KubeCon CloudNativeCon
North America 2026

Not All Memory Is Equal: How To Properly Observe Memory Usage

Braydon Kains and Roger Coll







(there's no Collector logo but I actually pulled this svg from the Collector documentation page on opentelemetry.io)

```
$ ps -p $(pidof otelcontribcol) -o pid,args,rss
    PID COMMAND                         RSS
 260938 /tmp/otelcontribcol             217252
```

```
$ ps -p $(pidof otelcorecol) -o pid,args,rss
    PID COMMAND                         RSS
 262781 /tmp/otelcorecol                61808
```

```
$ cat /proc/$(pidof otelcontribcol)/status
Name:   otelcontribcol
 ...
VmPeak:  1742352 kB
VmSize:  1742352 kB
VmLck:         0 kB
VmPin:         0 kB
VmHWM:    221816 kB
VmRSS:    221816 kB
RssAnon:           61032 kB
RssFile:          160784 kB
RssShmem:              0 kB
```







Braydon Kains

Google Cloud, Observability

@braydonk

Roger Coll

Grafana, Pyroscope

@rogercoll

Both working on System Semantic Conventions and the `host_metrics` receiver

Is Random Access Memory infinite?



Twitter Testimonies



Latest Review



Contact Info



Download More RAM Co.
Wind up, NV 12345
Email:
info@downloadmoreram.com

Welcome

What is this site you ask? Download more RAM does exactly what it says on the tin. Our website gives you the ability to 'download more RAM' at no cost! Just select the amount you need and download! Your speed will be improved instantly!

What is RAM?

Random-access memory (usually known by its acronym, RAM) is a form of computer data storage. Today, it takes the form of integrated circuits that allow stored data to be accessed in any order (i.e., at random). The word random thus refers to the fact that any piece of data can be returned in a constant time, regardless of its physical location and whether or not it is related to the previous piece of data.

Please select the amount of RAM to download:

1GB



Overview

* 1GB CT12864AA800 Memory
* 240-pin DIMM
* DDR2 PC2-6400, CL=6

Was: ~~$99.99~~ Now: FREE

Download Now

2GB



Overview

* 2 GB (2 x 1 GB)
* 240-pin DIMM
* DDR2 800 MHz (PC2-6400)

Was: ~~$149.99~~ Now: FREE

Download Now

4GB



Overview

* 4 GB (2 x 2 GB)
* 240-pin DIMM
* DDR2 800 MHz (PC2-6400)

Was: ~~$199.99~~ Now: FREE

Download Now

KubeCon CloudNativeCon North America 2026

RAM Memory Corsair Vengaence 32GB 2x16GB DDR5 6000 MHz CL38 Intel XMP AMD Expo Black

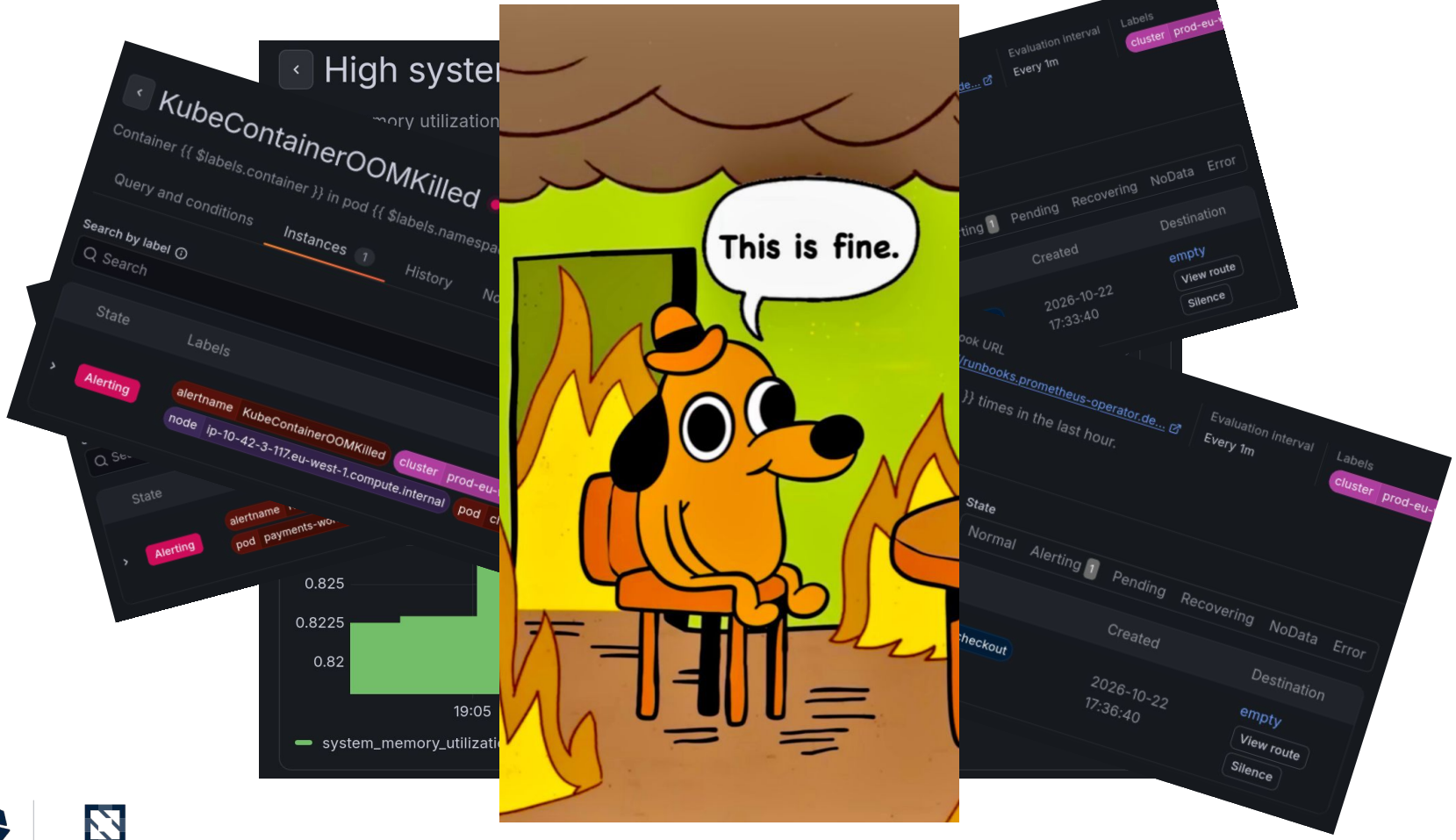
4.6/5 ★ 66 opinions (5) Questions (1)

Color: **Black**

728.21€

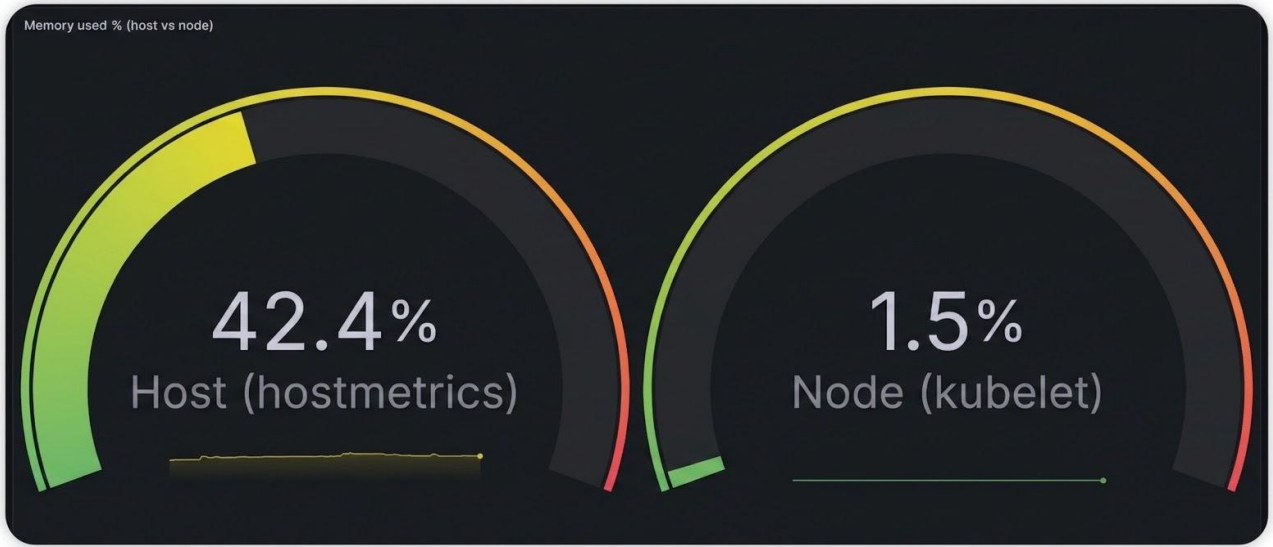
542.90€

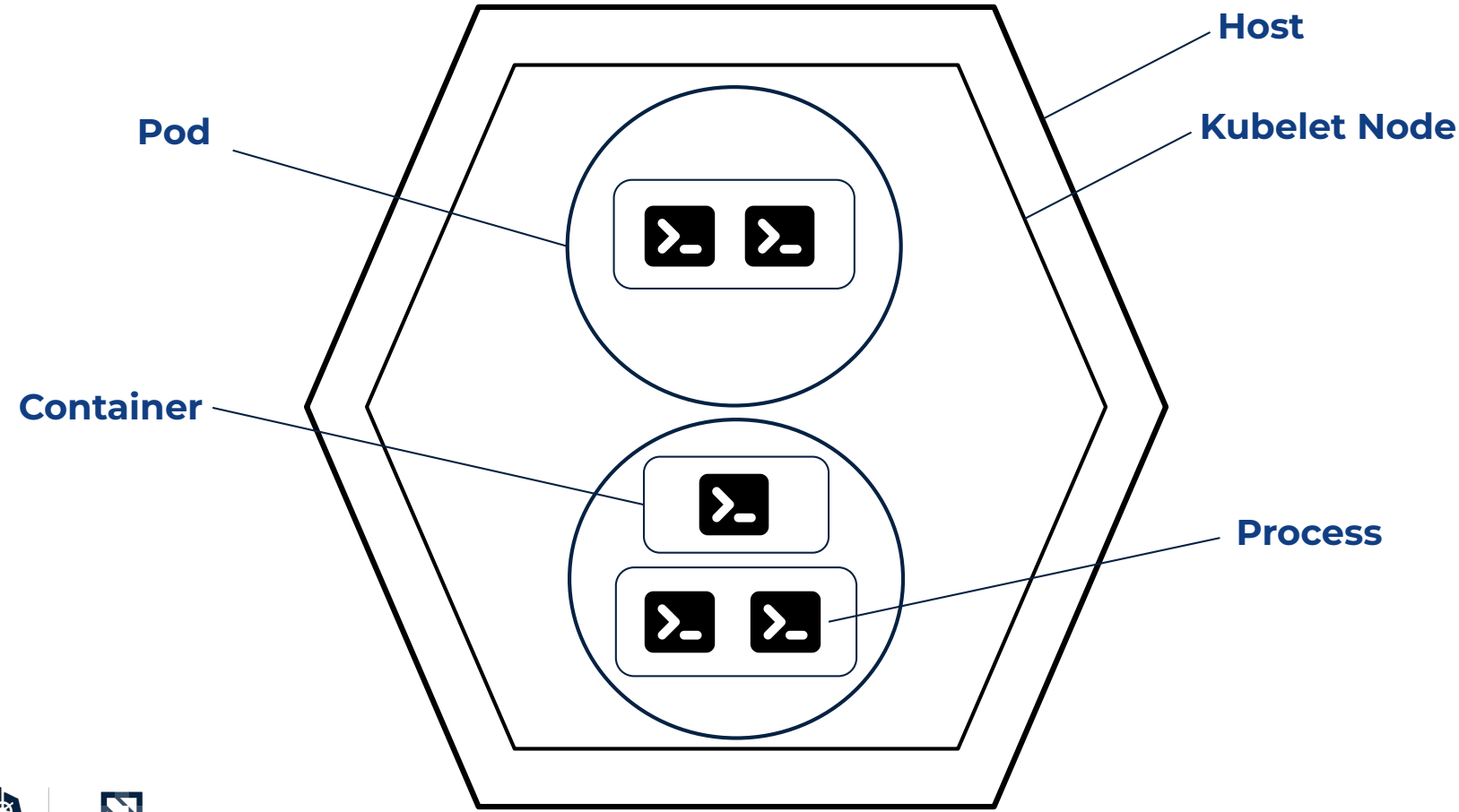
$0.0052	2	0.5 GiB
$0.0104	2	1 GiB
$0.0208	2	2 GiB
$0.0416	2	4 GiB
$0.0832	2	8 GiB



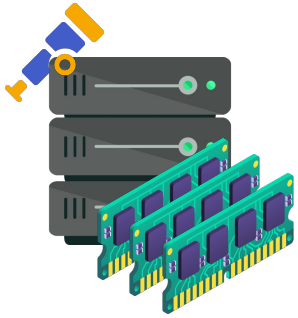
KubeCon CloudNativeCon North America 2026

How do we measure memory usage?


Memory used % (host vs node)
42.4%
Host (hostmetrics)
1.5%
Node (kubelet)


Host
Kubelet Node
Pod
Container
Process

Host OS



system.memory.usage

```
zsh — zsh
$ cat /proc/meminfo
MemTotal:       65106908 kB
MemFree:        36258316 kB
MemAvailable:   49890044 kB
Buffers:         1206856 kB
Cached:          9599892 kB
SwapCached:            0 kB
Active:         14871952 kB
Inactive:        8494672 kB
...
```

Kubelet Node



k8s.node.memory.usage

```
$ cat /sys/fs/cgroup/memory.stat
anon 18409885696
file 24835018752
kernel 1739911168
kernel_stack 36601856
pagetables 164429824
sec_pagetables 0
percpu 7989384
sock 40960
...
```

Kubelet Node



k8s.node.memory.rss

```
$ cat /sys/fs/cgroup/memory.stat
anon 18409885696
file 24835018752
kernel 1739911168
kernel_stack 36601856
pagetables 164429824
sec_pagetables 0
percpu 7989384
sock 40960
...
```

Kubelet Node



k8s.node.memory.working_set

```
$ cat /sys/fs/cgroup/memory.stat
anon 18409885696
file 24835018752
kernel 1739911168
kernel_stack 36601856
pagetables 164429824
sec_pagetables 0
percpu 7989384
sock 40960
...
inactive_file 20568109056
```

k8s.pod.memory.usage
k8s.pod.memory.working_set
k8s.pod.memory.rss
k8s.pod.memory.available

k8s.container.memory.usage
k8s.container.memory.working_set
k8s.container.memory.rss
k8s.container.memory.available
k8s.container.memory.oom_kills

process.memory.usage
process.memory.virtual
process.memory.utilization
process.memory.linux.rss
process.memory.linux.dirty
process.memory.linux.pss



[Braydon] #3 Cover the kernel concepts we'll need

How does memory management work?

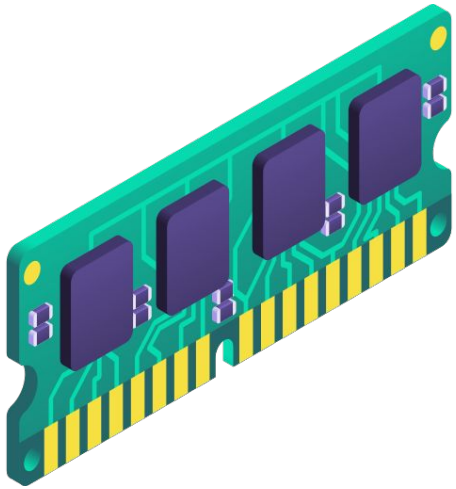
THE LINUX MEMORY MANAGER

Lorenzo Stoakes

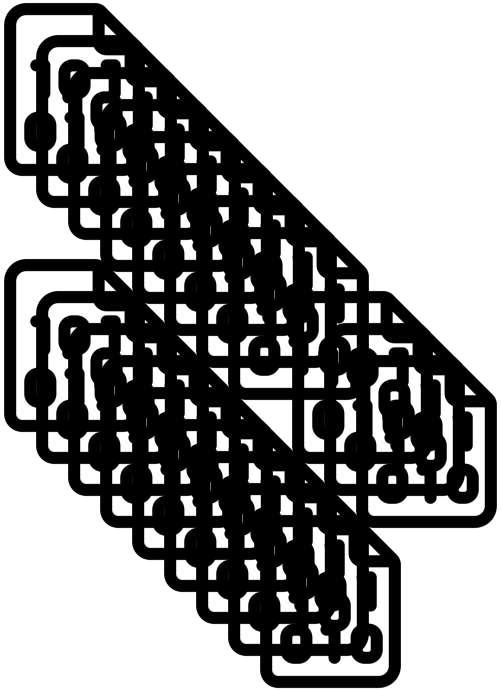


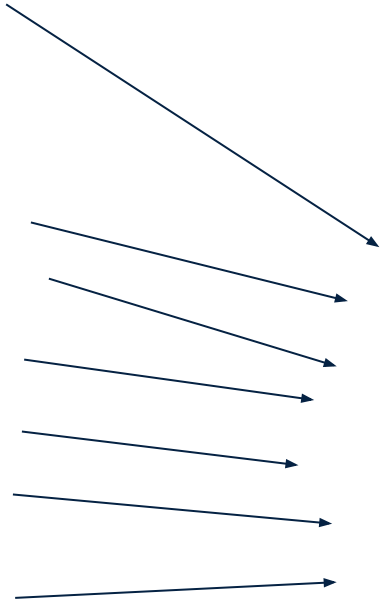
no starch press

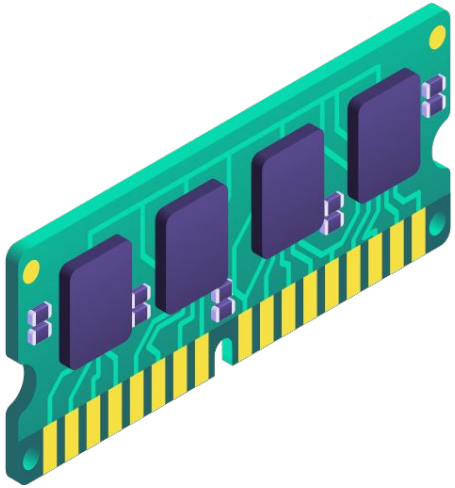

101
010



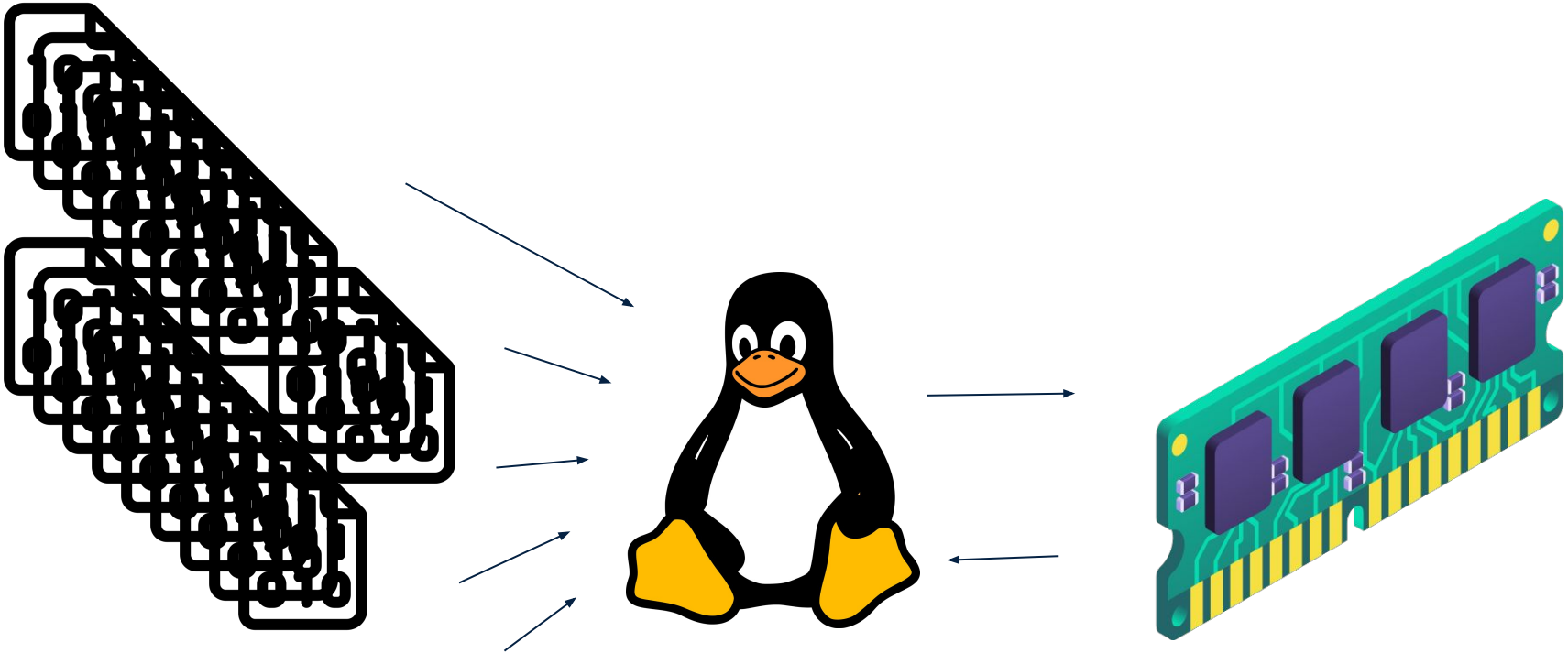


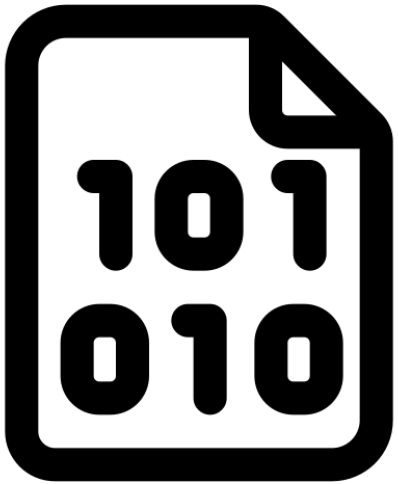












00000000

....

....

....

111111111

0000000

....

....

....

....

....

....

....

....

....

....

....

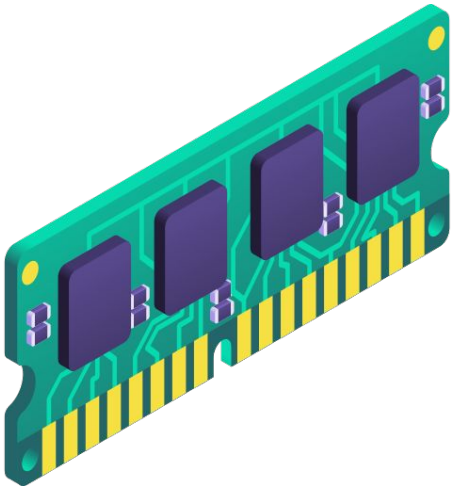
....

....

....

....

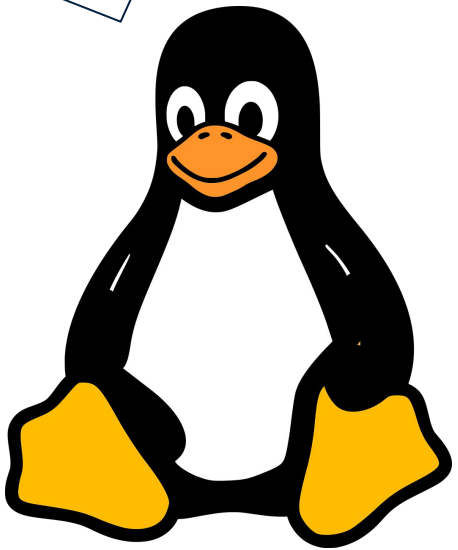
FFFFFFFF

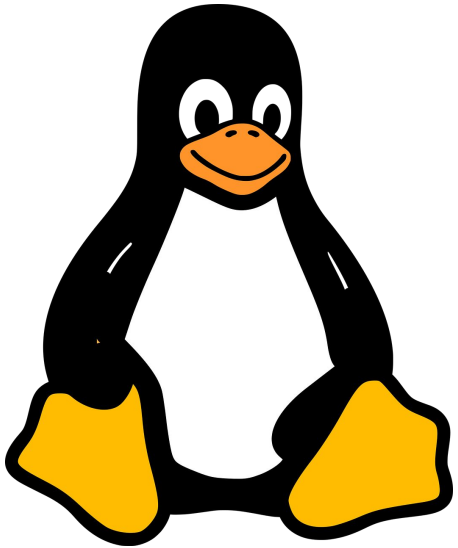


malloc(128)

Okay I'll see what I can do







128 bytes

0000000

....

....

....

....

....

....

....

....

....

....

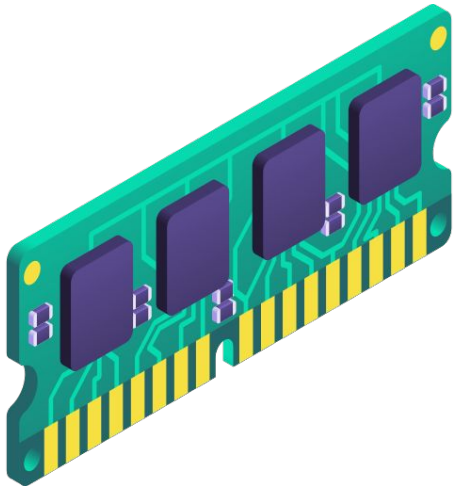
....

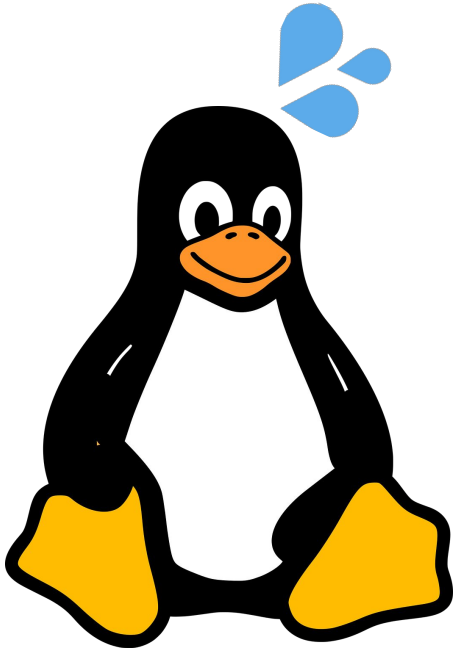
....

....

....

FFFFFFFF





1000 bytes

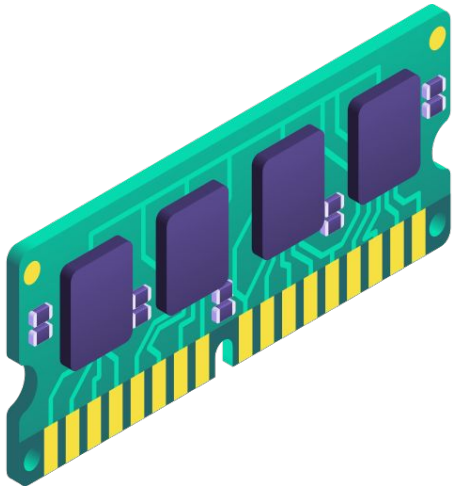
128 bytes

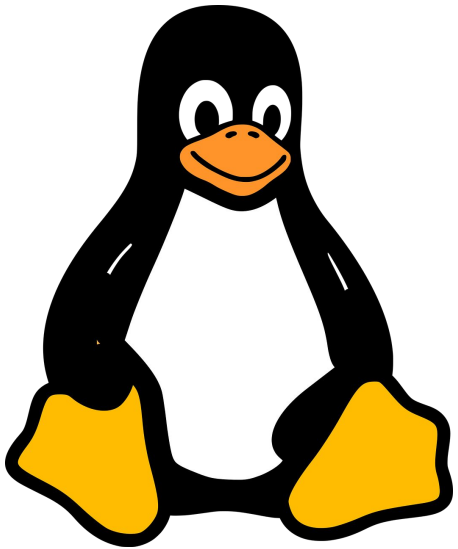
10 bytes

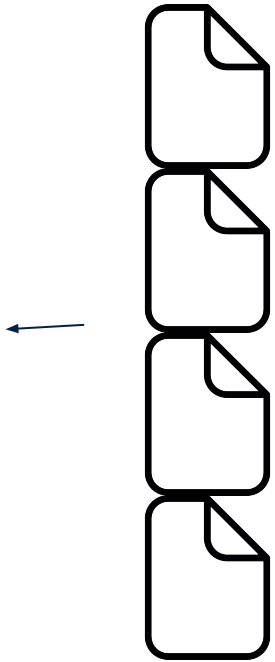
52 bytes

0000000

FFFFFFFF







4 KiB

0000000

....

....

....

....

....

....

....

....

....

....

....

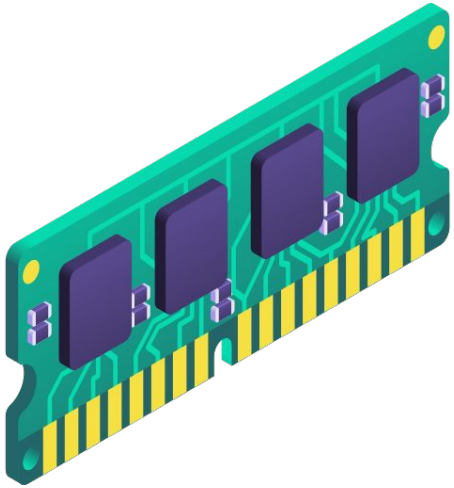
....

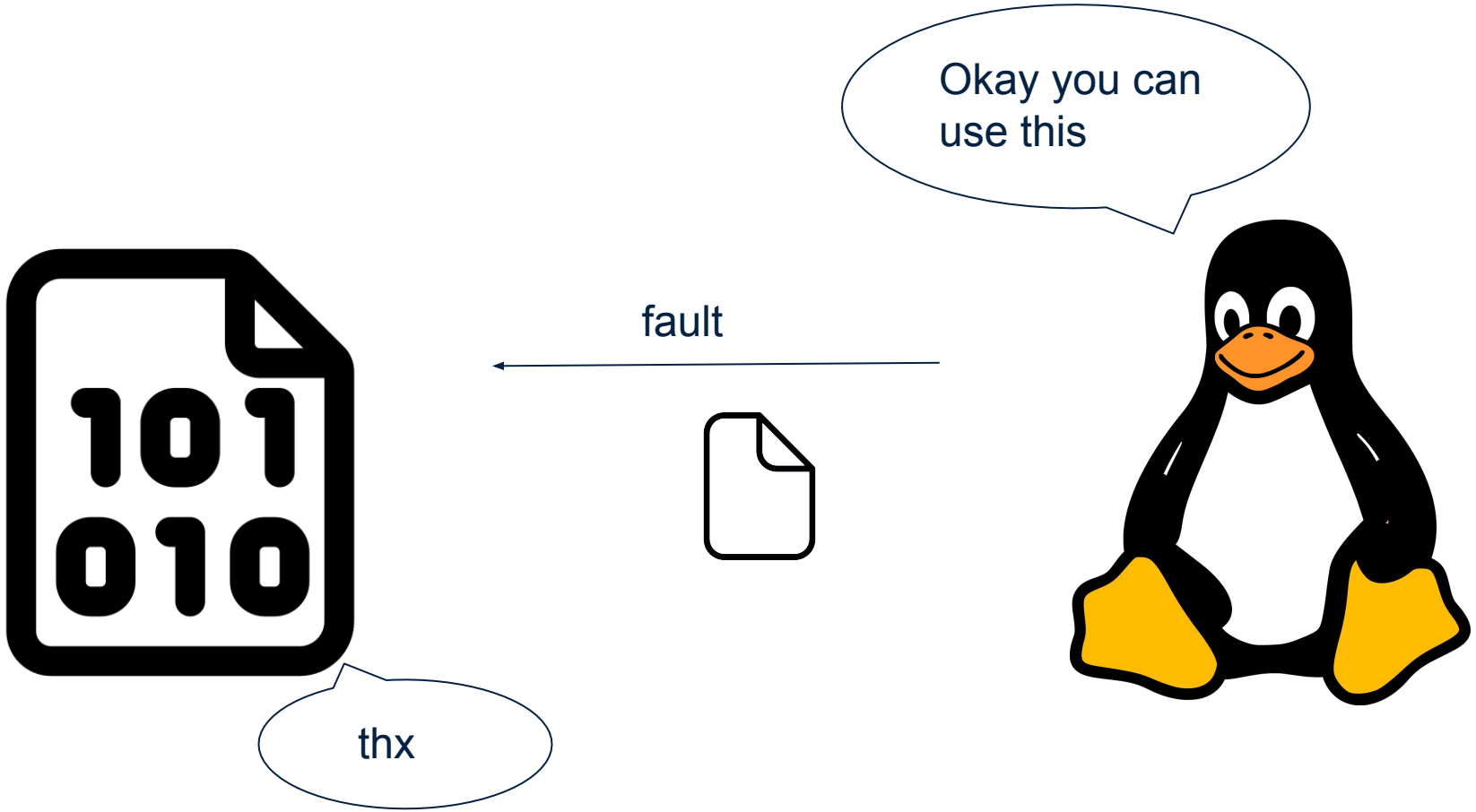
....

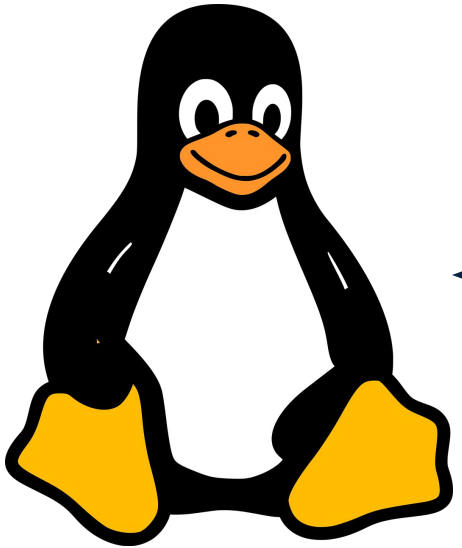
....

....

FFFFFFFF

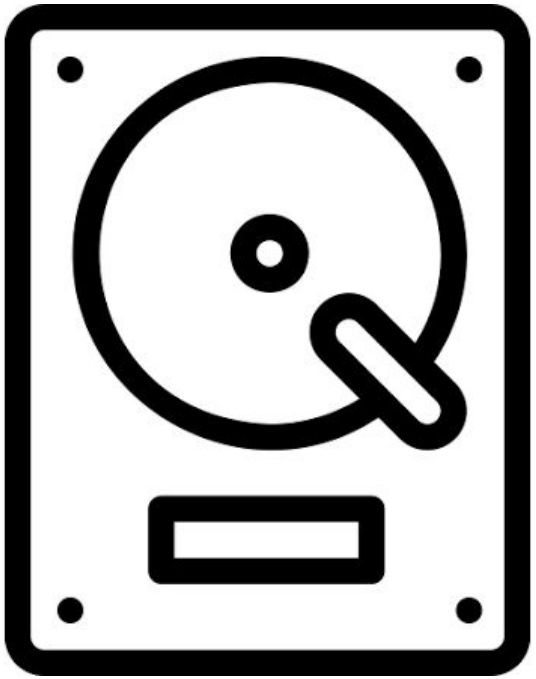


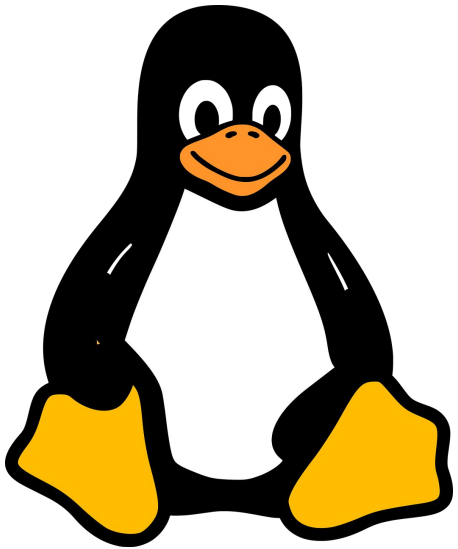

Okay you can use this
fault
101
010
thx

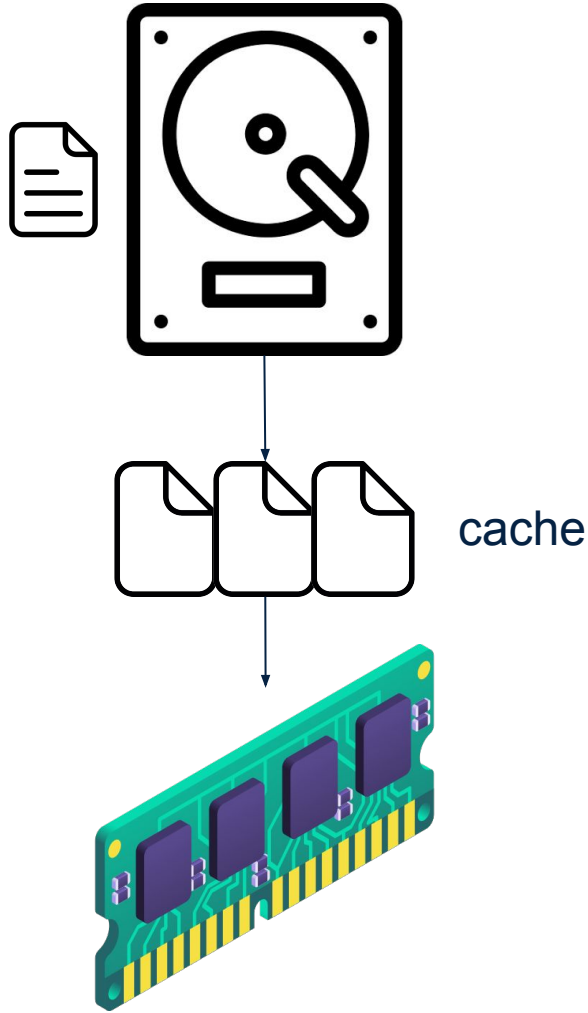


slow…

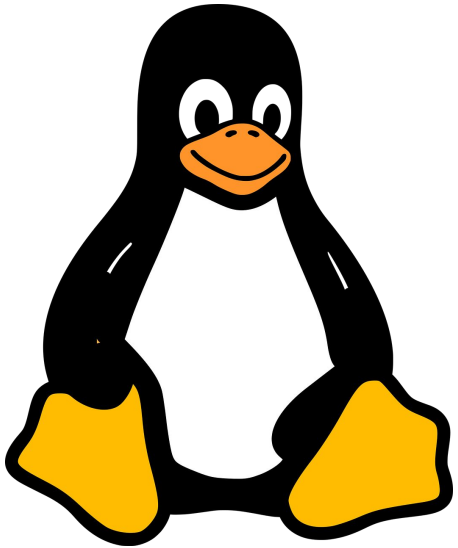




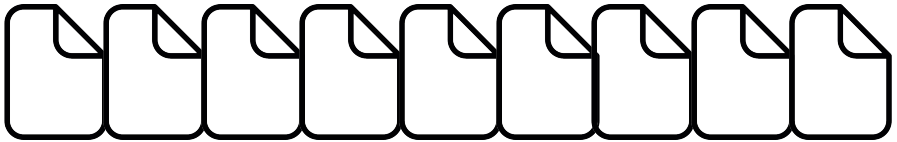




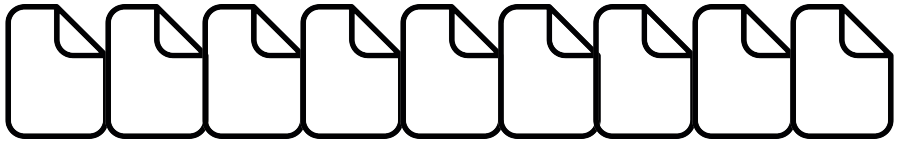
fast :)



Anonymous Pages
(contain arbitrary data from programs)



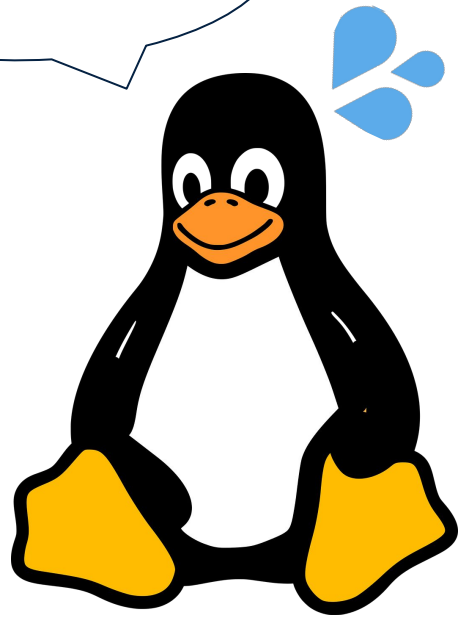
Page Cache
(contains data cached from files on disk)

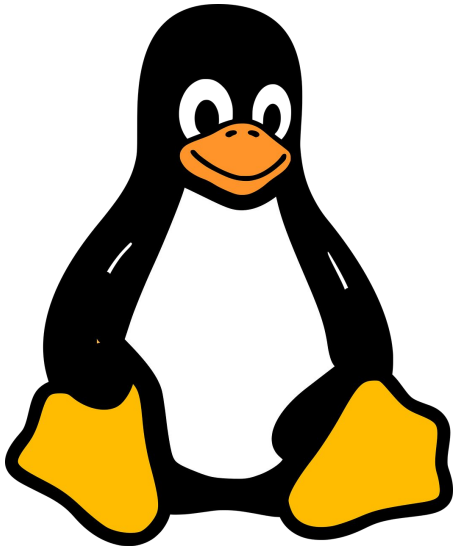


malloc(128)

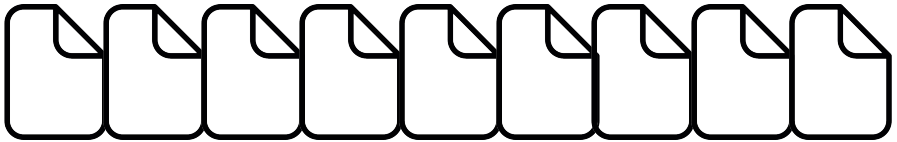
uuuuhhhhhh



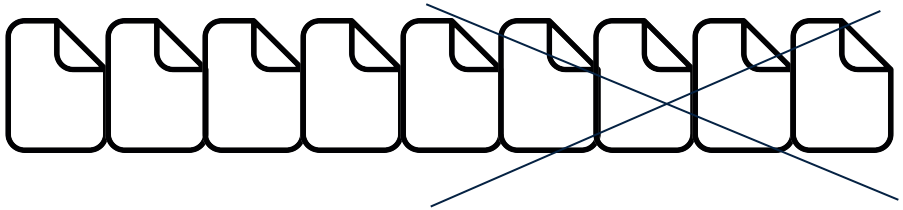




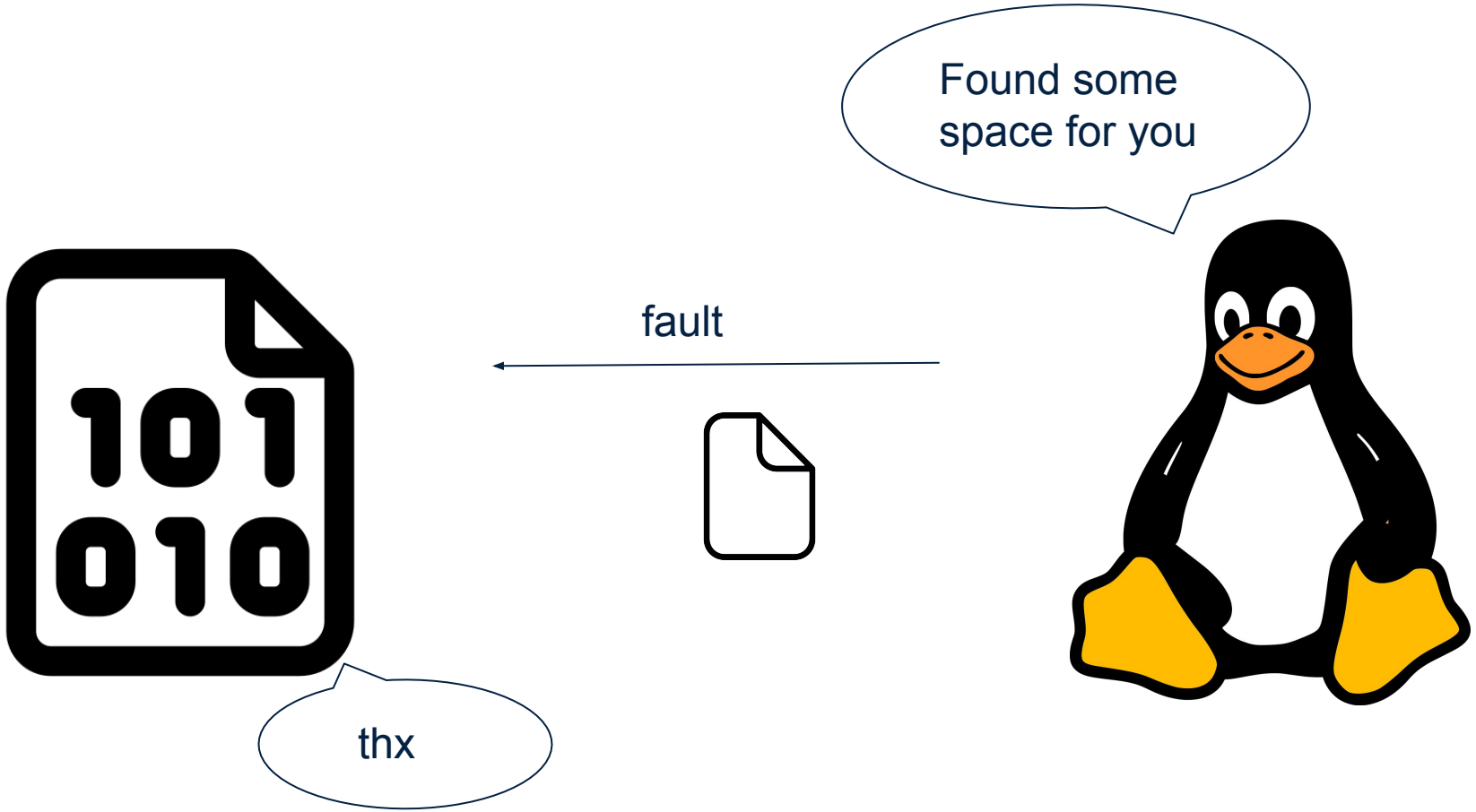
Anonymous Pages



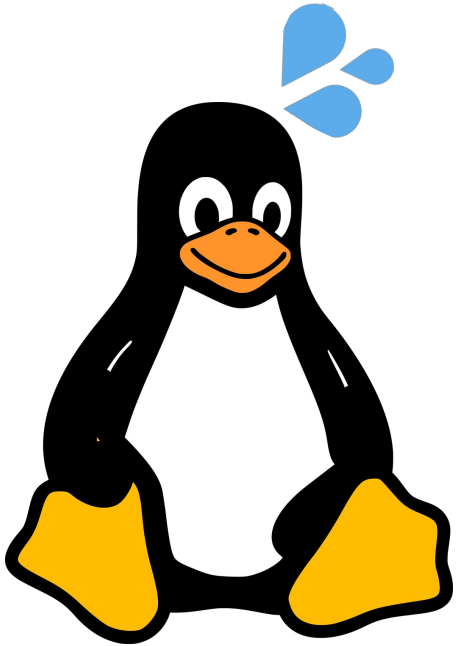
Page Cache



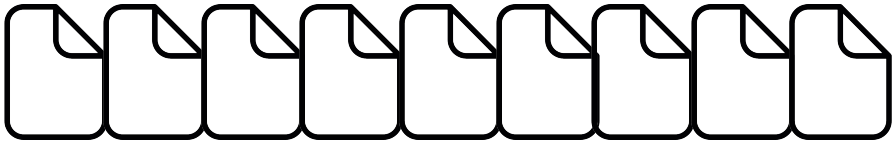
Reclaimed
(we can re-read this data from the file if we need it)


Found some space for you
fault
101
010
thx

Can’t reclaim anything!



Anonymous Pages



Page Cache


No hard feelings
101
010
OOM Kill
Oh dear

Example Memory Map

Memory mappings

KubeCon CloudNativeCon North America 2026

```
$ pmap $(pidof otelcontribcol)
0000000000400000 164740K r-x-- otelcontribcol
000000000a4e1000 240308K r---- otelcontribcol
0000000018f8e000   3648K rw--- otelcontribcol
000000001931e000  33504K rw---   [ anon ]
00003d10f8000000  53248K -----   [ anon ]
...
00007be1c6aba000      8K r----   [ anon ]
00007be1c6abc000      8K r-x--   [ anon ]
00007fff13aff000    136K rw---   [ stack ]
ffffffffff600000      4K --x--   [ anon ]
 total          1740652K
```

Pages to Functions

ELF Segments to Pages

KubeCon CloudNativeCon North America 2026

```
$ readelf -lW otelcontribcol

Program Headers:
  Type  Offset     VirtAddr           FileSiz    MemSiz     Flg
  LOAD  0x000000   0x0000000000400000 0xa0d47f1  0xa0d47f1  R E
  LOAD  0xa0d5000  0x000000000a4d5000 0xea94de8  0xea94de8  R
  LOAD  0x18b6a000 0x0000000018f6a000 0x38f1a0   0x2447120  RW

 Section to Segment mapping:
  Segment Sections...
   02     .note.go.buildid .note.gnu.build-id .text
   03     .rodata .gopclntab .typelink .itablink
   04     .go.buildinfo .go.fipsinfo .go.module .noptrdata .data .bss .noptrbss
```

40.3%

```
$ objdump -d -j .text --no-show-raw-insn otelcontribcol
otelcontribcol:     file format elf64-x86-64
Disassembly of section .text:

0000000000b3a1e0:
  b3a1e0:   cmp    0x10(%r14),%rsp
  b3a1e4:   jbe    b3a2a0
  b3a1ea:   push   %rbp
  b3a1eb:   mov    %rsp,%rbp
  b3a1ee:   call   b3a2e0
  b3a1f3:   test   %rax,%rax
```

```
func (r *otlpReceiver) Start(ctx context.Context,
host component.Host) error {
      err := r.startGRPCServer(ctx, host);
      ...
}

func (r *otlpReceiver) startGRPCServer(host
component.Host) error {
      ...
}
```

go.opentelemetry.io/collector/receiver/otlpreceiver

```
$ go tool objdump -s 'otlpreceiver*$' otelcontribcol
...
c2a760     80 T otlpreceiver.(*otlpReceiver).startHTTPServer.func1
c2a7c0    699 T otlpreceiver.(*otlpReceiver).Start
c2aa80    165 T otlpreceiver.(*otlpReceiver).Shutdown
c2ab40    535 T otlpreceiver.handleTraces
c2ad90   2349 T otlpreceiver.(*otlpReceiver).startGRPCServer
```



```
$ go tool nm -size -sort address otelcontribcol | rg '.init()$'
 1aefbe0        147 T github.com/aws/aws-sdk-go-v2/internal/rand.init
 1aefc80         75 T github.com/aws/aws-sdk-go-v2/internal/rand.init.0
 a343ce0        202 T github.com/vmware/govmomi/vsan/types.init.817
 a4b1ca0         61 T crypto/internal/fips140/bigmod.init.0
 a4b2e40       1445 T crypto/internal/fips140/ecdsa.init
 a4c5580        176 T crypto/internal/fips140/ed25519.init
...
```

11,493 **init()** functions

2,006 **packages**

4,355 **pages** of .text (17 MiB)

KubeCon CloudNativeCon
North America 2026

Demo

Collector under pressure



THE METRICS

system.memory.usage != k8s.node.memory.usage



Includes reclaimable

k8s.{pod/container}.memory.usage



memory.current (cgroup file)



OOM kill (Linux)

k8s.{pod/container}.memory.working_set



Required memory



Kubelet eviction

*.memory.pressure.stall_time



memory.reclaim

k8s.{pod/container}.memory.rss



Heap/stack (anon)



Memory leaks

The Core Takeaway

KubeCon CloudNativeCon
North America 2026

Q & A

Thank you!

Demo Source

