

---

---

# Nieskończona liczba małp

Mariusz Wołoszyn

---

---

# Infinite monkey theorem

One of the earliest postulation by french mathematician Émile Borel in 1913.

Origins traced to Aristotle's On Generation and Corruption and Cicero's De natura deorum.

*“Given infinite time (...) almost surely will produce any text.”*

---

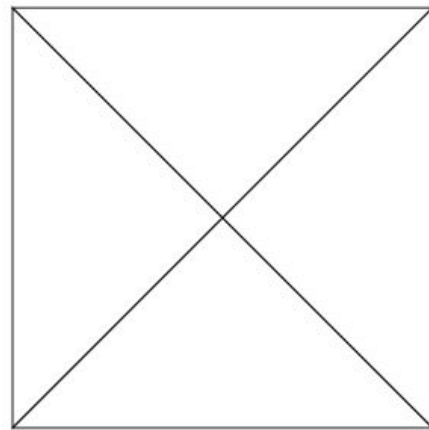


---

# Almost surely?

“In probability theory, one says that an event happens almost surely (sometimes abbreviated as a.s.) if it happens with probability one.”\*

1. Diagonals of the square are zero.
2. Probability hitting exactly on a diagonal is zero
3. One **almost never** hit the diagonal (i.e. it will **almost surely** not hit the diagonal).
4. Nonetheless the set of points on the diagonals is not empty and a point on a diagonal is no less possible than any other point: the diagonal does contain valid outcomes of the experiment.\*\*)



---

\*) [https://en.wikipedia.org/wiki/Almost\\_surely](https://en.wikipedia.org/wiki/Almost_surely)

\*\*) ibidem

---

# Experiments

In 2003, lecturers and students from the [University of Plymouth](#) MediaLab Arts course used a £2,000 grant from the [Arts Council](#) to study the literary output of real monkeys. They left a computer keyboard in the enclosure of six [Celebes crested macaques](#) in [Paignton Zoo](#) in [Devon](#) in [England](#) for a month, with a radio link to broadcast the results on a website.<sup>[11]</sup>

Not only did the monkeys produce nothing but five total pages largely consisting of the letter S,<sup>[12]</sup> the lead male began bashing the keyboard with a stone, and the monkeys followed by soiling it. Mike Phillips, director of the university's Institute of Digital Arts and Technology (i-DAT), said that the artist-funded project was primarily performance art, and they had learned "an awful lot" from it. He concluded that **monkeys "are not random generators"**. They're more complex than that. ... They were quite interested in the screen, and they saw that when they typed a letter, something happened. There was a level of intention there."<sup>[11][13]</sup>

The full text created by the monkeys is available to read here.<sup>[14]</sup>

[https://en.wikipedia.org/wiki/Infinite\\_monkey\\_theorem#Real\\_monkeys](https://en.wikipedia.org/wiki/Infinite_monkey_theorem#Real_monkeys)

---



---

# Wirtualne małpy

Według artykułu w [The New Yorker](#) program komputerowy uruchomiony przez Dana Olivera ze Scottsdale w Arizonie, zwrócił 4 sierpnia 2004 roku następujący wynik: Po 42 162 500 tryliardach ( $4,2 \times 10^{28}$ ) lat jedna z grupy „małp” stworzyła tekst:

**VALENTINE. Cease toldor:**eFLP0FRjWK78aXzVOwm)-';8.t...

Pierwszych 19 liter tej sekwencji znajduje się w komedii [Dwaj panowie z Werony](#). Inne grupy odtworzyły 18 liter z tragedii [Tymon Ateńczyk](#), 17 z tragedii [Troilus i Kresyda](#) oraz 16 z [Ryszarda III](#)<sup>[27]</sup>.

[https://pl.wikipedia.org/wiki/Twierdzenie\\_o\\_niesko%C5%84czonej\\_liczbie\\_ma%C5%82p](https://pl.wikipedia.org/wiki/Twierdzenie_o_niesko%C5%84czonej_liczbie_ma%C5%82p)

---





# Więcej małp

1 lipca 2003 roku uruchomiono stronę internetową *The Monkey Shakespeare Simulator*, zawierającą [aplet Javy](#), który symulował wielką populację losowo piszących małp. Intencją twórców było sprawdzenie jak długo zajmie wirtualnym małpom napisanie kompletnej sztuki Szekspira. Przykładowo aplet wyprodukował poniższy, zgodny w 24 znakach, fragment pochodzący ze sztuki [Henryk IV, część 2](#), wraz z oszacowaniem, że proces zajął „2 737 850 milionów miliardów miliardów miliardów małpo-lat” (tj. 2,7 sekstyliarda =  $2,7 \times 10^{39}$ ):

**RUMOUR. Open your ears; 9r"5j5&?OWTY Z0d...**

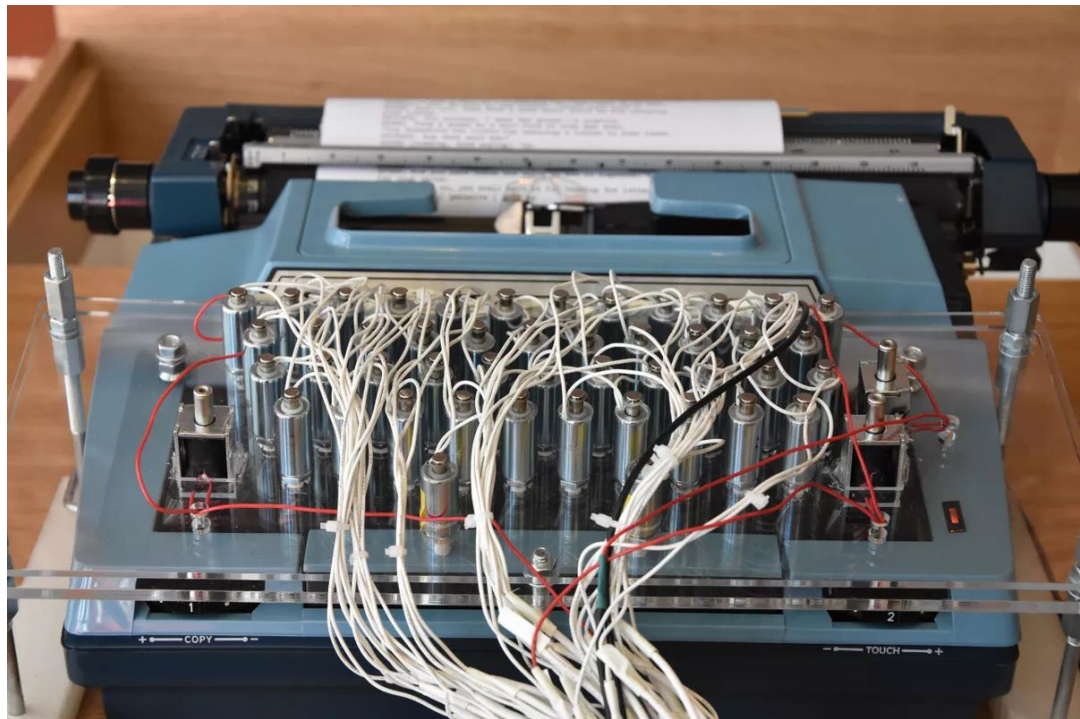
Ze względu na ograniczenia mocy obliczeniowej, program używał modelu probabilistycznego (wykorzystując generator liczb losowych) zamiast naprawdę losować tekst i porównywać ze sztukami. Gdy symulator wykrywał zgodność znaków (to jest, gdy wynik z generatora losowego zawierał się w danym przedziale), symulator naśladował wpisanie zgodnych znaków<sup>[28]</sup>. Obecnie strona już nie istnieje.

[https://pl.wikipedia.org/wiki/Twierdzenie\\_o\\_niesko%C5%84czonej\\_liczbie\\_ma%C5%82p](https://pl.wikipedia.org/wiki/Twierdzenie_o_niesko%C5%84czonej_liczbie_ma%C5%82p)

---

# AI Monkeys

What if we use AI monkeys?  
Can we generate source code?



# Generative models



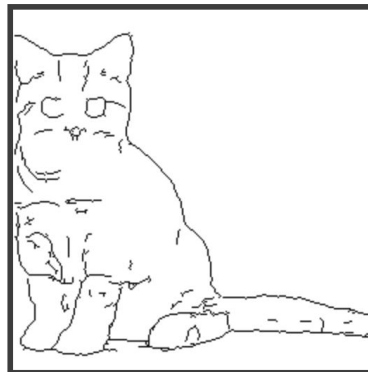
Neural nets can generate content:

[Faces](#), captions, cats, zebras, deep fakes, speech, etc...



It can write text too.

INPUT

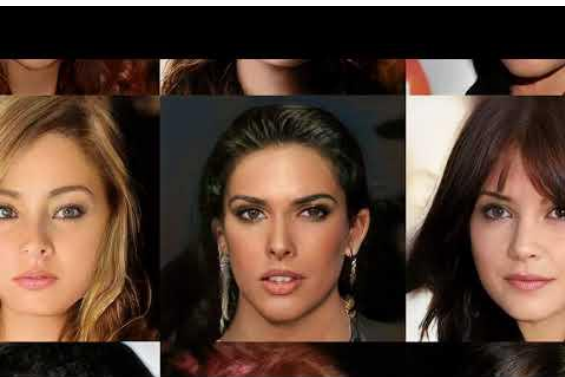


OUTPUT



pix2pix

process

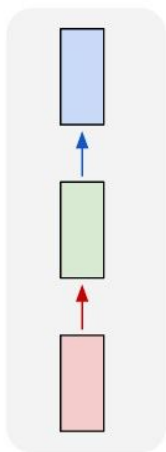




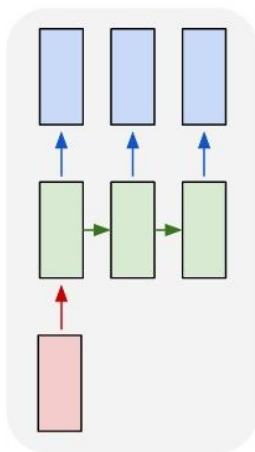
# NN Types



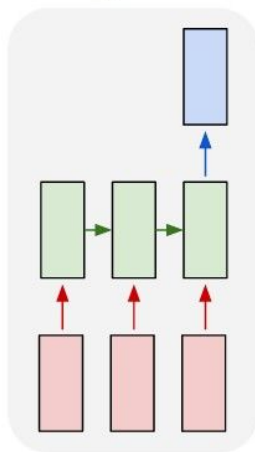
one to one



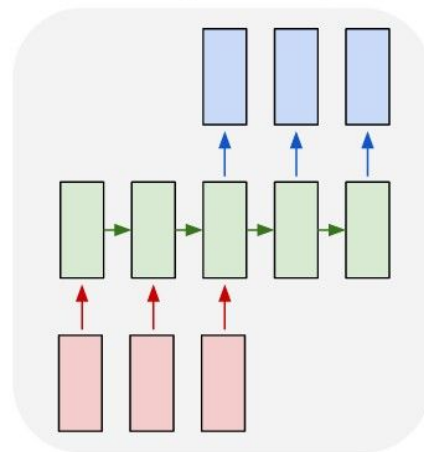
one to many



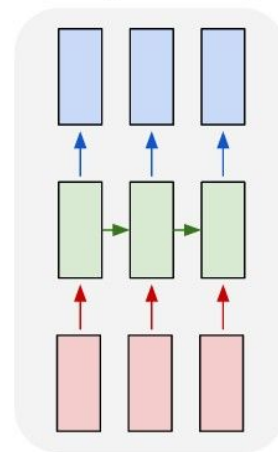
many to one



many to many

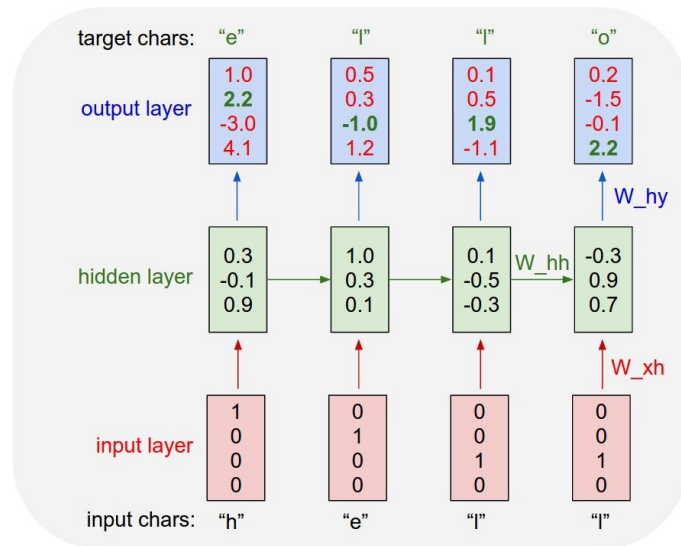
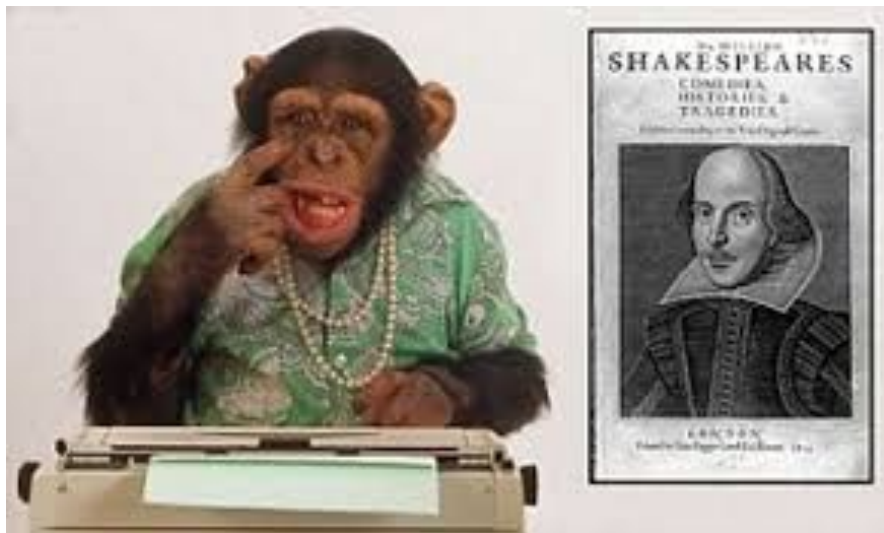


many to many



# RNN

## Recurrent networks



# AI Shakespeare

PANDARUS:

Alas, I think he shall be come approached and the day  
When little strain would be attain'd into being never fed,  
And who is but a chain and subjects of his death,  
I should not sleep.

Second Senator:

They are away this miseries, produced upon my soul,  
Breaking and strongly should be buried, when I perish  
The earth and thoughts of many states.

DUKE VINCENTIO:

Well, your wit is in the care of side and that.

Second Lord:

They would be ruled after this chamber, and  
my fair nudes begun out of the fact, to be conveyed,  
Whose noble souls I'll have the heart of the wars.

Clown:

Come, sir, I will make did behold your worship.

VIOLA:

I'll drink it.

VIOLA:

Why, Salisbury must find his flesh and thought  
That which I am not apt, not a man and in fire,  
To show the reining of the raven and the wars  
To grace my hand reproach within, and not a fair are hand,  
That Caesar and my goodly father's world;  
When I was heaven of presence and our fleets,  
We spare with hours, but cut thy council I am great,  
Murdered and by thy master's ready there  
My power to give thee but so much as hell:  
Some service in the noble bondman here,  
Would show him to her wine.

KING LEAR:

O, if you were a feeble sight, the courtesy of your law,  
Your sight and several breath, will wear the gods  
With his heads, and my hands are wonder'd at the deeds,  
So drop upon your lordship's head, and your opinion  
Shall be against your honour.





# Wikipedia

Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. His generals were the powerful ruler of the Portugal in the [[Protestant Immineners]], which could be said to be directly in Cantonese Communication, which followed a ceremony and set inspired prison, training. The emperor travelled back to [[Antioch, Perth, October 25|21]] to note, the Kingdom of Costa Rica, unsuccessful fashioned the [[Thrales]], [[Cynth's Dajoard]], known in western [[Scotland]], near Italy to the conquest of India with the conflict. Copyright was the succession of independence in the slop of Syrian influence that was a famous German movement based on a more popular servicious, non-doctrinal and sexual power post. Many governments recognize the military housing of the [[Civil Liberalization and Infantry Resolution 265 National Party in Hungary]], that is sympathetic to be to the [[Punjab Resolution]] (PJS)[<http://www.humah.yahoo.com/guardian.cfm/7754800786d17551963s89.htm> Official economics Adjoint for the Nazism, Montgomery

was swear to advance to the resources for those Socialism's rule, was starting to signing a major tripad of aid exile.]]

# Scientific “papers”

For  $\bigoplus_{n=1,\dots,m}$  where  $\mathcal{L}_{m,\bullet} = 0$ , hence we can find a closed subset  $\mathcal{H}$  in  $\mathcal{H}$  and any sets  $\mathcal{F}$  on  $X$ ,  $U$  is a closed immersion of  $S$ , then  $U \rightarrow T$  is a separated algebraic space.

*Proof.* Proof of (1). It also start we get

$$S = \text{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by  $\coprod Z \times_U U \rightarrow V$ . Consider the maps  $M$  along the set of points  $\text{Sch}_{fppf}$  and  $U \rightarrow U$  is the fibre category of  $S$  in  $U$  in Section, ?? and the fact that any  $U$  affine, see Morphisms, Lemma ???. Hence we obtain a scheme  $S$  and any open subset  $W \subset U$  in  $\text{Sh}(G)$  such that  $\text{Spec}(R') \rightarrow S$  is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that  $f_i$  is of finite presentation over  $S$ . We claim that  $\mathcal{O}_{X,x}$  is a scheme where  $x, x', s'' \in S'$  such that  $\mathcal{O}_{X,x'} \rightarrow \mathcal{O}'_{X',x'}$  is separated. By Algebra, Lemma ?? we can define a map of complexes  $\text{GL}_{S'}(x'/S'')$  and we win.  $\square$

To prove study we see that  $\mathcal{F}|_U$  is a covering of  $\mathcal{X}'$ , and  $\mathcal{T}_i$  is an object of  $\mathcal{F}_{X/S}$  for  $i > 0$  and  $\mathcal{F}_p$  exists and let  $\mathcal{F}_i$  be a presheaf of  $\mathcal{O}_X$ -modules on  $\mathcal{C}$  as a  $\mathcal{F}$ -module. In particular  $\mathcal{F} = U/\mathcal{F}$  we have to show that

$$\tilde{M}^\bullet = T^\bullet \otimes_{\text{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1} \mathcal{F}$$

is a unique morphism of algebraic stacks. Note that

$$\text{Arrows} = (\text{Sch}/S)_{fppf}^{opp}, (\text{Sch}/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longrightarrow (U, \text{Spec}(A))$$

is an open subset of  $X$ . Thus  $U$  is affine. This is a continuous map of  $X$  is the inverse, the groupoid scheme  $S$ .

*Proof.* See discussion of sheaves of sets.  $\square$

The result to prove any open covering follows from the less of Example ???. It may replace  $S$  by  $X_{spaces, \acute{e}tale}$  which gives an open subspace of  $X$  and  $T$  equal to  $S_{Zar}$ , see Descent, Lemma ???. Namely, by Lemma ?? we see that  $R$  is geometrically regular over  $S$ .

*Proof.* Omitted.  $\square$

**Lemma 0.1.** Let  $\mathcal{C}$  be a set of the construction.

Let  $\mathcal{C}$  be a gerber covering. Let  $\mathcal{F}$  be a quasi-coherent sheaves of  $\mathcal{O}$ -modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

*Proof.* This is an algebraic space with the composition of sheaves  $\mathcal{F}$  on  $X_{\acute{e}tale}$  we have

$$\mathcal{O}_X(\mathcal{F}) = \{\text{morph}_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where  $\mathcal{G}$  defines an isomorphism  $\mathcal{F} \rightarrow \mathcal{F}$  of  $\mathcal{O}$ -modules.  $\square$

**Lemma 0.2.** This is an integer  $\mathbb{Z}$  is injective.

*Proof.* See Spaces, Lemma ???.  $\square$

**Lemma 0.3.** Let  $S$  be a scheme. Let  $X$  be a scheme and  $X$  is an affine open covering. Let  $\mathcal{U} \subset \mathcal{X}$  be a canonical and locally of finite type. Let  $X$  be a scheme. Let  $X$  be a scheme which is equal to the formal complex.

The following to the construction of the lemma follows.

Let  $X$  be a scheme. Let  $X$  be a scheme covering. Let

$$b : X \rightarrow Y' \rightarrow Y \rightarrow Y' \times_X Y \rightarrow X.$$

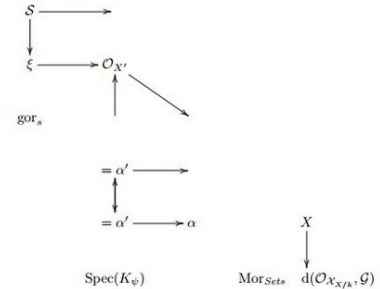
be a morphism of algebraic spaces over  $S$  and  $Y$ .

*Proof.* Let  $X$  be a nonzero scheme of  $X$ . Let  $X$  be an algebraic space. Let  $\mathcal{F}$  be a quasi-coherent sheaf of  $\mathcal{O}_X$ -modules. The following are equivalent

- (1)  $\mathcal{F}$  is an algebraic space over  $S$ .
- (2) If  $X$  is an affine open covering.

Consider a common structure on  $X$  and  $X$  the functor  $\mathcal{O}_X(U)$  which is locally of finite type.  $\square$

This since  $\mathcal{F} \in \mathcal{F}$  and  $x \in \mathcal{G}$  the diagram



is a limit. Then  $\mathcal{G}$  is a finite type and assume  $S$  is a flat and  $\mathcal{F}$  and  $\mathcal{G}$  is a finite type  $\mathcal{F}$ . This is of finite type diagrams, and

- the composition of  $\mathcal{G}$  is a regular sequence,
- $\mathcal{O}_{X'}$  is a sheaf of rings.  $\square$

*Proof.* We have see that  $X = \text{Spec}(R)$  and  $\mathcal{F}$  is a finite type representable by algebraic space. The property  $\mathcal{F}$  is a finite morphism of algebraic stacks. Then the cohomology of  $X$  is an open neighbourhood of  $U$ .  $\square$

*Proof.* This is clear that  $\mathcal{G}$  is a finite presentation, see Lemmas ??.

A reduced above we conclude that  $U$  is an open covering of  $\mathcal{C}$ . The functor  $\mathcal{F}$  is a “field”

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_x \longrightarrow \mathcal{O}_{X',x}^{-1} \mathcal{O}_{X,x}(\mathcal{O}_{X,x}^\vee)$$

is an isomorphism of covering of  $\mathcal{O}_{X'}$ . If  $\mathcal{F}$  is the unique element of  $\mathcal{F}$  such that  $X$  is an isomorphism.

The property  $\mathcal{F}$  is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme  $\mathcal{O}_X$ -algebra with  $\mathcal{F}$  are opens of finite type over  $S$ . If  $\mathcal{F}$  is a scheme theoretic image points.  $\square$

If  $\mathcal{F}$  is a finite direct sum  $\mathcal{O}_{X_X}$  is a closed immersion, see Lemma ???. This is a sequence of  $\mathcal{F}$  is a similar morphism.



# Linux “source”

```

/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
         */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
        goto bail;
    }
    segaddr = in_SB(in.addr);
    selector = seg / 16;
    setup_works = true;
    for (i = 0; i < blocks; i++) {
        seq = buf[i++];
        bpf = bd->bd.next + i * search;
        if (fd) {
            current = blocked;
        }
    }
    rw->name = "Getjbbregs";
    bprm_self_clearl(&iv->version);
    regs->new = blocks[(BPF_STATS << info->historidac) | PFMR_CLOBATHINC_SECONDS << 12;
    return segtable;
}

```

```

/*
 * If this error is set, we will need anything right after that BSD.
 */
static void action_new_function(struct s_stat_info *wb)
{
    unsigned long flags;
    int lel_idx_bit = e->edd, *sys & ~((unsigned long) *FIRST_COMPAT);
    buf[0] = 0xFFFFFFFF & (bit << 4);
    min(inc, slist->bytes);
    printk(KERN_WARNING "Memory allocated %02x/%02x, "
        "original MLL instead\n"),
        min(min(multi_run - s->len, max) * num_data_in),
        frame_pos, sz + first_seg);
    div_u64_w(val, inb_p);
    spin_unlock(&disk->queue_lock);
    mutex_unlock(&s->sock->mutex);
    mutex_unlock(&func->mutex);
    return disassemble(info->pending_bh);
}

static void num_serial_settings(struct tty_struct *tty)
{
    if (tty == tty)
        disable_single_st_p(dev);
    pci_disable_spool(port);
    return 0;
}

```



---

# Pan “Tadeusz”



nie było jej; mignęła tylko śród okienka  
jej różowa wstążeczka i biała sukienka.  
widać na grzędach, jaką pszy minim do poki.  
a niech jeszcze nigdy uchu. potku tylko,  
kle kweszło nagot wszyscy rzierz seról niczą!  
kropliki tłuszczy raz przerzega\* u wzroju,  
albo jak prosze zabłą. lecz przez wyjana!  
tak szlachta to kochano za tysku zalata  
że na kichek, jakby oblicat i pan stać,  
i książę staworszy zawogi, kulczniko hrobo.  
odeszczegi się naż tak też górą zmiada słańca.

wszystko pan! też tak nie z dobrzy jeszcze długim litci.  
podpajła wrzek hrebką gławę, jak sobie z ramienie.

catej szerskali! znarż hesłuskie się kowica,  
wstądzie z halbeem skryja harłani mówką podaleki,  
motołu wez sza cala już skoczył w prosy,  
i drogą, i jak we grzędy, i najnie wyszejenia,  
on bornie starnych zarszy hajba siedział przed rajny,

---

---

# Generating FreeBSD



Getting source code and concatenating to single corpus.

```
svn checkout http://svn.freebsd.org/base/head/sys/
```

Warning this does NOT work properly on FreeBSD :)

```
find . -name '*.c' -exec cat {} >> p.c \;
```

---



---

# Code Char stats

22620908 spaces

7746120 'e'

7097026 '0'

6550779 't'

5815508 ', '

5496898 ' \_'

5393945 '\\n'

5213376 '\\t'

5045093 'x'

followed by other

characters...



'j', 82093

'''', 65663

'z', 65579

'@', 27440

'J', 21658

'?', 19961

'~', 16699

'\$', 12202

'`', 9678

'^', 5244

---

---

# Tensorflow

```
rv++)
    if (strcmp((*p_drv)->driver_name, name) == 0)
        break;
    e ne _e e _
s c e __eet ei__es_t c_t _ s_e e _ e _ tt
t i s _ e _ ie_t _e nss s t _
e t te e
_t_t e a s _ s__st e es_ _ts_e
```

```
rv++)
    if (strcmp((*p_drv)->driver_name, name) == 0)
        break;
    d
i nK
RAp,e=n "mNtowi*eipTf
;P{c&
M t sr6gh n )u_d>s
mstc_tc=c * tLRaia_eUesBUepmbgpea>o Lccurpsehyii{ (b_ ea_ gs={oic})(_gmRfc:tlul!b ';p mhlgEe(dem bhb n neC*log
aen B& oa a nt g
dfUn
```



```
voiddevi_fseg_sed */ * |GREGBUTHECHORKER_CHET */ * |= tistet_sind.corrent */
*/ * (sitinderad_sind txa,

{"ute therod tintamma_tio_if ftiond tdivapedind_tat thad_tatatatethethe thalak_se * ahe tx00_d tist teret_tet_tethe thale ale thamsetitititititit
```

# pycharm

```
if (error)
    * tx terromaker *tethe tx",
                                                                    */
*tesethale tx00_sind.c001Bad_tarod ahtist * (st * * |==,
    stad_t tatad foider tx0, 0, 0,
    *
        */
        * (st tetx019"
    /** (ameditarainte */
    */

*/ * | tdinditet_t tet_sititint */ * (se ag_staterutad_
```



\_\_\_\_\_

```

ctl_set_busy(&io->scsiio);
io->scsiio.be_max_unlock(&softc->done_len, status);
if (scsi_cmd->open_count != 0) {
    return = 1;
    if (sense_len > bus != NULL) {
        sendordention->protocol = NULL;
        if (softc->delete_device_stats == N
        softc->desc_periph_lock(&softc->par
}

```

```

if (be_lun->vn == NULL)
    cbe_lun->flags |= CTL_LUN_FLAG_NO_MEDIA;
/* Tell the user the blocksize we ended up using in state, we can for the sort the lock.
 * The requests.
 */
softc->flags |= SA_FLAG_CAN_PROBE;
    param_param_len = sizeof(softc->lock);
    status = CTL_FLAG_DATA_ORT(&cbb->control_byte);
    if ((error == 0) {
        if (error != 0) {
            if (cdb->pos.control) {
                softc->flags |= CAM_DIR_IN:
            if (softc->flags & CTL_FLAG_DATA_IN | STATQ_DIR_IN &case 0) {
                if (cdb->csio.cdb stats.ccb state

```

---

# AI for coding

- Intelligent autocompletion (tabnine: <https://tabnine.com/blog/deep/>)
- Automated bug fixing:
  - FB AI:  
<https://engineering.fb.com/developer-tools/finding-and-fixing-software-bugs-automatically-with-sapfix-and-sapienz/>
  - Luc Esape <https://github.com/lucesape>; his bug fixes went through code review... the caveat? He's an AI :)  
(<https://medium.com/@martin.monperrus/human-competitive-patches-in-automatic-program-repair-with-repairator-359042e00f6a>, [https://www.theregister.co.uk/2018/10/17/luc\\_esape\\_bug\\_fixer/](https://www.theregister.co.uk/2018/10/17/luc_esape_bug_fixer/))
  - [Deep coding: when the machine learns to code by itself](#)

