (32-bit)
  Version:       2
  Abbrev Offset: 0
  Pointer Size:  8
<0><b>: Abbrev Number: 1 (DW_TAG_compile_unit)
   <c>   DW_AT_name        : go
   <f>   DW_AT_language    : 22        (Go)
   <10>   DW_AT_low_pc      : 0x401000
   <18>   DW_AT_high_pc     : 0x4d15d1
   <20>   DW_AT_stmt_list   : 0x0
   <24>   DW_AT_comp_dir    : /home/amathew/source/go
<1><3c>: Abbrev Number: 2 (DW_TAG_subprogram)
   <3d>   DW_AT_name        : main.pan
   <46>   DW_AT_low_pc      : 0x401000
   <4e>   DW_AT_high_pc     : 0x401110
   <56>   DW_AT_external    : 1
<1><58>: Abbrev Number: 2 (DW_TAG_subprogram)
   <59>   DW_AT_name        : main.sig
   <62>   DW_AT_low_pc      : 0x401110
   <6a>   DW_AT_high_pc     : 0x4012f0
   <72>   DW_AT_external    : 1
<2><73>: Abbrev Number: 4 (DW_TAG_variable)
   <74>   DW_AT_name        : p
   <76>   DW_AT_location    : 5 byte block: 9c 11 98 7f 22     (DW_OP_call_frame_cfa; DW_OP_consts: -104; DW_OP_plus)
   <7c>   DW_AT_type        : <0x46cff>
<2><84>: Abbrev Number: 4 (DW_TAG_variable)
   <85>   DW_AT_name        : err
   <89>   DW_AT_location    : 5 byte block: 9c 11 a0 7f 22     (DW_OP_call_frame_cfa; DW_OP_consts: -96; DW_OP_plus)
   <8f>   DW_AT_type        : <0x3f72d>
```

Backtrace

- DW_FORM_ref_addr value width mismatch
- Discovering (kernel/OS) threads — libthread_db and clone()
- Special unwinding for goroutines under certain conditions (e.g. GC)
- Complex Go types and type dereferencing
- Multidimensional array representation
- Extracting goroutines
- and more…

- DW_FORM_ref_addr value width mismatch
- Discovering (kernel/OS) threads — libthread_db and clone()
- Special unwinding for goroutines under various conditions (e.g. GC)
- Complex Go types and type dereferencing
- **Multidimensional array representation**
- Extracting goroutines
- and more…

Multidimensional Arrays

Each array dimension is described by a debugging information entry with either the tag DW_TAG_subrange_type or the tag DW_TAG_enumeration_type. These entries are children of the array type entry and are ordered to reflect the appearance of the dimensions in the source program (i.e. leftmost dimension first, next to leftmost second, and so on).

DWARF2 Standard, Page 39 — http://www.dwarfstd.org/doc/dwarf-2.0.0.pdf

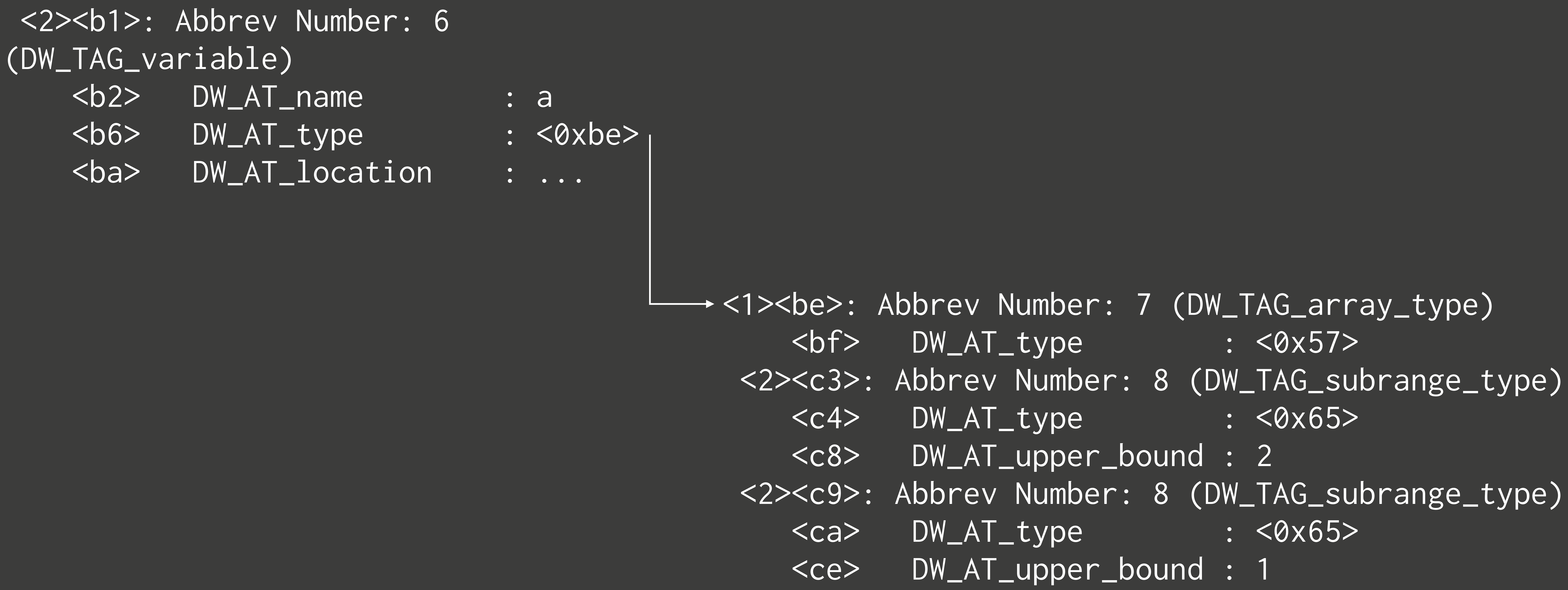
```
int
main()
{
        int a[3][2] = {
            [0] = {0, 1},
            [1] = {1, 2},
            [2] = {2, 3}
        };
       …

}
```

Variable information:

```
 <2><b1>: Abbrev Number: 6 (DW_TAG_variable)
    <b2>   DW_AT_name        : a
    <b6>   DW_AT_type        : <0xbe>
    <ba>   DW_AT_location    : ...
```

Type information:

```
 <1><be>: Abbrev Number: 7 (DW_TAG_array_type)
    <bf>   DW_AT_type        : <0x57>
 <2><c3>: Abbrev Number: 8 (DW_TAG_subrange_type)
    <c4>   DW_AT_type        : <0x65>
    <c8>   DW_AT_upper_bound : 2
 <2><c9>: Abbrev Number: 8 (DW_TAG_subrange_type)
    <ca>   DW_AT_type        : <0x65>
    <ce>   DW_AT_upper_bound : 1
```



```
 <2><b1>: Abbrev Number: 6
(DW_TAG_variable)
    <b2>   DW_AT_name        : a
    <b6>   DW_AT_type        : <0xbe>
    <ba>   DW_AT_location    : ...
```

```
<1><be>: Abbrev Number: 7 (DW_TAG_array_type)
    <bf>   DW_AT_type        : <0x57>
 <2><c3>: Abbrev Number: 8 (DW_TAG_subrange_type)
    <c4>   DW_AT_type        : <0x65>
    <c8>   DW_AT_upper_bound : 2
 <2><c9>: Abbrev Number: 8 (DW_TAG_subrange_type)
    <ca>   DW_AT_type        : <0x65>
    <ce>   DW_AT_upper_bound : 1
```

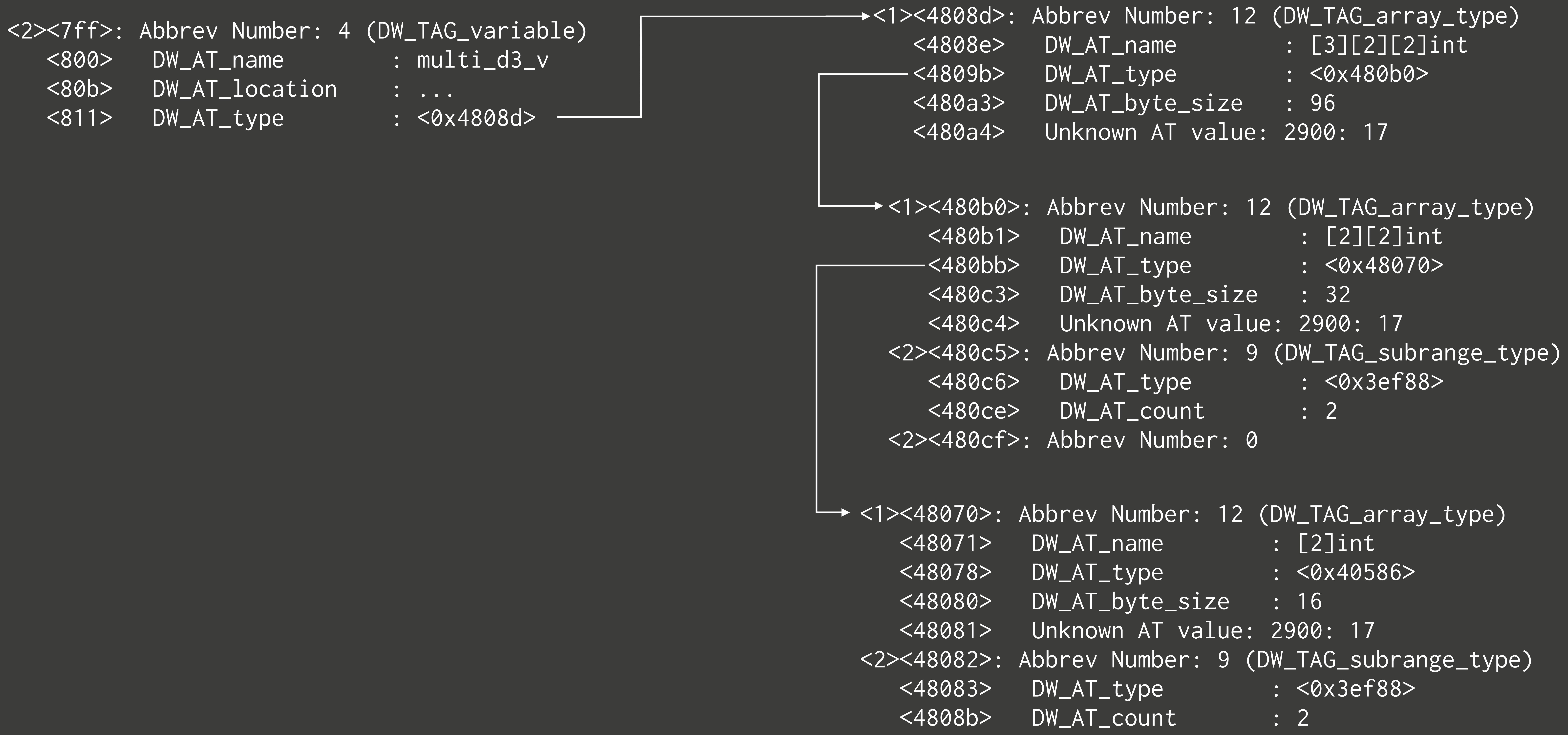
```
func main() {
        multi_d3_v := [3][2][2]int{
                {{3, 4}, {1, 2}},
                {{3, 4}, {3, 4}},
                {{3, 4}, {5, 6}},
        }
        …
}
```

Variable information:

```
 <2><7ff>: Abbrev Number: 4 (DW_TAG_variable)
    <800>   DW_AT_name        : multi_d3_v
    <80b>   DW_AT_location    : ...
    <811>   DW_AT_type        : <0x4808d>
```

Type information:

```
 <1><48070>: Abbrev Number: 12 (DW_TAG_array_type)
    <48071>   DW_AT_name        : [2]int
    <48078>   DW_AT_type        : <0x40586>
    <48080>   DW_AT_byte_size   : 16
    <48081>   Unknown AT value: 2900: 17
 <2><48082>: Abbrev Number: 9 (DW_TAG_subrange_type)
    <48083>   DW_AT_type        : <0x3ef88>
    <4808b>   DW_AT_count       : 2
 <2><4808c>: Abbrev Number: 0
 <1><4808d>: Abbrev Number: 12 (DW_TAG_array_type)
    <4808e>   DW_AT_name        : [3][2][2]int
    <4809b>   DW_AT_type        : <0x480b0>
    <480a3>   DW_AT_byte_size   : 96
    <480a4>   Unknown AT value: 2900: 17
 <2><480a5>: Abbrev Number: 9 (DW_TAG_subrange_type)
    <480a6>   DW_AT_type        : <0x3ef88>
    <480ae>   DW_AT_count       : 3
 <2><480af>: Abbrev Number: 0
 <1><480b0>: Abbrev Number: 12 (DW_TAG_array_type)
    <480b1>   DW_AT_name        : [2][2]int
    <480bb>   DW_AT_type        : <0x48070>
    <480c3>   DW_AT_byte_size   : 32
    <480c4>   Unknown AT value: 2900: 17
 <2><480c5>: Abbrev Number: 9 (DW_TAG_subrange_type)
    <480c6>   DW_AT_type        : <0x3ef88>
    <480ce>   DW_AT_count       : 2
 <2><480cf>: Abbrev Number: 0
```

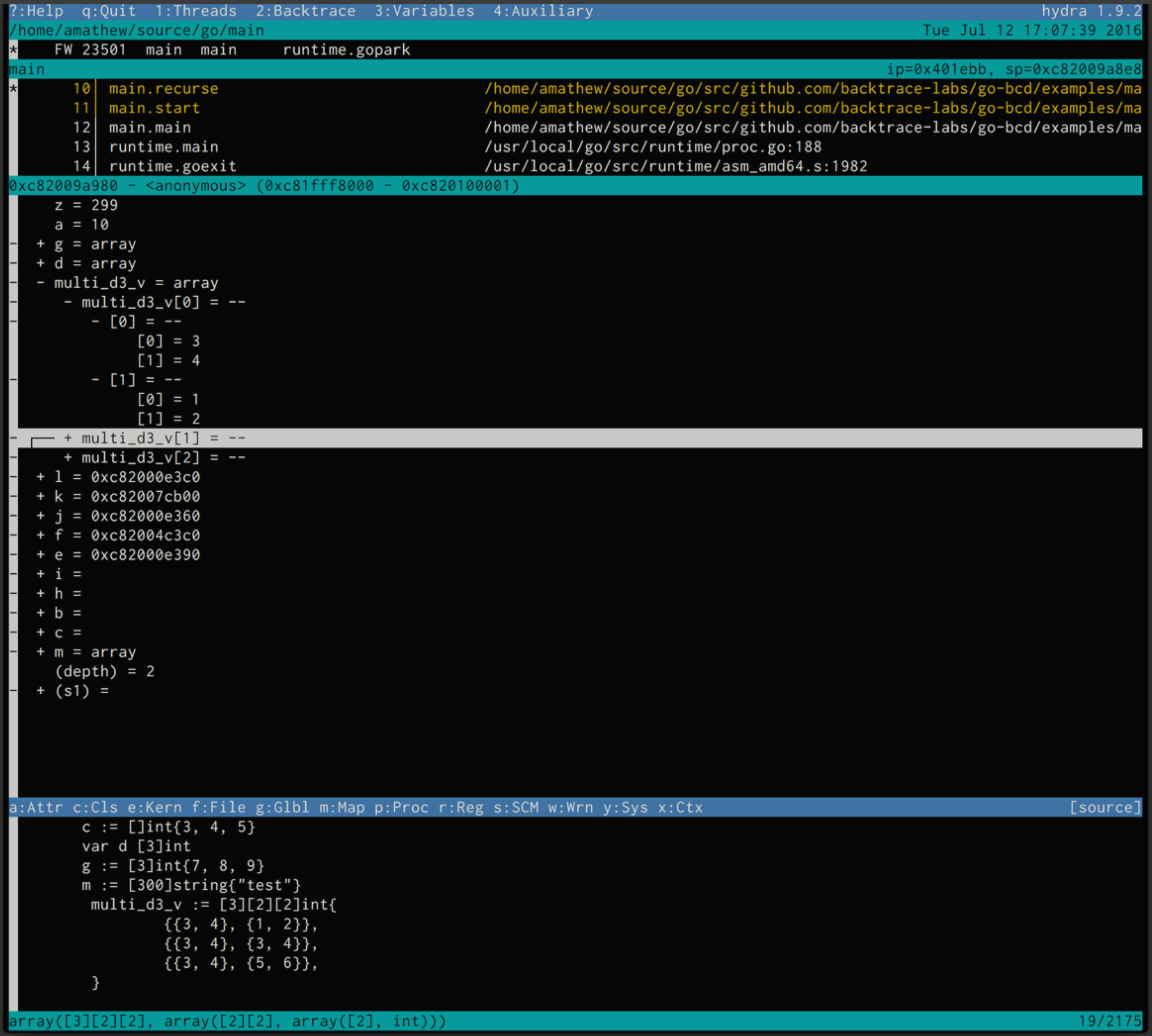


```
<2><7ff>: Abbrev Number: 4 (DW_TAG_variable)
   <800>   DW_AT_name          : multi_d3_v
   <80b>   DW_AT_location      : ...
   <811>   DW_AT_type          : <0x4808d>
```

```
<1><4808d>: Abbrev Number: 12 (DW_TAG_array_type)
   <4808e>   DW_AT_name         : [3][2][2]int
   <4809b>   DW_AT_type         : <0x480b0>
   <480a3>   DW_AT_byte_size    : 96
   <480a4>   Unknown AT value: 2900: 17
```

```
<1><480b0>: Abbrev Number: 12 (DW_TAG_array_type)
   <480b1>   DW_AT_name         : [2][2]int
   <480bb>   DW_AT_type         : <0x48070>
   <480c3>   DW_AT_byte_size    : 32
   <480c4>   Unknown AT value: 2900: 17
<2><480c5>: Abbrev Number: 9 (DW_TAG_subrange_type)
   <480c6>   DW_AT_type         : <0x3ef88>
   <480ce>   DW_AT_count        : 2
<2><480cf>: Abbrev Number: 0
```

```
<1><48070>: Abbrev Number: 12 (DW_TAG_array_type)
   <48071>   DW_AT_name         : [2]int
   <48078>   DW_AT_type         : <0x40586>
   <48080>   DW_AT_byte_size    : 16
   <48081>   Unknown AT value: 2900: 17
<2><48082>: Abbrev Number: 9 (DW_TAG_subrange_type)
   <48083>   DW_AT_type         : <0x3ef88>
   <4808b>   DW_AT_count        : 2
```



```
?:Help  q:Quit  1:Threads  2:Backtrace  3:Variables  4:Auxiliary                          hydra 1.9.2
/home/amathew/source/go/main                                                  Tue Jul 12 17:07:39 2016
*    FW 23501  main  main     runtime.gopark
main                                                                 ip=0x401ebb, sp=0xc82009a8e8
*       10| main.recurse                              /home/amathew/source/go/src/github.com/backtrace-labs/go-bcd/examples/ma
        11| main.start                                /home/amathew/source/go/src/github.com/backtrace-labs/go-bcd/examples/ma
        12| main.main                                 /home/amathew/source/go/src/github.com/backtrace-labs/go-bcd/examples/ma
        13| runtime.main                              /usr/local/go/src/runtime/proc.go:188
        14| runtime.goexit                            /usr/local/go/src/runtime/asm_amd64.s:1982
0xc82009a980 - <anonymous> (0xc81fff8000 - 0xc820100001)
     z = 299
     a = 10
-  + g = array
-  + d = array
-  - multi_d3_v = array
-     - multi_d3_v[0] = --
-        - [0] = --
              [0] = 3
              [1] = 4
-        - [1] = --
              [0] = 1
              [1] = 2
-  ┌── + multi_d3_v[1] = --
-       + multi_d3_v[2] = --
-  + l = 0xc82000e3c0
-  + k = 0xc82007cb00
-  + j = 0xc82000e360
-  + f = 0xc82004c3c0
-  + e = 0xc82000e390
-  + i =
-  + h =
-  + b =
-  + c =
-  + m = array
     (depth) = 2
-  + (s1) =

a:Attr c:Cls e:Kern f:File g:Glbl m:Map p:Proc r:Reg s:SCM w:Wrn y:Sys x:Ctx                   [source]
        c := []int{3, 4, 5}
        var d [3]int
        g := [3]int{7, 8, 9}
        m := [300]string{"test"}
         multi_d3_v := [3][2][2]int{
                 {{3, 4}, {1, 2}},
                 {{3, 4}, {3, 4}},
                 {{3, 4}, {5, 6}},
         }
array([3][2][2], array([2][2], array([2], int)))                                              19/2175
```

Follow-up post coming soon:
http://backtrace.io/blog

Backtrace

Building a Go Debugger

Abel Mathew
amathew@backtrace.io
@0xcd03

Backtrace