one system
all devices

pres 0#Intro

Geo Carncross <g@telemetry.com>

one system
all devices

pres 1#Intro

Geo Carncross <g@telemetry.com>

* Big: 1.6TB lzma'd logfiles per day
  from all over the world!

* Fast: Deadline is 50msec

* 100% Uptime is normal for us.

Our logfiles are money!

(we do other things, like fighting bad guys committing ad fraud,
and everyone in my office is enjoying a beer right now...

let me know if this is something you want to know more about)

* I'm part of the reason ads play fine, but your Youtube stutters

one system
all devices

pres 2#Intro

How to program computers.

one system
all devices

pres 3#Intro

How to program computers.
(no really)

* Yes, I will be talking about kOS

```
one system
all devices
```

```
pres 4#Intro - code bloat
```

How to program computers.

one system
all devices

pres 5#Intro - code bloat

How to program computers.

* Give it a meaningful definition

one system
all devices

pres 6#Intro - code bloat

How to program computers.

* Give it a meaningful definition

* Don't be distracted by other,
  informal uses of "code bloat"

one system
all devices

pres 7#Intro - code bloat

How to program computers.

* Give it a meaningful definition

* Don't be distracted by other,
  informal uses of "code bloat"

* Work the problem as it appears

one system
all devices

pres 8#Intro - kos

How to program computers.

kos is a small team (4 people)

pres 9#Intro - kos

one system
all devices

How to program computers.

kos is tiny

* k is arthur's language; something like scheme+apl
* kdb is a database faster than popular rdbms by a factor of 1000x
* z is a gui and window manager, icons, mouse and font
* kos kernel is modesetting, memory, vmm, isr, and filesystem

```
one system
all devices
```

```
pres 10#Intro - kos

How to program computers.

kos is tiny

* k is 300 lines of C across 5 files
* kdb adds only about 100 lines of C
* z adds another 60 lines of C
* kos kernel is 100 lines of C/assembly
```

one system
all devices

pres 11#Intro - kos

kos is how to program computers.

```
  for(i=0;i<1000000;++i);       i:1000000(1+)/1
```

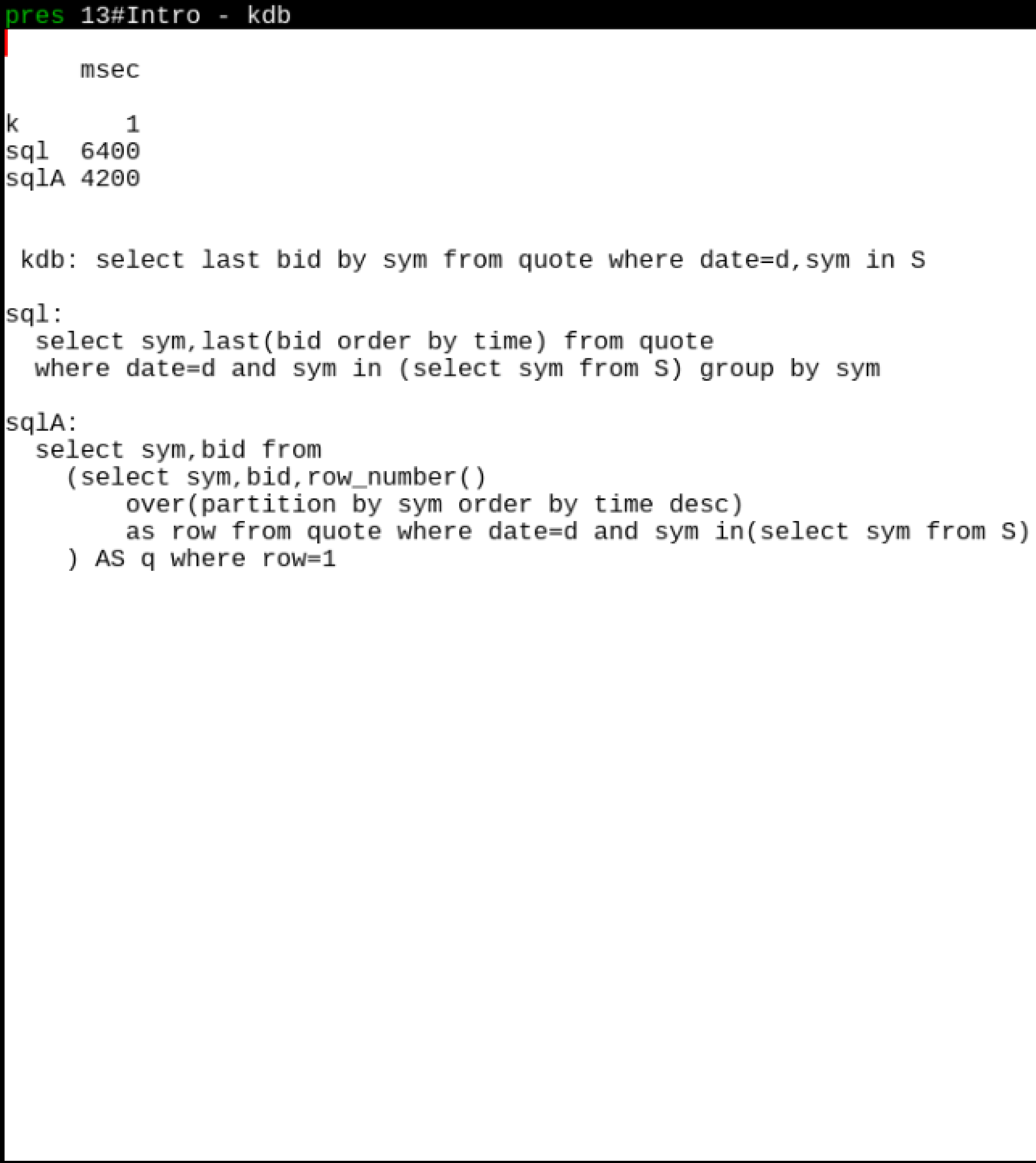
one system
all devices

pres 12#Intro - kos

kos is how to program computers.

```
  for(i=0;i<1000000;++i);       i:1000000(1+)/1
  real  0m0.008s                real    0m0.191s
  user  0m0.003s                user    0m0.185s
  sys   0m0.002s                sys     0m0.003s
```

one system
all devices



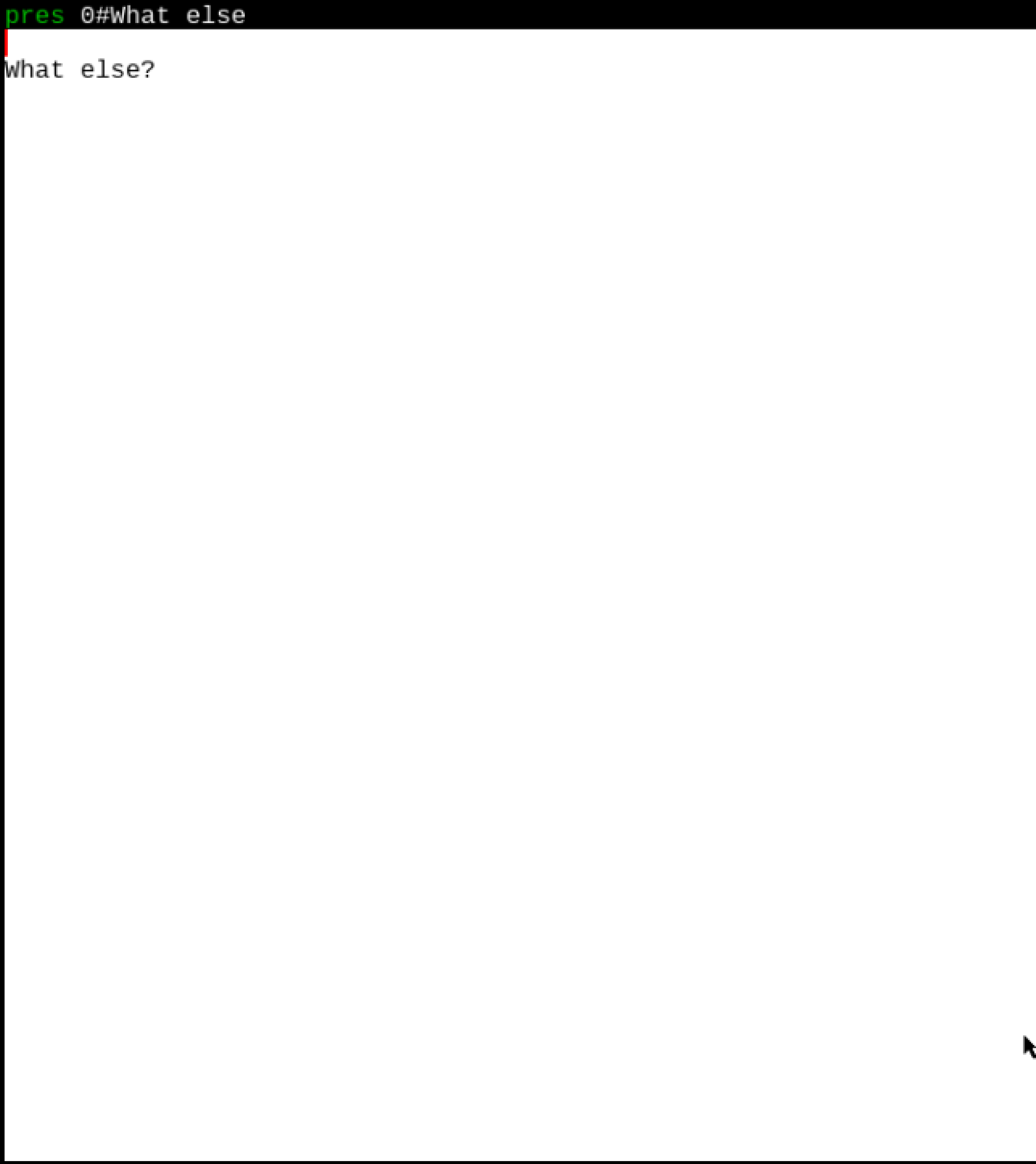
pres 13#Intro - kdb

```
     msec

k       1
sql  6400
sqlA 4200


 kdb: select last bid by sym from quote where date=d,sym in S

sql:
  select sym,last(bid order by time) from quote
  where date=d and sym in (select sym from S) group by sym

sqlA:
  select sym,bid from
    (select sym,bid,row_number()
        over(partition by sym order by time desc)
        as row from quote where date=d and sym in(select sym from S)
    ) AS q where row=1
```

```
one system
all devices
```

```
pres 14#Intro - how


* Unnecessary code is a bug and we try to remove it.


* Slow programs are the programmers responsibility.


* Bugs are mistakes we programmers make;
  they are not inevitable or ok.


* The programmer is responsible that the program is used
  correctly.


* "O" complexity ignores that computers are made of hardware

  * Arrays might be O(1) when small but O(log n) when bigger
  * B-trees are a terrible data structure for storing data
  * Hashtables aren't remotely close to O(1)


* "Best practices" are a straw man that make having
  real discussions about making systems difficult.

  * That includes syntax hilighting
  * ... and object-oriented programming
  * ... and what is "clean code"?
  * ... and unit-tests
  * ... and debian packages
  * ... and puppet/chef/etc


* Programming correctly is hard!


* Can you write an algorithm shorter or faster than me?

  If so I'd like to talk to you: PUTITYYEUOP
```

one system
all devices



```
pres 0#What else

What else?
```

```
pres 1#Hello World?
```

one system
all devices

```
main(){puts("Hello World");}  k:"Hello World"
real  0m0.006s                real    0m0.016s
user  0m0.001s                user    0m0.012s
sys   0m0.002s                sys     0m0.006s
```

one system
all devices

pres 2#What else

```
* Unnecessary code is a bug and we try to remove it.


* Slow programs are the programmers responsibility.


* Bugs are mistakes we programmers make;
  they are not inevitable or ok.


* The programmer is responsible that the program is used
  correctly.


* "O" complexity ignores that computers are made of hardware

  * Arrays might be O(1) when small but O(log n) when bigger
  * B-trees are a terrible data structure for storing data
  * Hashtables aren't remotely close to O(1)


* "Best practices" are a straw man that make having
  real discussions about making systems difficult.

  * That includes syntax hilighting
  * ... and object-oriented programming
  * ... and type systems
  * ... and unit-tests
  * ... and debian packages
  * ... and puppet/chef/etc


* Programming correctly is hard!
```

one system
all devices

pres 3#What else

What else?

one system
all devices

```
pres 4#Goodbye
```

Geo Carncross <g@telemetry.com>

```
n:0;p:0:"p.md";co:{p::0:0'""}
c::(&"#"=*:'p),#p;a::*(1 0+c[n,n+1])_p;x::($n),p c n
z::{1'((x**F),0;a[x])}'!#a;
lx:{2'(0;W;7);n::|/0,&/(-2+#c),n+x}
```