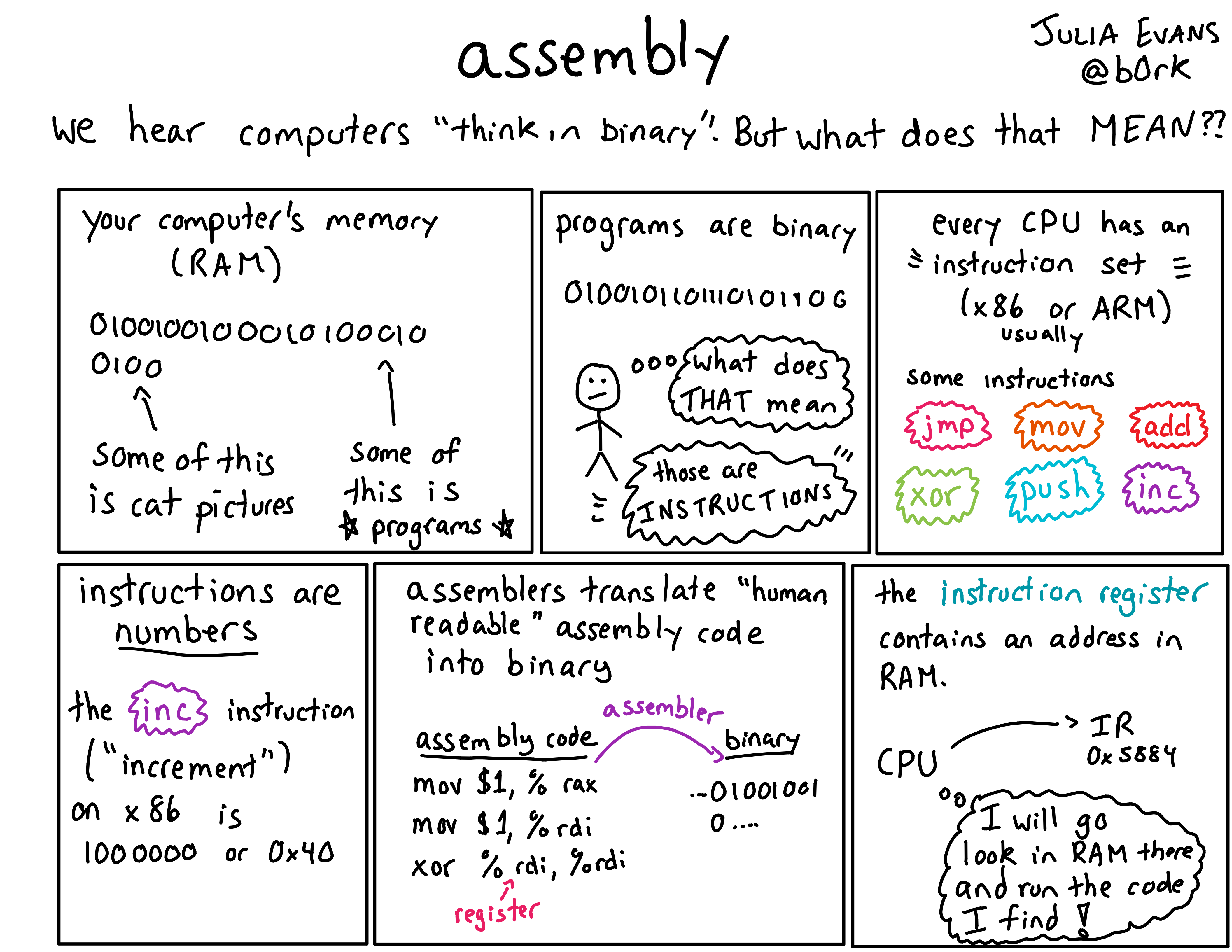
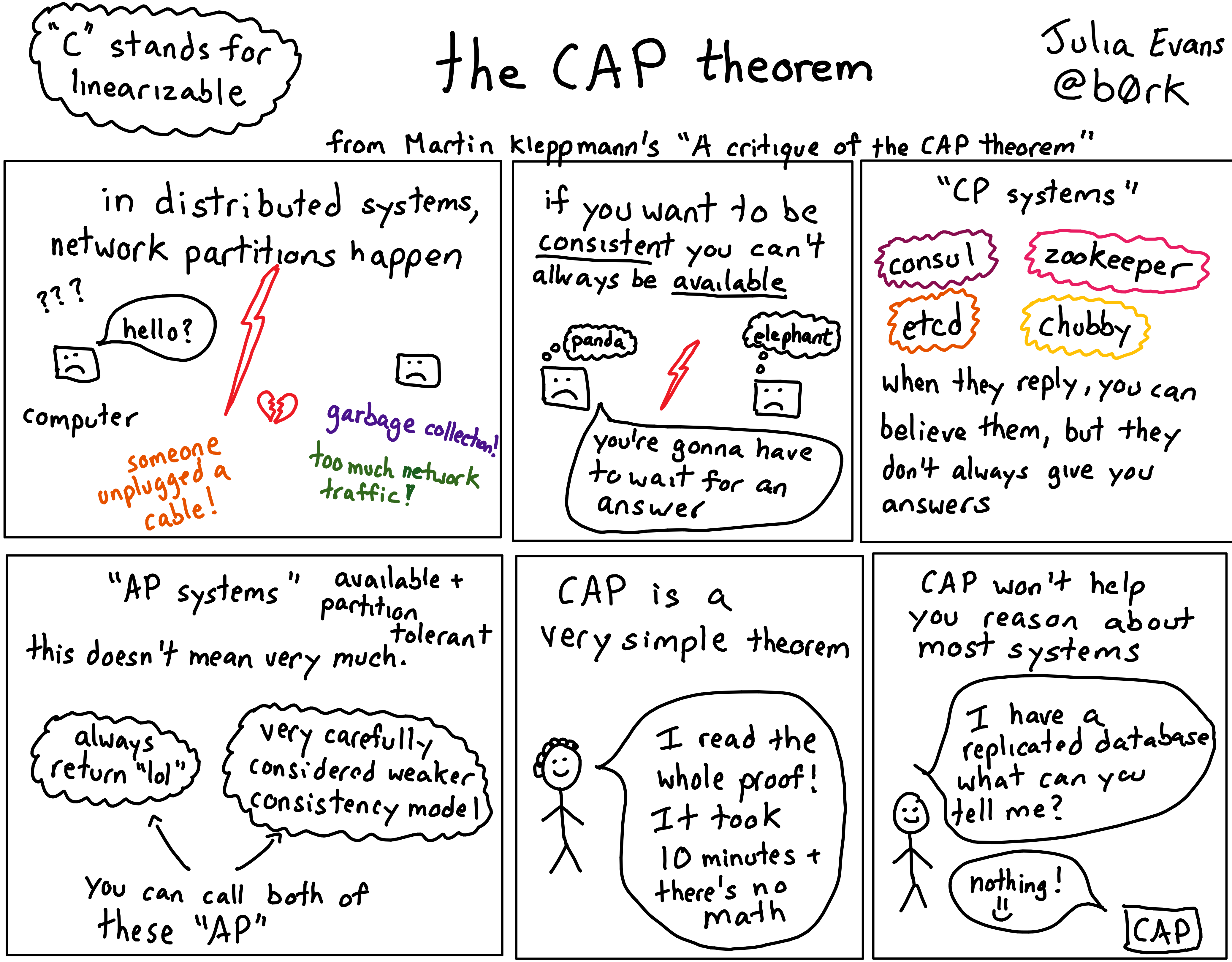


assembly
JULIA EVANS
@b0rk
we hear computers "think in binary". But what does that MEAN??
your computer's memory (RAM)
01001001000101000100100
Some of this is cat pictures
some of this is ☆ programs ☆
programs are binary
01001011011101011106
ooo what does THAT mean
"those are INSTRUCTIONS"
every CPU has an instruction set (x86 or ARM) usually
some instructions
jmp
mov
add
xor
push
inc
instructions are numbers
the inc instruction ("increment") on x86 is 1000000 or 0x40
assemblers translate "human readable" assembly code into binary
assembly code
assembler
binary
mov $1, % rax
mov $1, % rdi
xor %rdi, %rdi
register
...01001001
0....
the instruction register contains an address in RAM.
CPU
IR
0x5884
I will go look in RAM there and run the code I find!

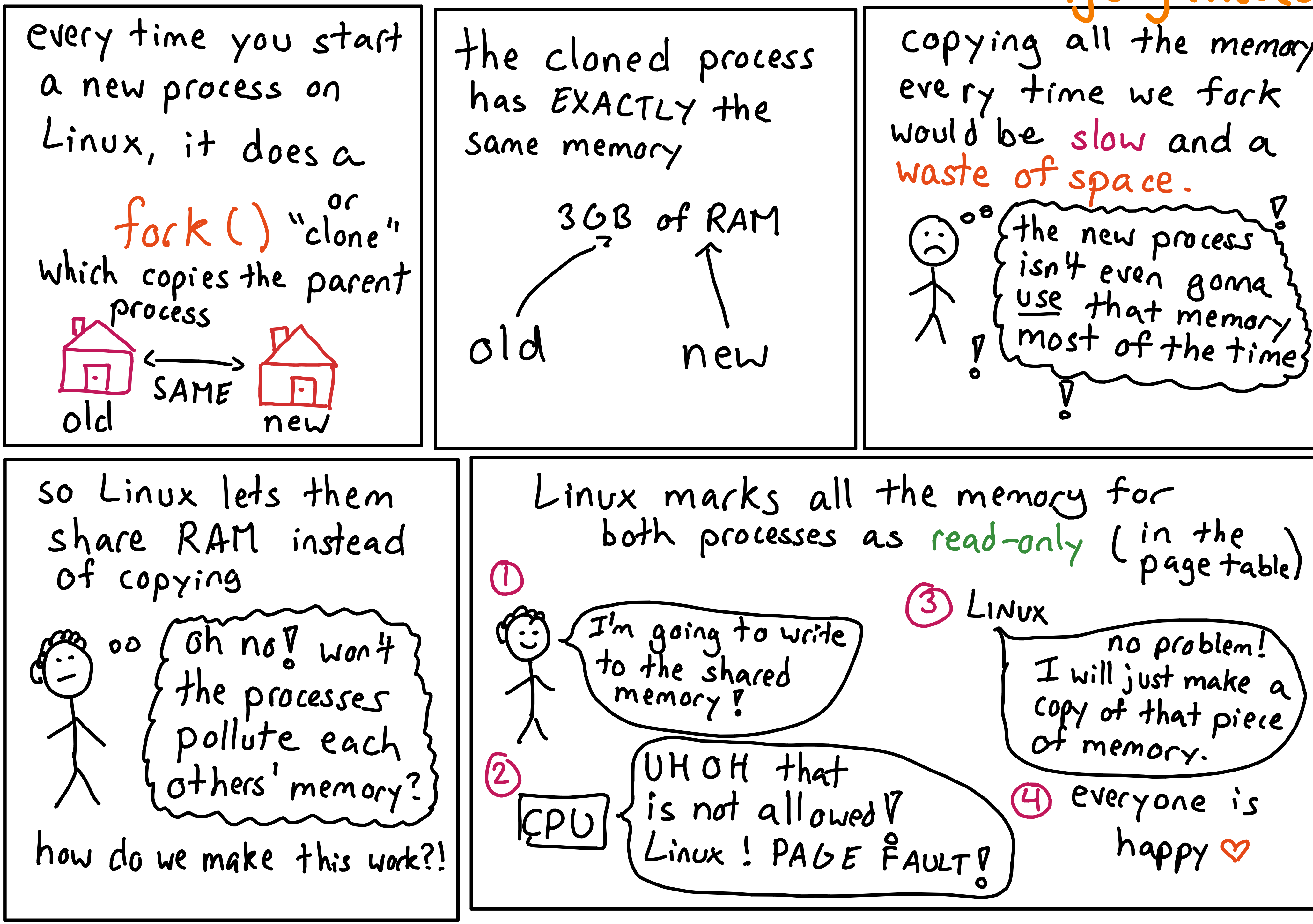


"C" stands for linearizable
the CAP theorem
Julia Evans @b0rk
from Martin Kleppmann's "A critique of the CAP theorem"
in distributed systems, network partitions happen
???
hello?
computer
someone unplugged a cable!
garbage collection!
too much network traffic!
if you want to be consistent you can't always be available
panda
elephant
you're gonna have to wait for an answer
"CP systems"
consul
zookeeper
etcd
chubby
when they reply, you can believe them, but they don't always give you answers
"AP systems" available + partition tolerant
this doesn't mean very much.
always return "lol"
very carefully considered weaker consistency model
You can call both of these "AP"
CAP is a very simple theorem
I read the whole proof! It took 10 minutes + there's no math
CAP won't help you reason about most systems
I have a replicated database what can you tell me?
nothing!
CAP

JULIA EVANS
@b0rk

copy on write

drawings.jvns.ca

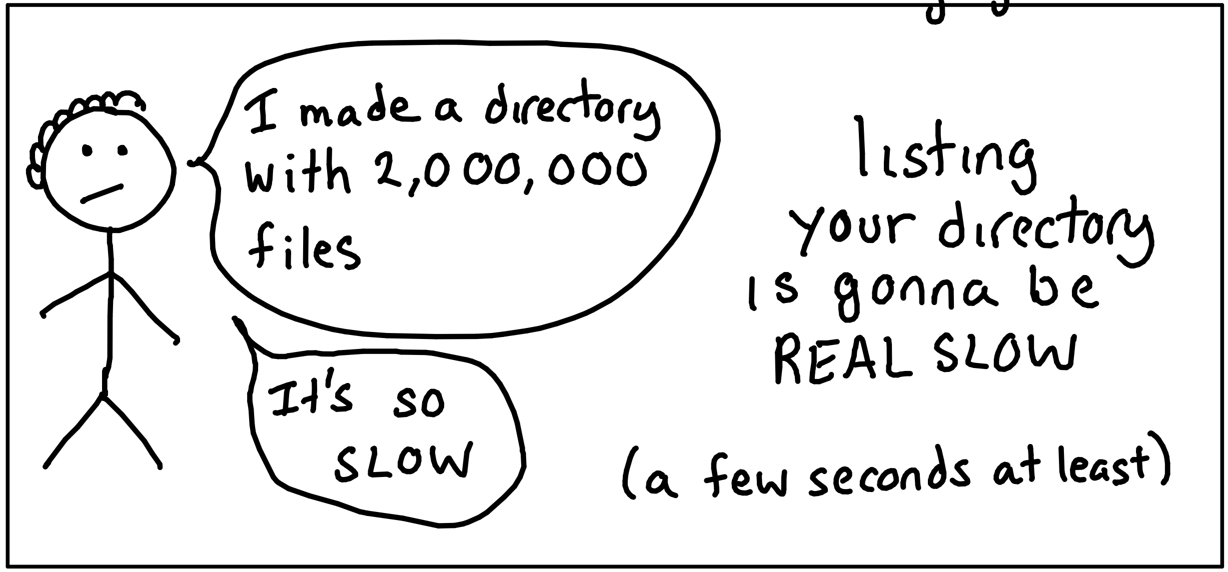


directories + symlinks

@b0rk
Julia Evans
drawings.jvns.ca

What's a directory?

filename	inode number
awesome.jpg	279932
blah.txt	13227
cumberbatch	233333

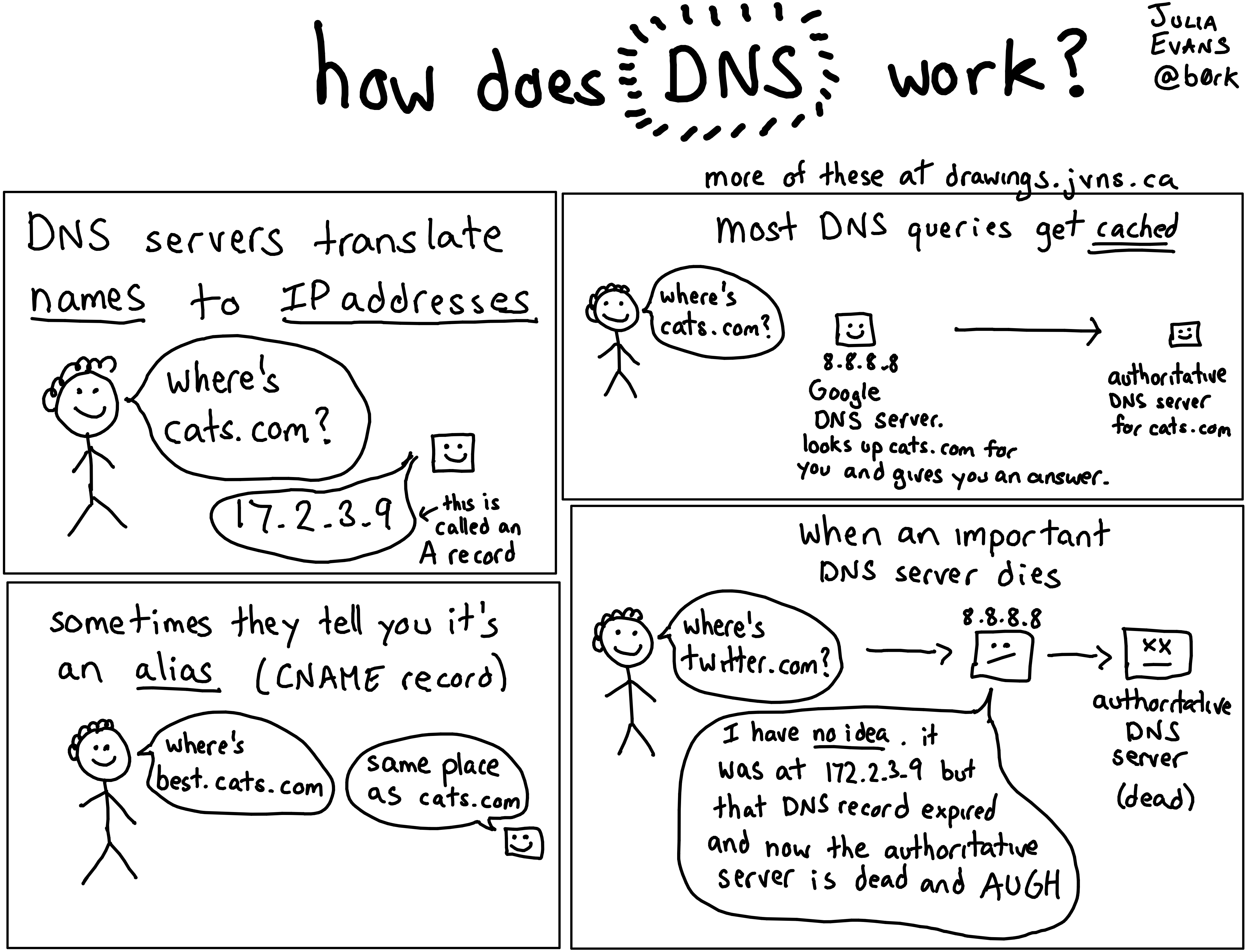


what's a symlink?

it's just a file with the name of another file in it!

$ readlink my-cool-link

/home/julia/long-complicated-file-name



how does DNS work?
JULIA EVANS @b0rk
more of these at drawings.jvns.ca
DNS servers translate names to IP addresses
where's cats.com?
17.2.3.9
this is called an A record
most DNS queries get cached
where's cats.com?
8.8.8.8
Google DNS server. looks up cats.com for you and gives you an answer.
authoritative DNS server for cats.com
sometimes they tell you it's an alias (CNAME record)
where's best.cats.com
same place as cats.com
when an important DNS server dies
where's twitter.com?
8.8.8.8
xx
authoritative DNS server (dead)
I have no idea. it was at 172.2.3.9 but that DNS record expired and now the authoritative server is dead and AUGH

floating point

JULIA EVANS
@b0rk

more at: drawings.jvns.ca

a double is 64 bits.

that means there are 2^{64} different doubles

going up to $\boxed{1.8 \times 10^{308}}$

some double arithmetic

$2^{52} + 0.2 = 2^{52}$ (the next number after 2^{52} is $2^{52}+1$)

$1 + \frac{1}{2^{53}} = 1$ (the next number after 1 is $1 + \frac{1}{2^{52}}$)

3×10^{100} = infinity ← infinity is a double

infinity - infinity = nan (not a number)

there are 2^{52} numbers between 1 and 2

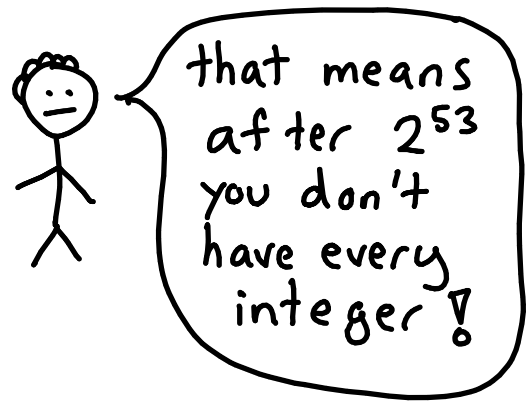
$1 + \frac{1}{2^{52}}, 1 + \frac{2}{2^{52}}, \ldots$

2^{51} numbers between 2+4

2^{50} between 4 and 8

etcetera.

Javascript only has doubles (and Lua!)

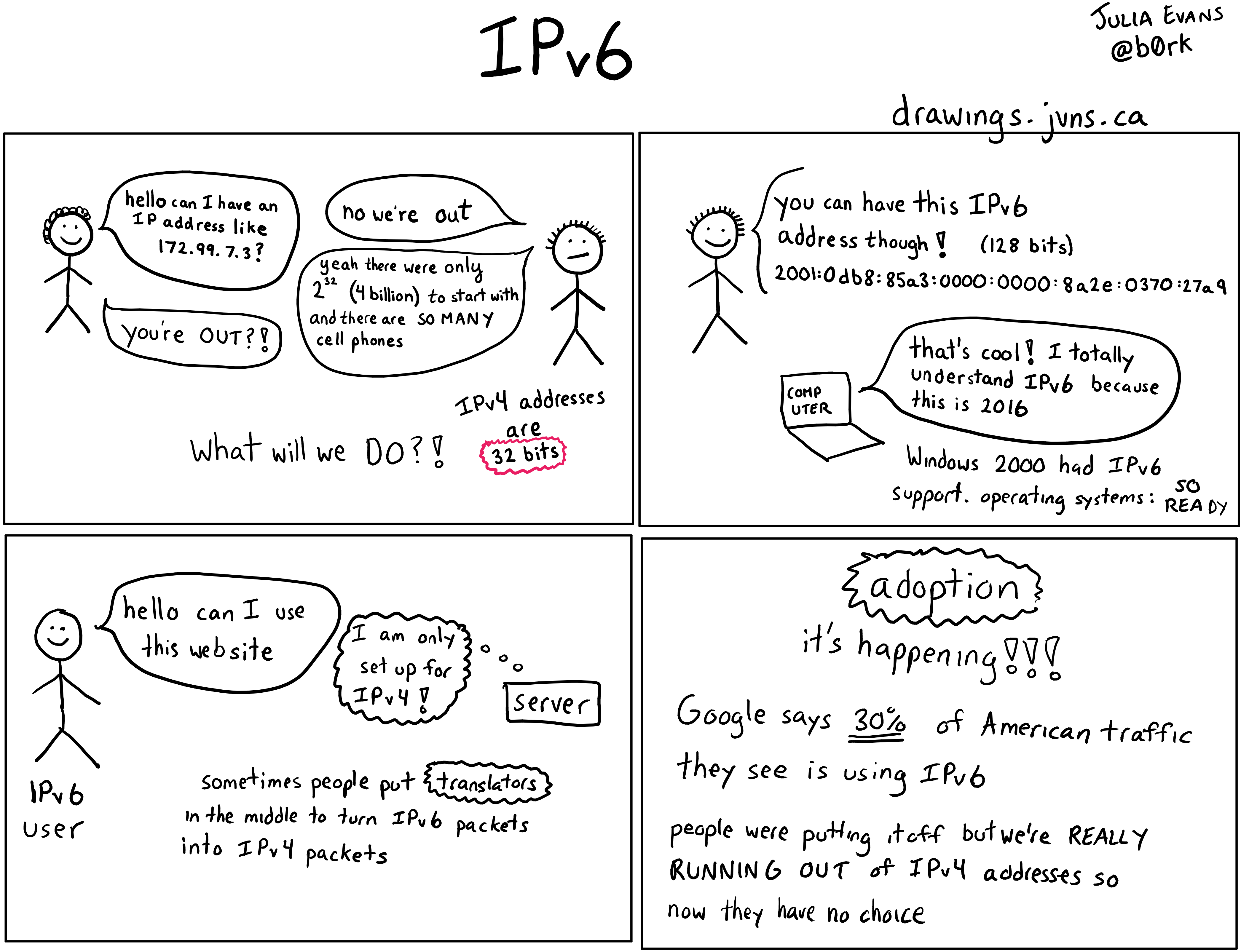


printing doubles is nontrivial

the shortest version of 25.648538980428e8

is 2.564854e9

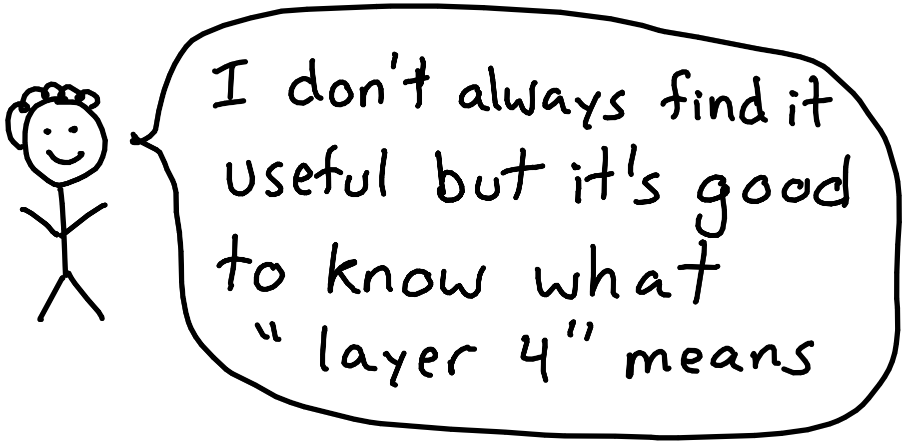
↑ calculating the shortest representation takes time!

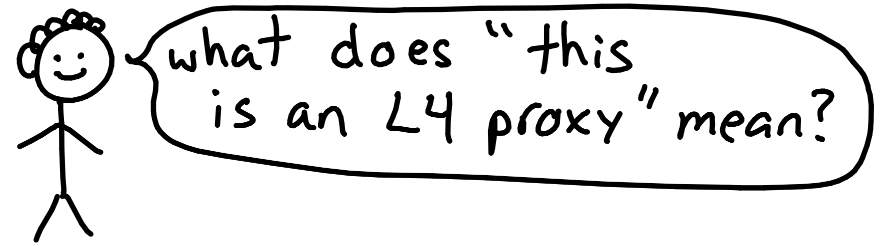


IPv6
JULIA EVANS
@b0rk
drawings.jvns.ca
hello can I have an IP address like 172.99.7.3?
no we're out
you're OUT?!
yeah there were only 2^32 (4 billion) to start with and there are SO MANY cell phones
IPv4 addresses are 32 bits
What will we DO?!
you can have this IPv6 address though! (128 bits)
2001:0db8:85a3:0000:0000:8a2e:0370:27a9
COMPUTER
that's cool! I totally understand IPv6 because this is 2016
Windows 2000 had IPv6 support. operating systems: SO READY
hello can I use this website
I am only set up for IPv4!
server
IPv6 user
sometimes people put translators in the middle to turn IPv6 packets into IPv4 packets
adoption
it's happening!!!
Google says 30% of American traffic they see is using IPv6
people were putting it off but we're REALLY RUNNING OUT of IPv4 addresses so now they have no choice

the "OSI model" for networking

JULIA EVANS
@b0rk





LAYERS

1: electrical engineering stuff, wires, frequencies, wifi

2: Ethernet protocol + others

3: IP (IP addresses)

4: TCP + UDP (ports)

5+6: nobody ever talks about these

7: HTTP and friends

If a load balancer is labelled "L7" it usually means it looks at the Host: header inside your HTTP packets.

layer 3 networking tool

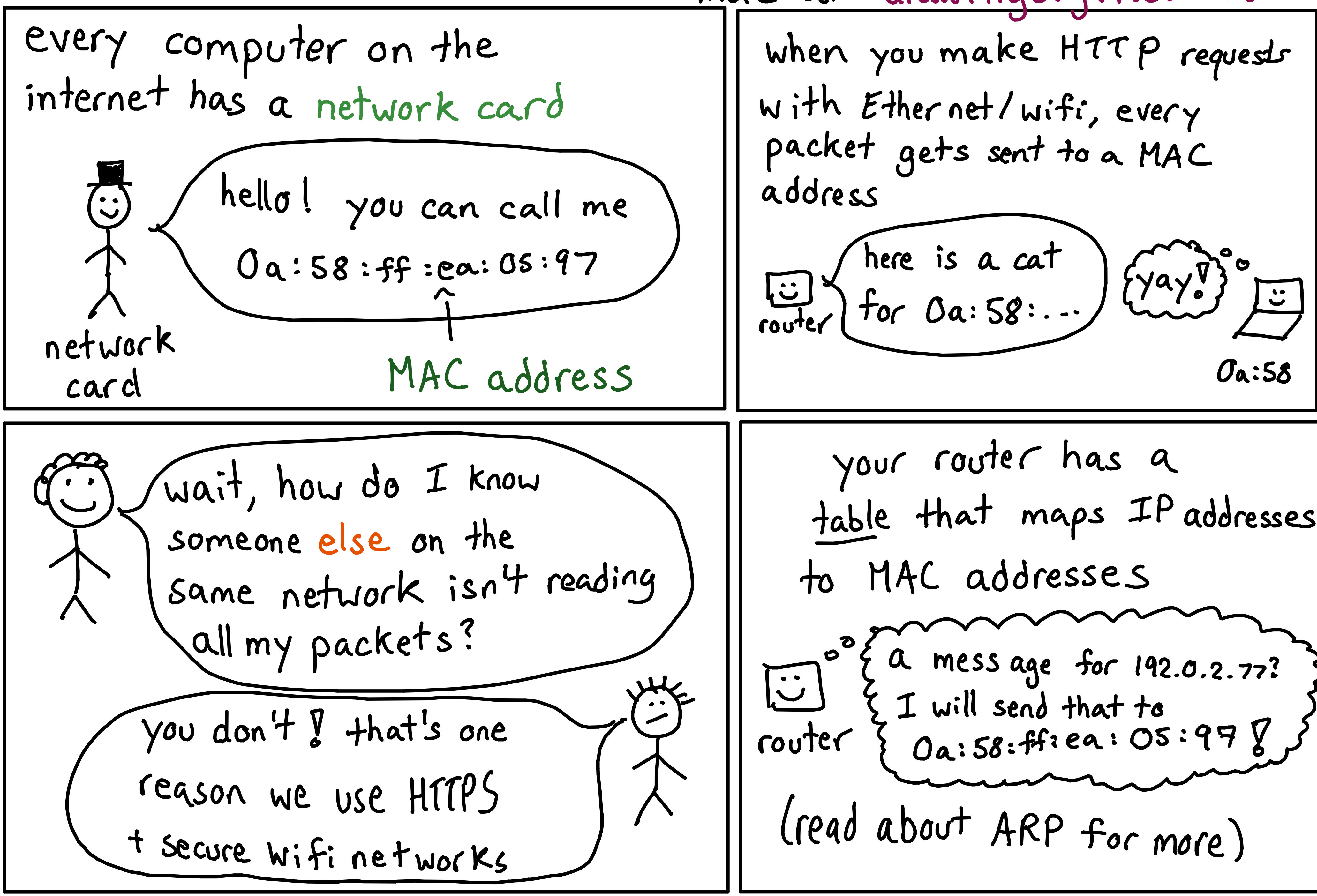
↑ ignores layer 4 and above

I only know about IP addresses! I don't even know what a port is let alone what the packet says

Julia Evans
@b0rk

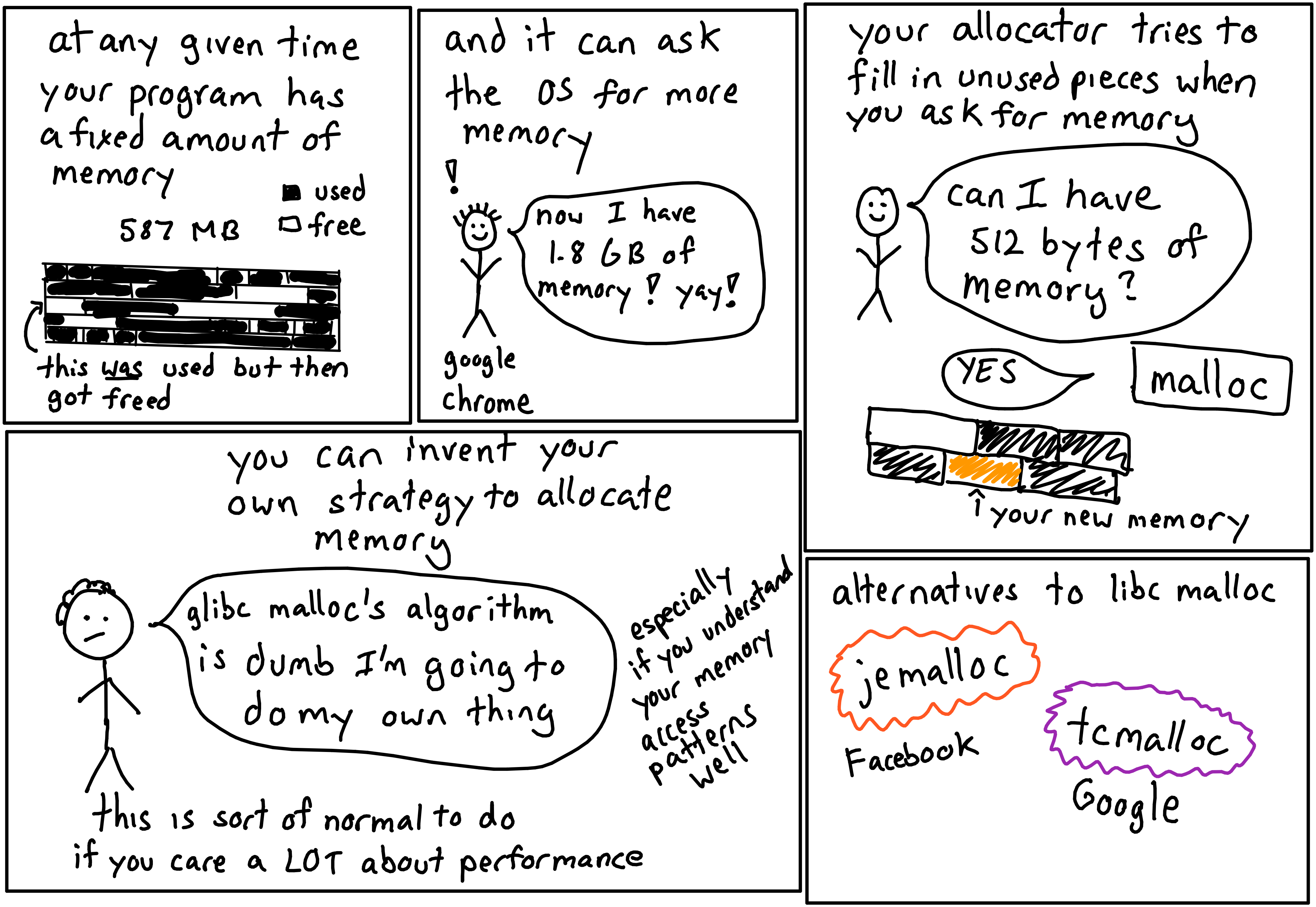
what's a MAC address?

more at: drawings.jvns.ca!



memory allocation

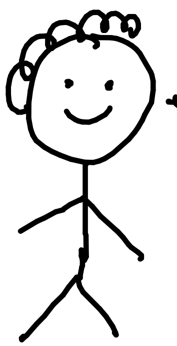
Julia Evans
@b0rk

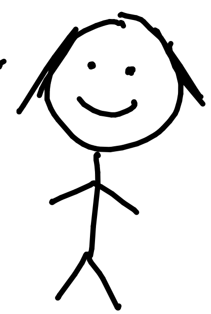


man pages = awesome

(sometimes. Quality may vary ☺)

JULIA EVANS
@b0rk





man pages are split up into <u>sections</u>

1 2 3 4 5 6 7 8

/usr/share/man/man5 has section 5 on my machine.

1. programs

   $ man grep
   $ man ls

2. system calls

   $ man sendfile

3. C functions

   $ man 3 printf
   $ man fopen

4. devices

   $ man null
   for /dev/null docs

5. file formats

   $ man sudoers
   for /etc/sudoers
   GREAT → $ man proc

6. games

   (not very useful)
   man sl is good if you have sl though

7. miscellaneous

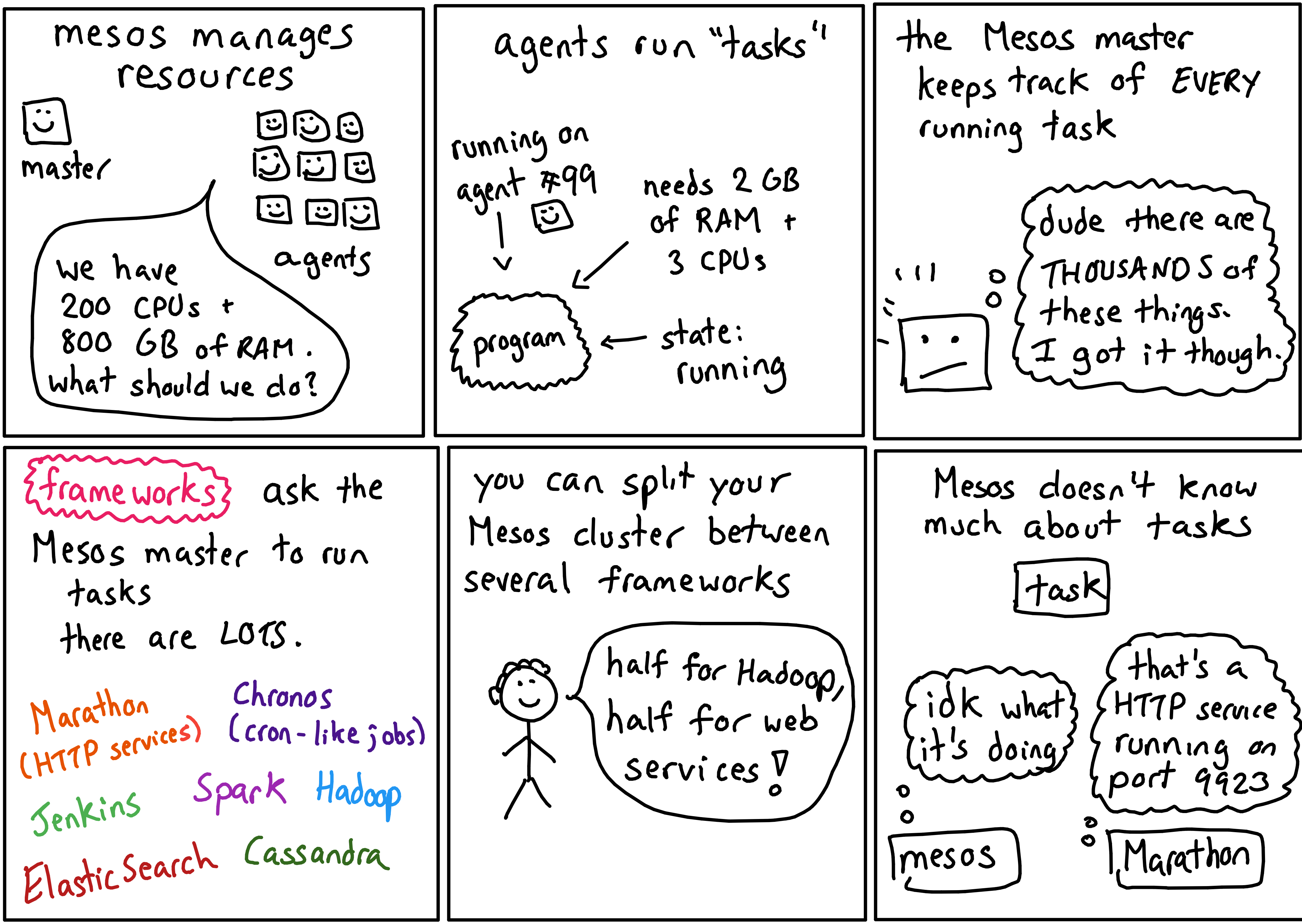
   $ man 7 pipe
   $ man 7 symlink
   (these are cool!)

8. sysadmin programs

   $ man apt
   $ man chroot

mesos

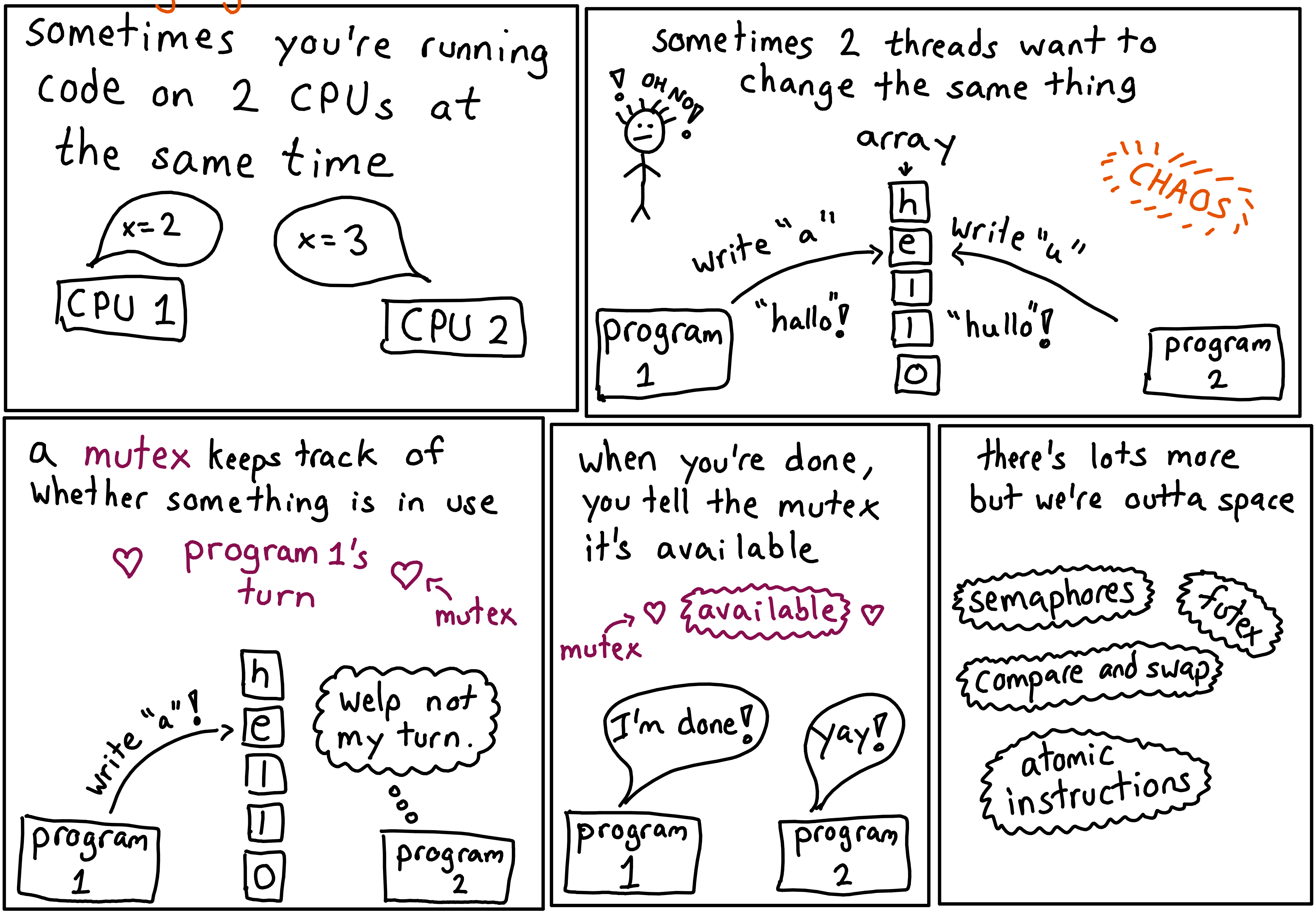
JULIA EVANS
@b0rk

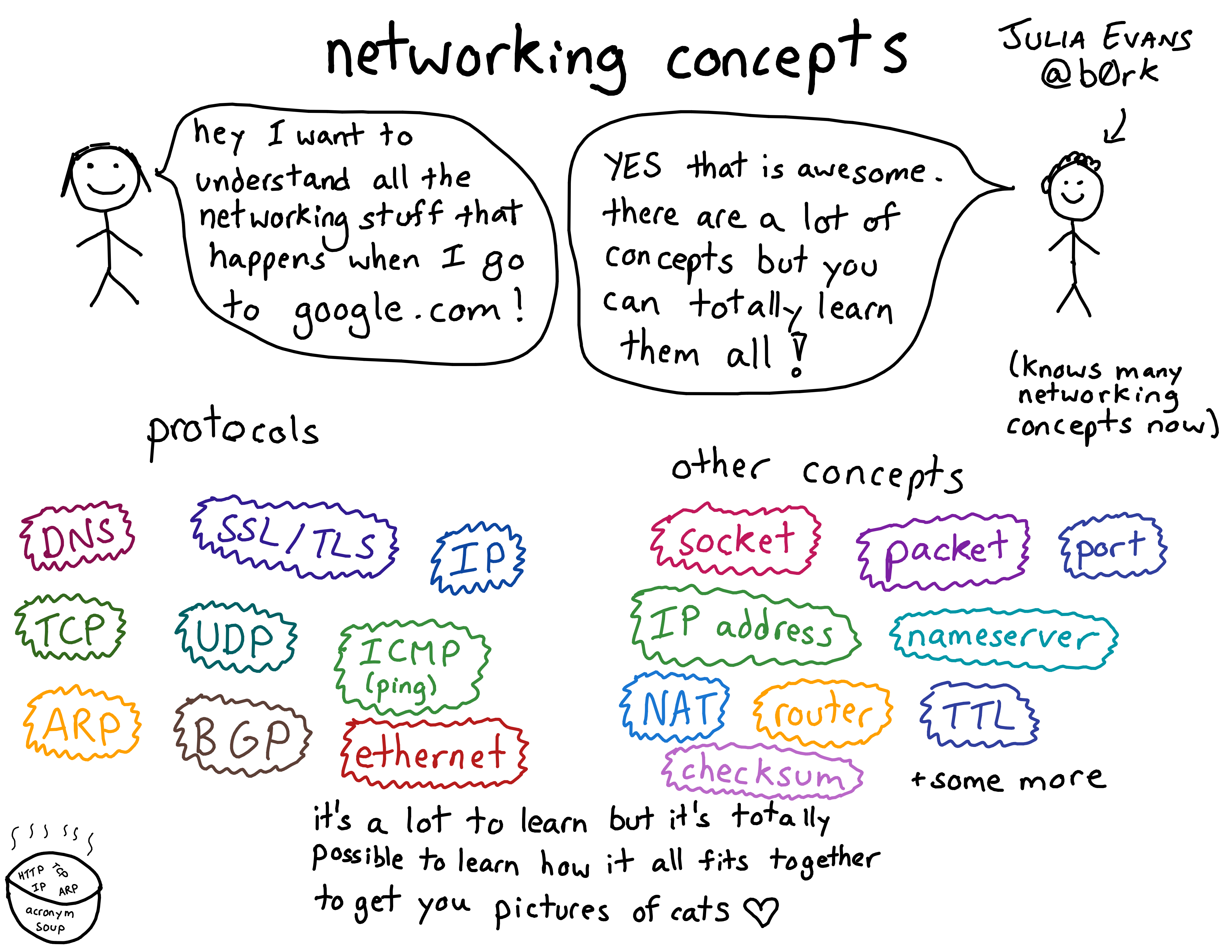


drawings.jvns.ca

mutexes

JULIA EVANS
@b0rk




networking concepts
JULIA EVANS
@b0rk
hey I want to understand all the networking stuff that happens when I go to google.com!
YES that is awesome. there are a lot of concepts but you can totally learn them all!
(knows many networking concepts now)
protocols
DNS
SSL/TLS
IP
TCP
UDP
ICMP (ping)
ARP
BGP
ethernet
other concepts
socket
packet
port
IP address
nameserver
NAT
router
TTL
checksum
+ some more
HTTP
TCP
IP
ARP
acronym soup
it's a lot to learn but it's totally possible to learn how it all fits together to get you pictures of cats ♡

Julia Evans
@b0rk

anatomy of a packet

more at
drawings.jvns.ca

When you get a webpage like Facebook, it comes into your computer in many small packets

Let's see what those look like!

Packets are split into a few sections (or "headers")

ethernet/wifi

82:53:ac:99:2f:33 ← MAC address

← "physical layer". this gets changed constantly as your packet moves between computers.

IP ("internet protocol")

FROM: 172.96.2.3 TO: 123.9.2.32

← in charge of getting your packet to the right server (like an address on an envelope)

TCP (or UDP)

sequence number: 877392 ← counts bytes sent so far

checksum: 8447 ↑ detect corrupted data

from: port 9979 to: port 80

← in charge of preventing data corruption and helping you retry lost packets

video streaming uses UDP instead. UDP does not try to be reliable.

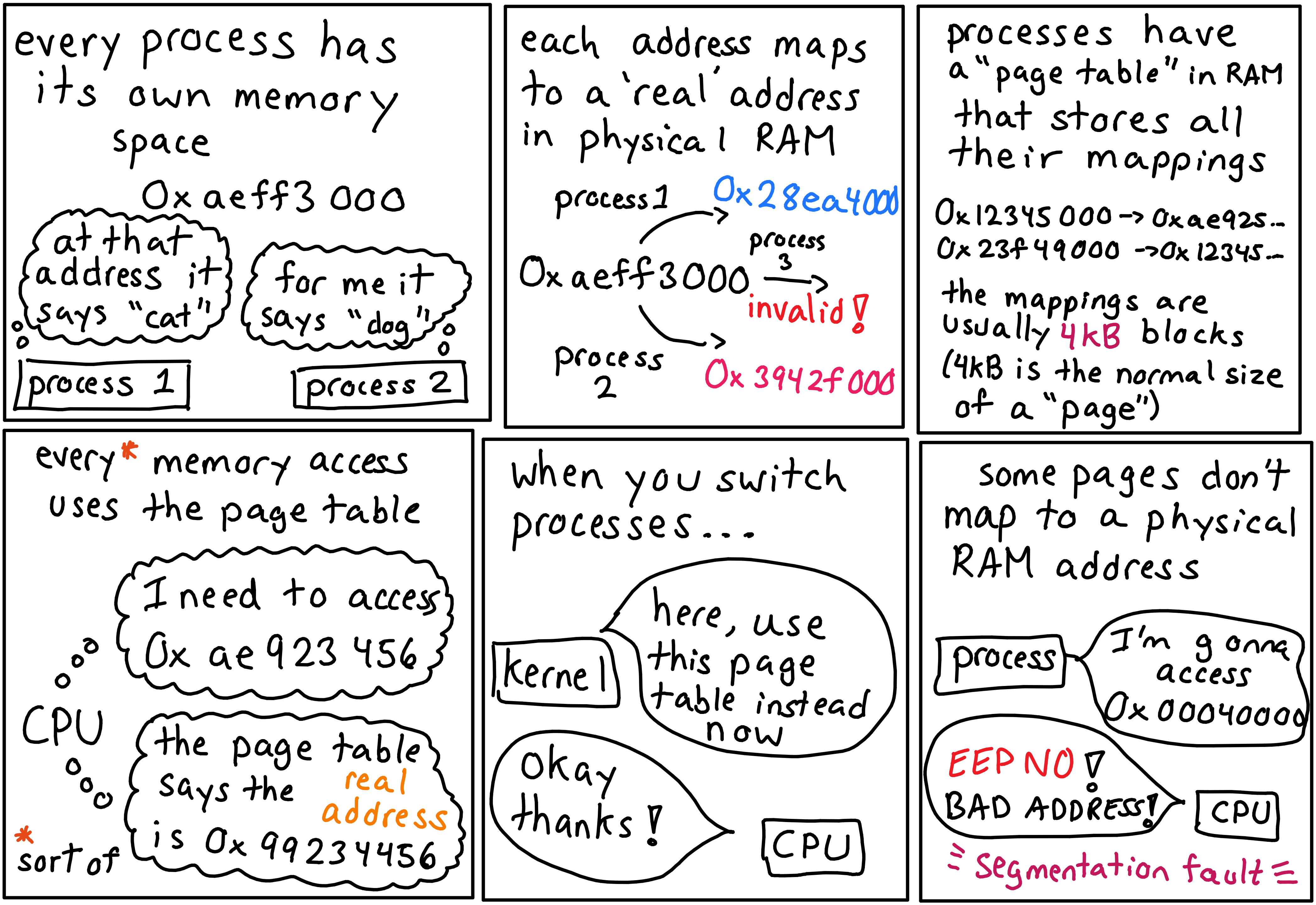
HTTP (or whatever)

GET / HTTP/1.1
Host: google.com
Accept-Language: en-US

← the actual data you're trying to send!

Julia Evans
@b0rk

page table (in 32 bit memory)



unix permissions

Julia Evans
@b0rk

3 kinds of things you can do to a file

read write execute

$ ls -l awesome.png

rw- rw- r-- bork staff

bork can do this (user)

staff can do this (group)

ANYONE can do this

$ ls -l /bin/ping

rwsr-xr-x root root

setuid flag

This means ping always runs as root (who owns it), no matter who started ping

what's this 755 business?

7	means	rwx
6	→	rw-
5	→	r-x
4	→	r--

it's binary!

5 → 101 → r-x

755 means rwx r-x r-x

more weird permissions things

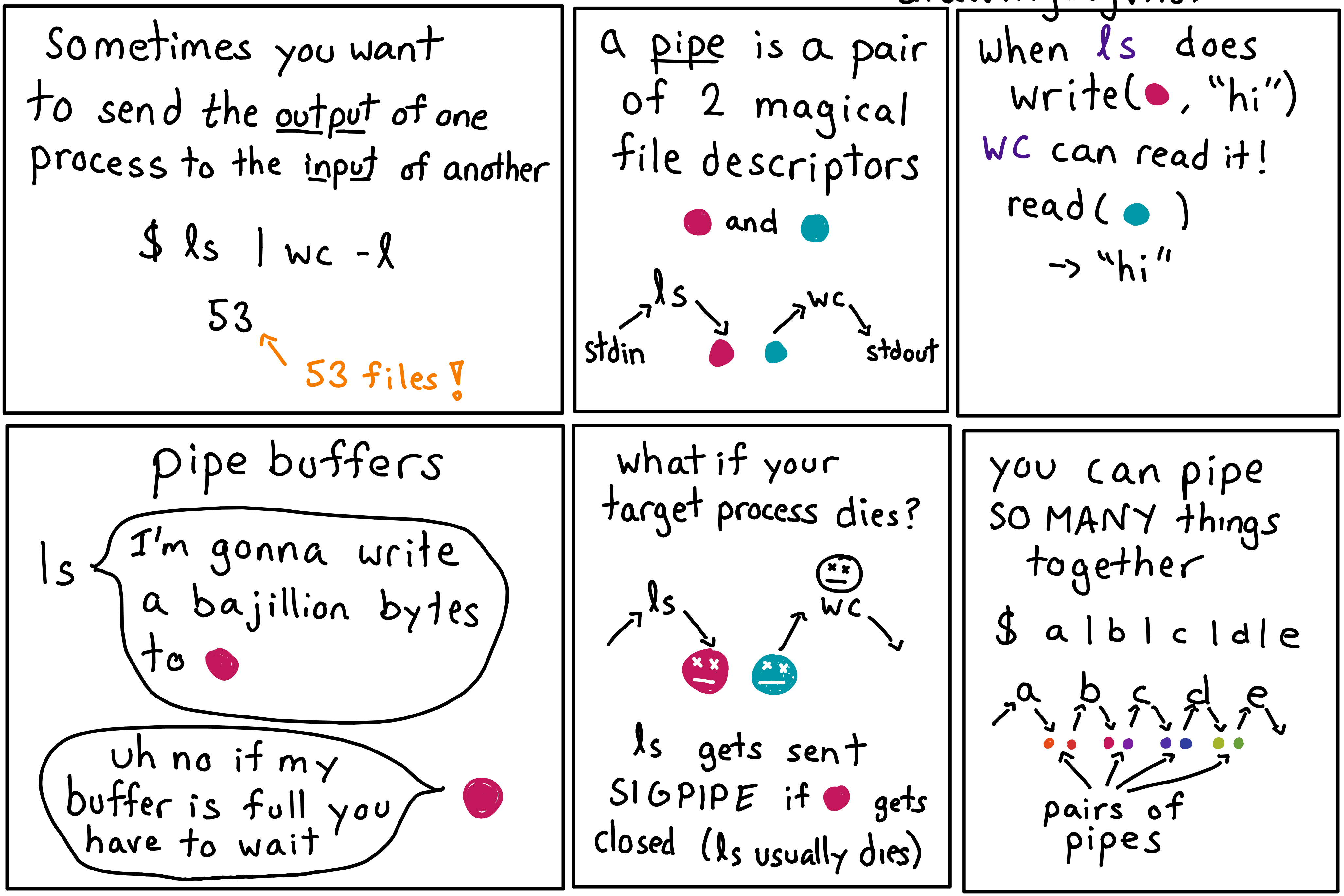
setgid

sticky bit

but I ran out of space

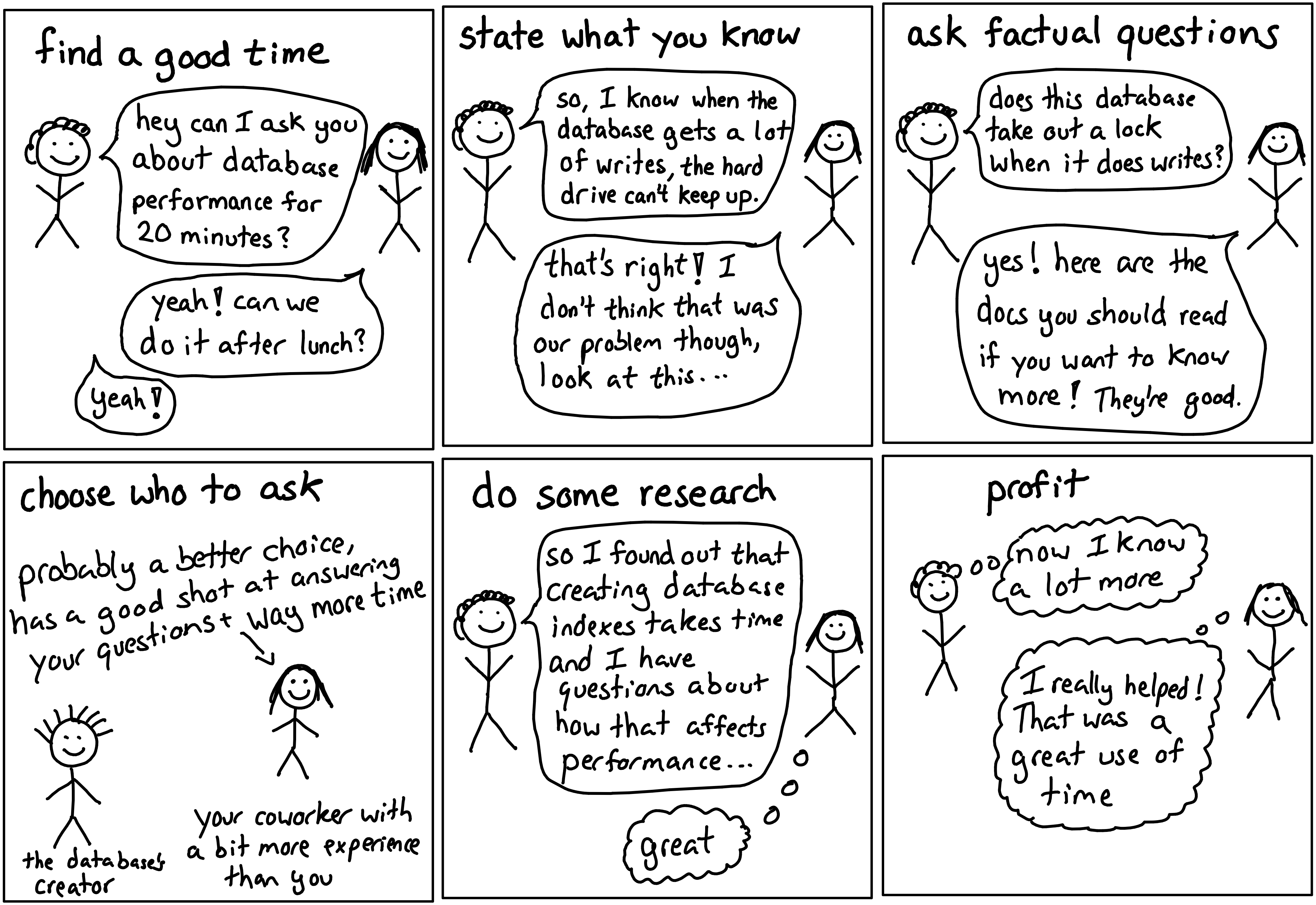
pipes

JULIA EVANS
@bØrk
drawings.jvns.ca



Julia Evans
@b0rk

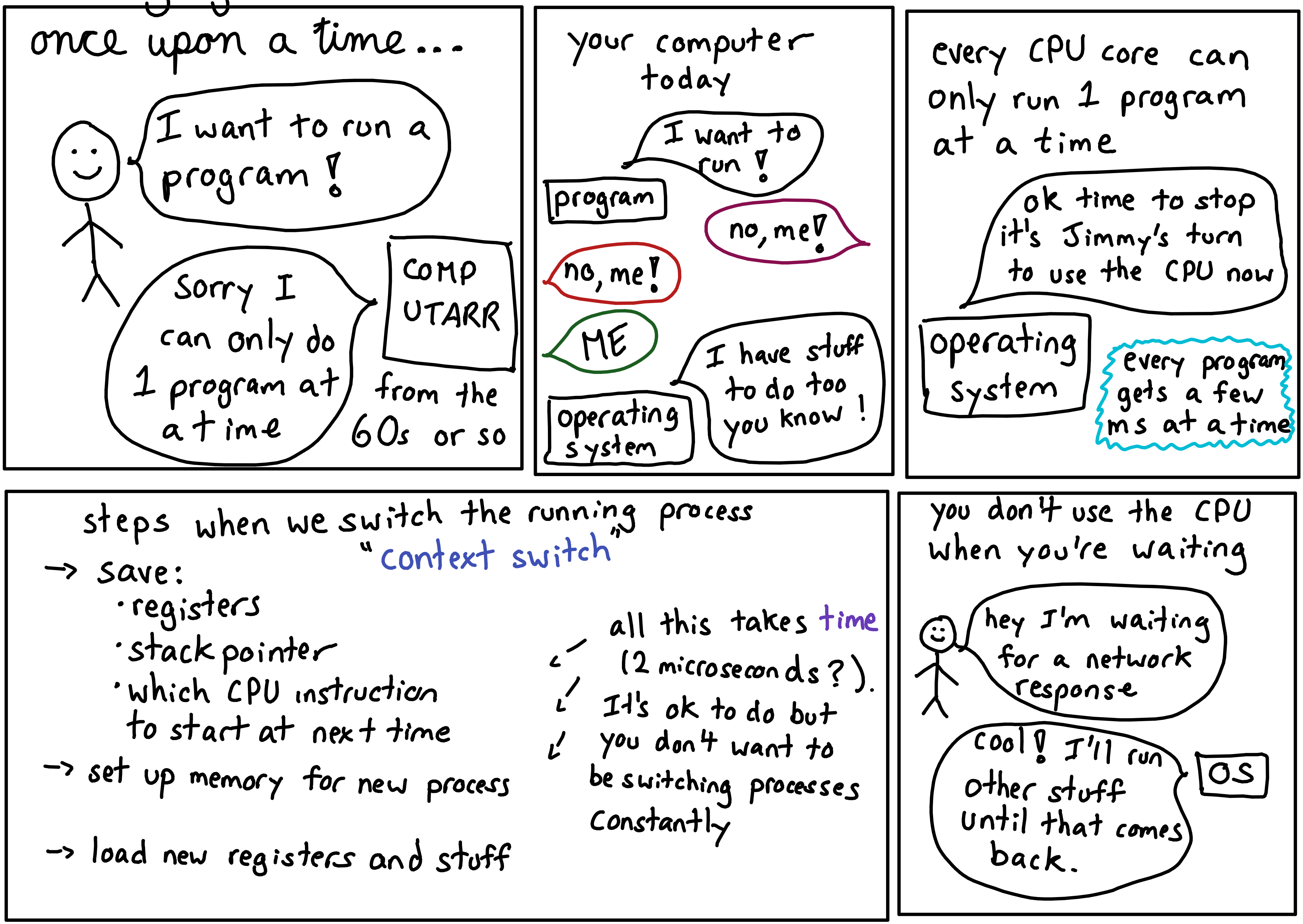
asking good questions



drawings.jvns.ca

CPU scheduling

Julia Evans
@b0rk



more at drawings.jvns.ca

☆ the stack ☆
(in a C program)

Julia Evans
@b0rk

your program has

-> local variables

```
int x = 2;
```

-> a function to return to

```
void parent() {
    do_thing();
}
```

-> function arguments

```
make_cat(name, fluffiness)
```

these all live in a part of memory called **the stack**

example program

```
int fun() {
    int x = 2;
    do_thing(2);
    int y = 4; ☆
}

void do_thing(b) {
    int x = b+1;
    int z = 4; ⇐
    return;
}
```

the stack at ⇐

do_thing:

locals! — 3 (x)

4 (z)

return address! — address for ☆

args ! — 2 (b)

fun:

2 (x)

return addr

(no params)

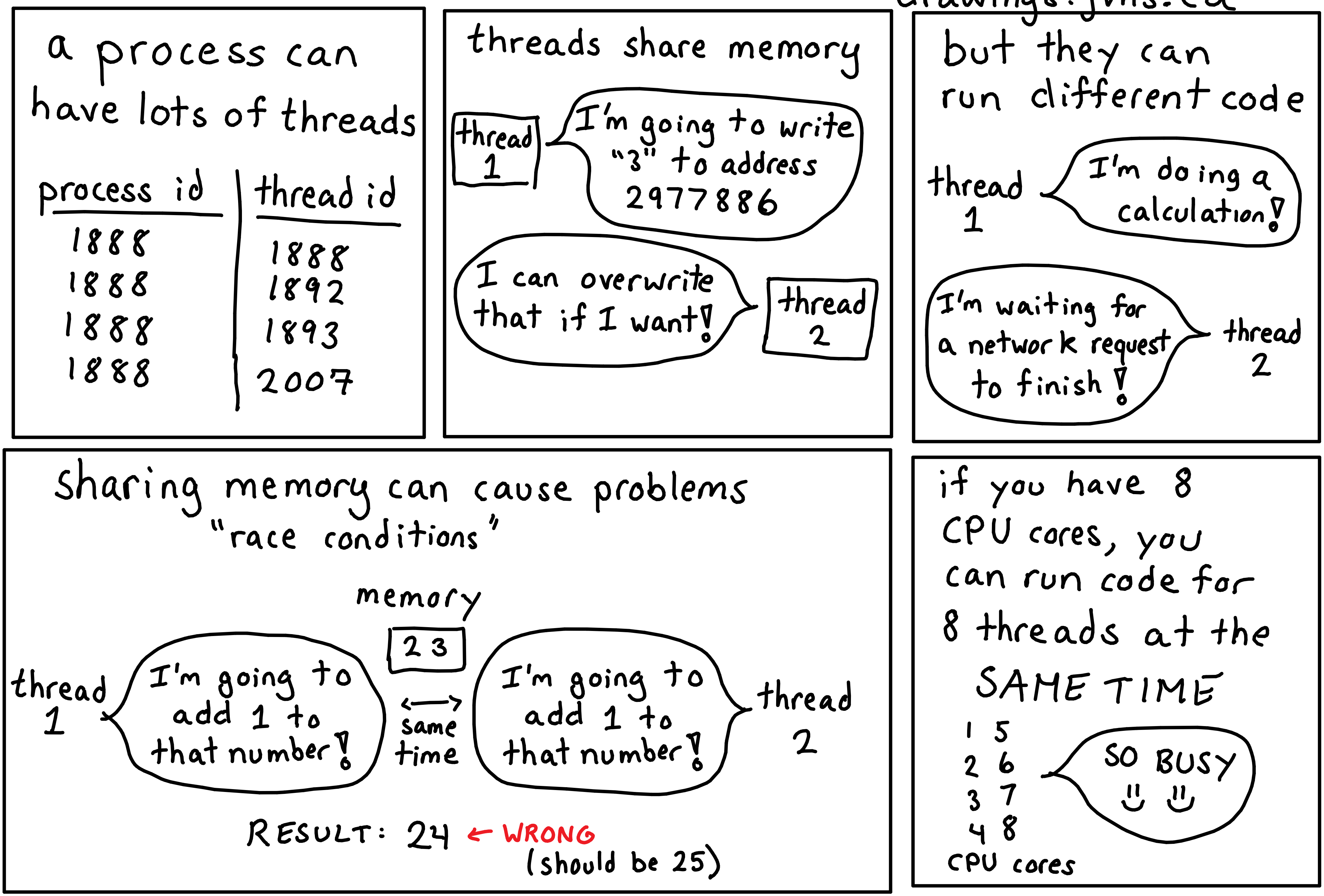
there's a limit to how big your stack can get! Exceed it and you get a

STACK OVERFLOW

What's a thread?

JULIA EVANS
@b0rk

drawings.jvns.ca



user space vs. kernel space

JULIA EVANS
@b0rk

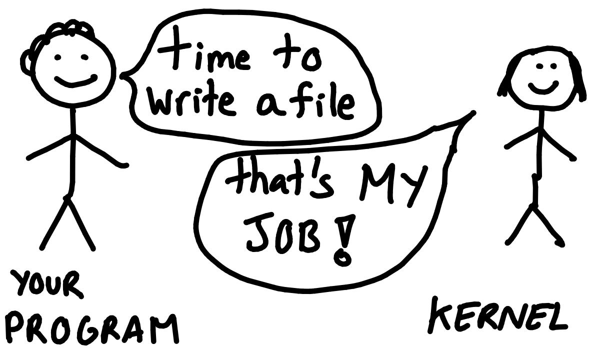
drawings.jvns.ca

the Linux kernel has **millions** of lines of code

- ☆ read + write files
- ☆ decide which programs get to use the CPU
- ☆ make the keyboard work

when Linux kernel code runs, that's called **kernel space**

when *your* program runs, that's **user space**



time for a **context switch** to kernel space

your program switches back and forth

```
str= "my string"
x= x+2
file.write(str)
```

← ☆ switch to kernel space ☆

```
y= x+4
str= str*y
```

← ☆ aand we're back to user space! ☆

timing your process

$ time find /home

0.15 user   0.73 system

↑ time spent in your process

↑ time spent by the kernel doing work for your process

≡ VIRTUAL MEMORY ≡