CCNY
Engineering 101

Sam Fenster
www-cs.ccny.cuny.edu/~fenster/
fenster@ccny.cuny.edu

Sam Fenster

www-cs.ccny.cuny.edu/~fenster/
fenster@ccny.cuny.edu

Associate director,
Computer Engineering program.

I organize these Friday talks.

Today:

- What Engr 101 is about;

- my fields:
  - Computer Science,
  - Computer Engineering.

Engineering 101 will teach what people in the engineering fields do:

- in general, and
- in each specific field.

- My definition of “engineering”:

☞ Using scientific and technical knowledge to solve practical problems.

Engineering majors offered at CCNY:

- Biomedical Engineering (2 in this class)
- Chemical Engineering (0)
- Civil Engineering (0)
- Computer Engineering (11)
- Computer Science (3)
- Earth Systems Science & Environmental Engr. (4)
- Electrical Engineering (22)
- Mechanical Engineering (48)

Not- or not-yet engineers in this class: 30.

Engineering 101 meets twice a week:

- Ongoing team lab project(s);

- Friday guest talks

  by faculty from each department;

  by working engineers.

Engineering 101 will show you some subset of

what people in these fields do, that is,
what you can do with each of these degrees.

(Do you even want engineering? Which area?)

Engineering 101 will show you some subset of

what people in these fields do, that is,
what you can do with each of these degrees.

(Do you even want engineering? Which area?)

- **Lab:** Design & problem-solving, using one or two areas of engineering…
  - …not necessarily in your major.

Engineering 101 will show you some subset of

what people in these fields do, that is,
what you can do with each of these degrees.

(Do you even want engineering? Which area?)

- **Friday talks:** Professionals will explain:

- What the everyday work is like in different jobs & organizations;

- The large tasks & problems that each field addresses;

- The areas of study & methods used:
  science, math, management & organization,
  and how they fit together.

For example:

- Today: **Computing.**

  — What is computer engineering? Computer science?

- Feb. 7th: **The Career Center.**

  — How do you find summer work?

- Feb. 14th: **Ethics in engineering.**

  — How do engineers avoid hurting people?

- Feb. 21st: **Alison Conway, transportation engineer.**

  — What problems does civil engineering solve?
  — How it uses mechanical, computing, environment, …

- *later*: **Zahn Innovation Center.**

  — How to develop your own product & company.

Other classes only cover a single group of methods,
not broad practice,

...and almost nothing about what to do with this
knowledge after graduation.

Senior Design 1 & 2 will put some of it together.

Seek out such non-classroom knowledge,
in school and outside,
during the next 3–4 years.

- **Engineering 101 class details**:

Lab projects and Friday talks are run entirely separately.

- Friday talks:
  - List, video, material: www-cs.ccny.cuny.edu/~fenster/engr101
  - Zoom, logged in as your CCNY account. You have the link. Not using Brightspace.
  - Your attendance will be your final grade, combined with your lab grade(s).
  - Questions are encouraged at the end of each talk.

Computer Science and **Computer Engineering:**

what people in these fields do, that is,
what you can do with each of these degrees.

- What the everyday work is like in different jobs & organizations;
- The large tasks & problems that each field addresses;
- The areas of study & methods used: science, math, management & organization, and how they fit together.

Computer Science and **Computer Engineering [Cp.E.]:**
what people in these fields do, that is,
what you can do with each of these degrees.

- What the everyday work is like in different jobs & organizations;
- The large tasks & problems that each field addresses;

☞ The areas of study & methods used:
science, math, management & organization; how they fit together.

- **Software Engineering**

- **Theory & Algorithms**

- **Scientific programming**

- **Software Engineering**

- **Theory & Algorithms**

- **Scientific programming**

- **Software Engineering**

- **Theory & Algorithms**

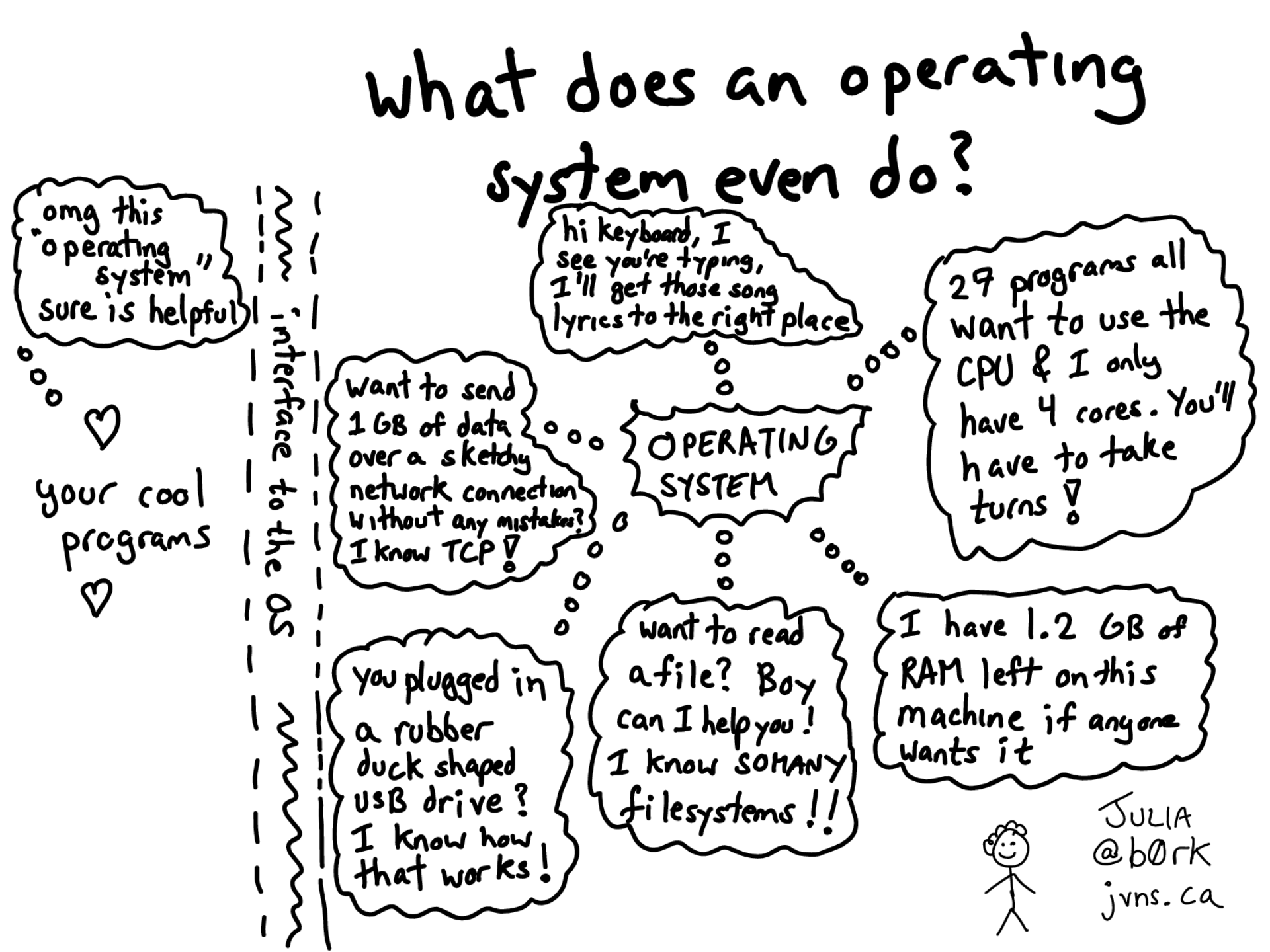
- **Scientific programming**

Designing, writing, testing and maintaining software:

- a Web browser, MS Word, an operating system (Unix/Linux, MS Windows, Mac OS) **[Cp.E.]**
  a library, a C++ compiler **[Cp.E.]**

- embedded software **[Cp.E.]**: iPhone, car, network devices, TV, cameras & video.

What does an operating system even do?



JULIA
@b0rk
jvns.ca

- **Software Engineering**

- **Theory & Algorithms**

- **Scientific programming**

Designing, writing, testing and maintaining software: (continued)

- Making programs correct & bug-free:
  - proofs of code logic correctness,
  - testing code,
  - programming languages & semantics,
  - reusable software libraries,
  - project management.

- Software Engineering

- **Theory & Algorithms**

- Scientific programming

- Software Engineering

- **Theory & Algorithms**

- Scientific programming

- *Cryptography* — keeping a secret.
  Encryption that can't be cracked. Cracking it.
  Authenticating (proving it's you):
  trusting information, gaining access.

- *Storing & retrieving massive data* — databases, text search, maps, graphic objects.

- Software Engineering

- **Theory & Algorithms**

- Scientific programming

Example: Binary search

How would you look up a name in the Manhattan phone directory?

- The dumb way: check every name
  1,500,000 operations $\times$ 100 μsec = 2.5 minutes

- The smart way: start in the middle; repeat
  $\log_2$1,500,000 operations $\times$ 100 μsec
  = 21 operations $\times$ 100 μsec = .002 seconds!

- Software Engineering

- **Theory & Algorithms**

- Scientific programming

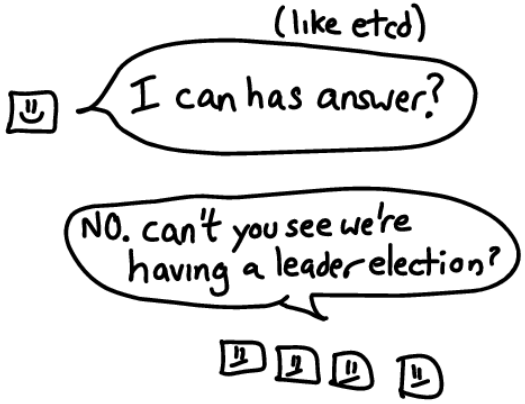
Distributed & parallel processing:

- Multiprocess **operating systems** [Cp.E.];
- Simultaneous access to your bank, Amazon, CCNY registration;
- Parallel processing [Cp.E.]: several CPUs access the same memory chips & keep the data correct;
- Multiple local machines read/write disk data: web farms, business, games.

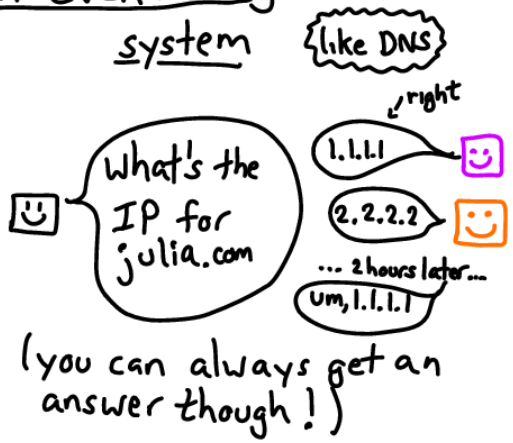
scenes from distributed systems

https://jvns.ca/blog/2016/11/10/a-few-drawings-about-linux/

a "linearizable" system



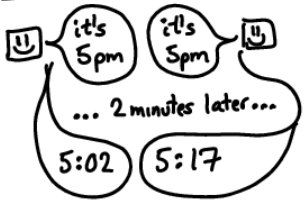
an eventually consistent system



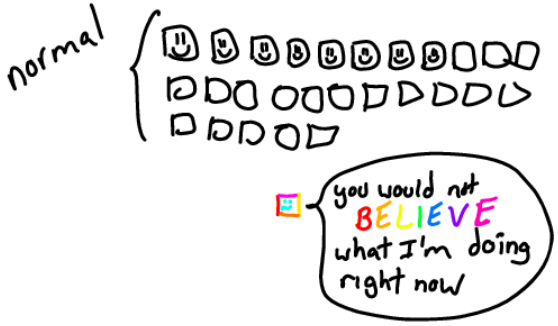
replication is hard

the network is fine BUT

clocks lie



with 1000s of machines... it gets weird



- Software Engineering

- Theory & Algorithms

- **Scientific programming**

- Software Engineering

- Theory & Algorithms

- **Scientific programming**

Math:

- Solve equations.
- Solve integrals.
- Approximate functions.
- Know accuracy.
- For graphics & on-screen motion,
  driving physical objects & devices,
  finding patterns, design & manufacturing.

- Software Engineering

- Theory & Algorithms

- **Scientific programming**

Image & signal processing **[Cp.E.]***:*

- It's all just a sequence/array of numbers.
- Transmit, store, compress & get information from (noisy) images, video & sound.
- Render images & sound from conceptual representation. Graphics, voice synthesis.
- Process signals: zoom, blur, sharpen, EQ, echo.

- Software Engineering

- Theory & Algorithms

- **Scientific programming**

Image & signal processing:

- Optical Character Recognition,
- Face recognition,
- Voice recognition,
- Medical (volume, aim),
- Tracking (driving).

- Software Engineering

- Theory & Algorithms

- **Scientific programming**

~~Image & signal~~ ***Data*** *processing:*

- Machine learning

- Big data

- **Computer Engineering:**

  - Hardware design (E.E.) —
  from gates & flip-flops to CPUs, FPUs,
  graphics drivers & cards, data buses, networks.

  - Program code to run hardware (C.Sc.) —
  networks, machine code & microcode & registers,
  image & sound processing (in & out).

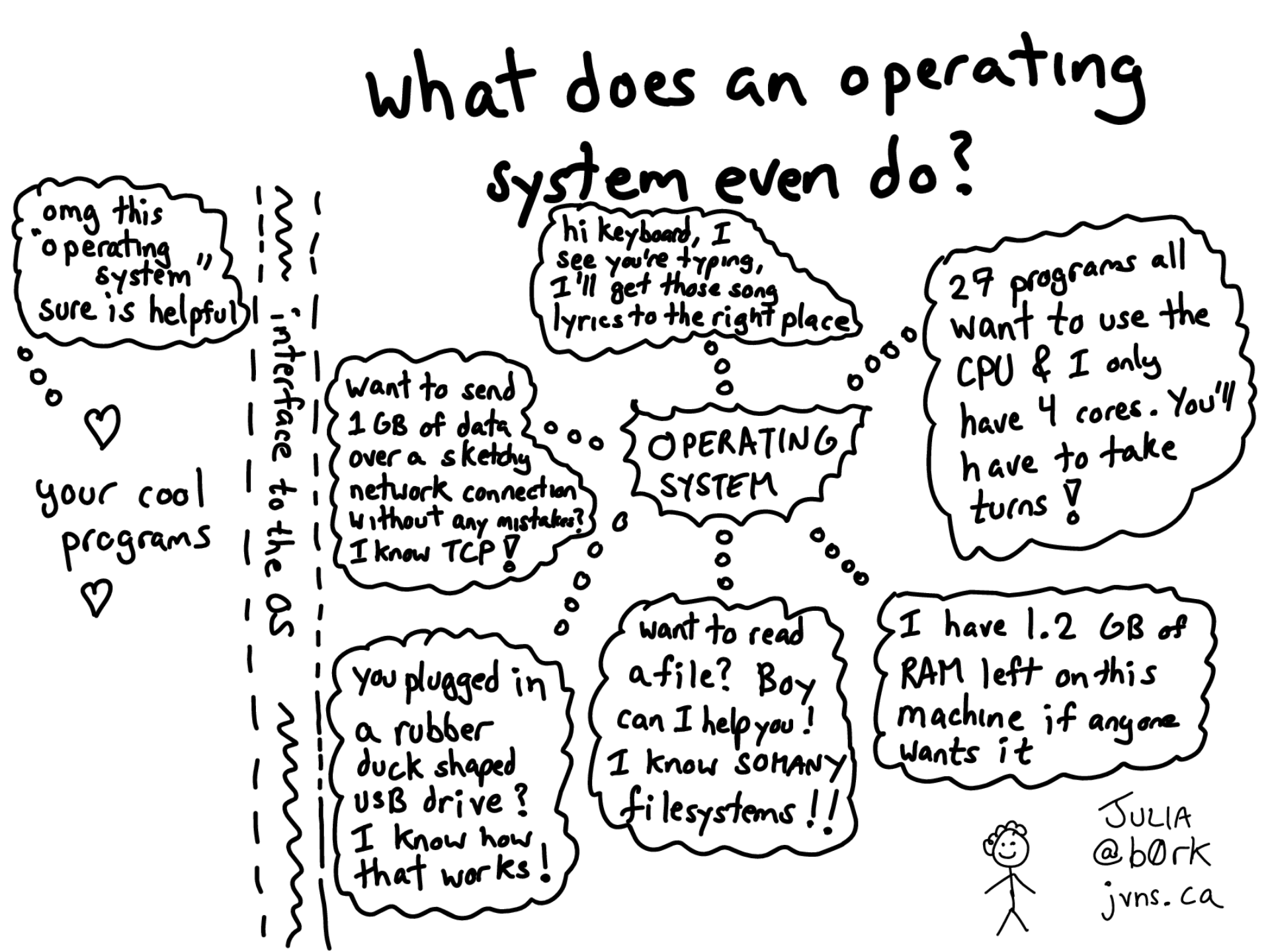
  - Compilers — programming language converted to machine code, CPU & registers ("architecture").

```
#include <iostream>
using namespace std;

int main ()
{
  cout << "Hello, world!\n";
  return 0;
}
```

```
                .LC0:
0000 48656C6C           .string   "Hello, world!\n"
     6F2C2077
     6F726C64
     210A00
                        .text
                        .globl    main
                main:
                .LFB971:
                        .cfi_startproc
0000 55                 pushq     %rbp
                        .cfi_def_cfa_offset 16
                        .cfi_offset 6, -16
0001 4889E5             movq %rsp, %rbp
                        .cfi_def_cfa_register 6
0004 BE000000           movl $.LC0, %esi
     00
0009 BF000000           movl $_ZSt4cout, %edi
     00
000e E8000000           call _ZStlsISt11char_traitsIcEERSt13basic_ostreamIcT_ES5_PKc
     00
0013 B8000000           movl $0, %eax
     00
0018 5D                 popq %rbp
                        .cfi_def_cfa 7, 8
0019 C3                 ret
                        .cfi_endproc
```

What does an operating system even do?



JULIA
@b0rk
jvns.ca

Institutions / organizations / workplaces:

- Software companies sell code to others;
  Device companies sell hardware (& code).

- Code & hardware used internally:
  - web departments & companies,
  - banks & finance,
  - manufacturing,
  - government (MTA, air traffic, IRS, purchasing),
  - medical & hospital,
  - telecommunications,
  - education & research.

I am:

Sam Fenster
www-cs.ccny.cuny.edu/~fenster/
fenster@ccny.cuny.edu

↑

Friday lecture information (including these slides) is at:

www-cs.ccny.cuny.edu/~fenster/engr101

☞ Check every week for changes or cancellations!