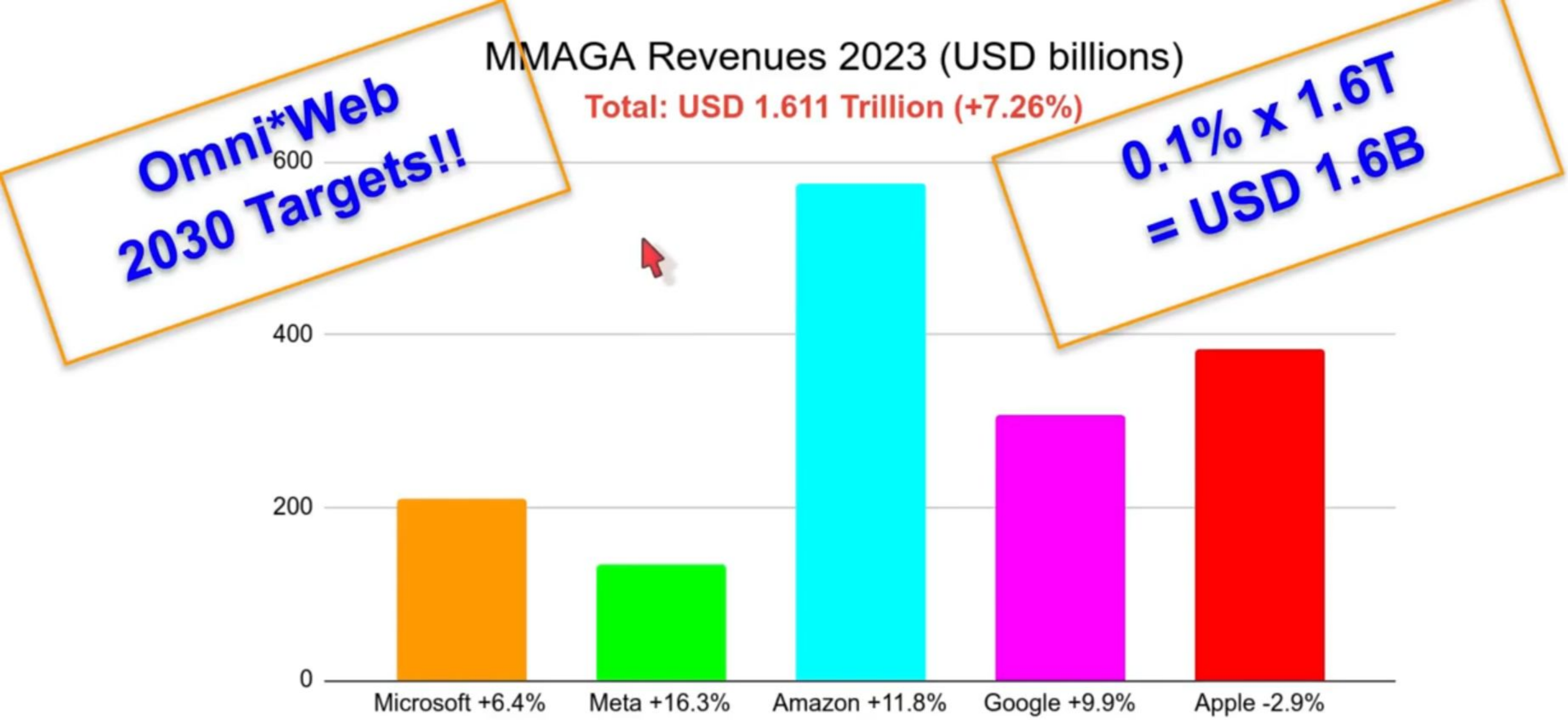
Part D: “Trojan Horse Super-Shell?”

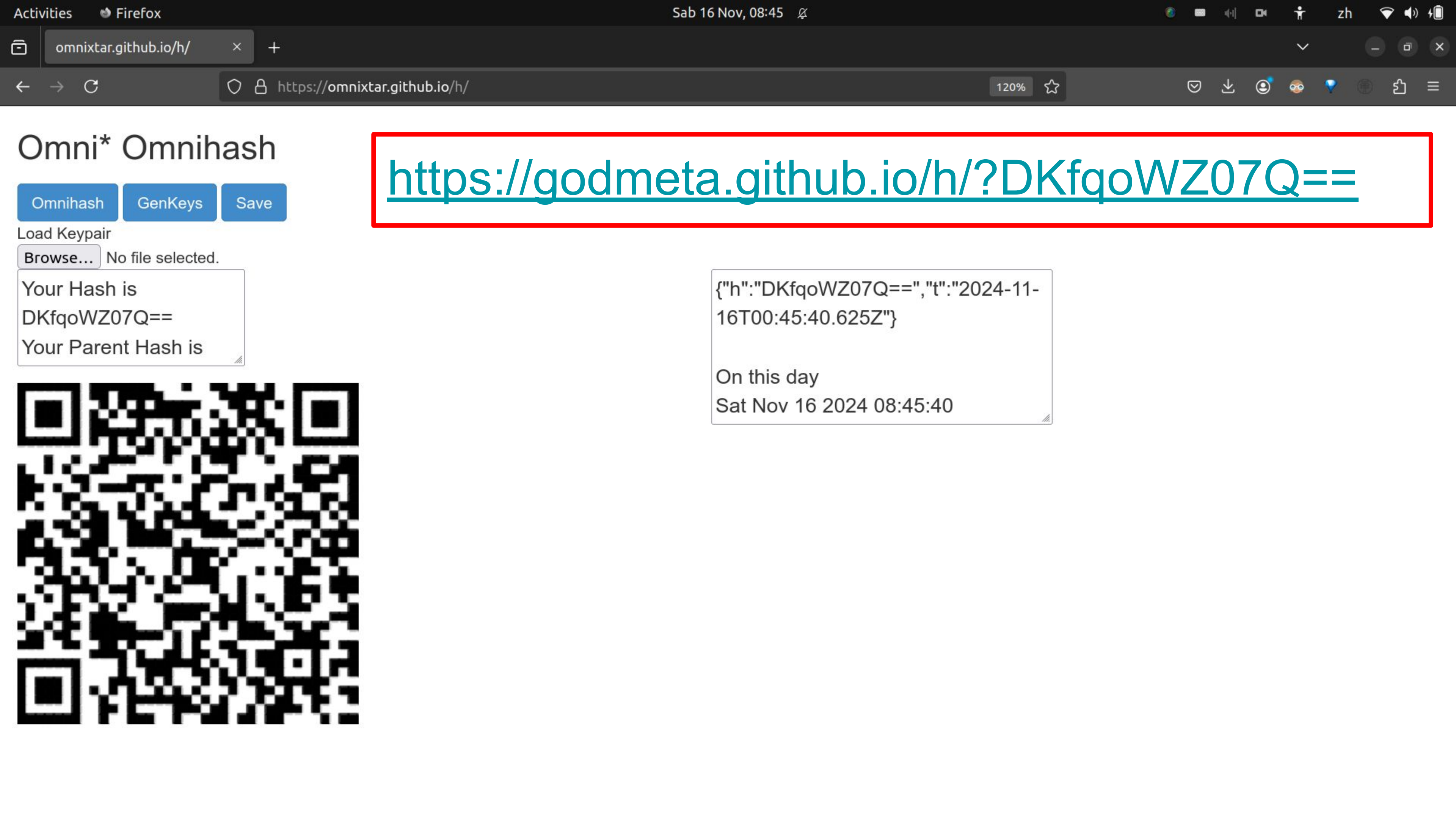
– What tools have we invented?
– Where are we today?
– Where are we heading?
– How do we get there?



A literal trillion dollar question: Do you know that the combined revenues of the top 5 technology companies in 2023 already exceeded USD 1.6 trillion and are growing strongly above 5% annually?

Part D: “Trojan Horse Super-Shell?”

– How do we get there?
– Consolidating Users across Open Source Projects & Commercial Social Media by *adding Omni*Shell* to generate Omnihash User ID as *universal decentralised ID*.
– Add *hash address* in comment section in social media apps to redirect.



Omni* Omnihash

https://godmeta.github.io/h/?DKfqoWZ07Q==

Load Keypair

No file selected.

Your Hash is
DKfqoWZ07Q==
Your Parent Hash is

{"h":"DKfqoWZ07Q==","t":"2024-11-16T00:45:40.625Z"}

On this day
Sat Nov 16 2024 08:45:40

From Technofeudalism to Technosocialism: FORTH unexpected roles in Decentralised Systems

Technofeudalism is a term coined by former Greek finance minister Yanis Varoufakis as the title of a book which compares top American technology companies to feudal lords, who literally enslaved individual users and free software programmers, exploiting their data and source code without fairly rewarding them.

Technosocialism has been used by many other authors but I use it to describe the very real but underreported progress made by free software programmers and free software, despite the lack of fair mechanisms for rewards so far. Free software programmers or technosocialists have made technofeudalism possible as a matter of fact, but they do not get to enjoy the rewards.

Bitcoin and related projects were supposed to address this problem, and are successful in their own way, so we are now half way towards technosocialism.

At section interval, recap past sections, preview this section

Trojan horse: small size of Omni*Shell, 3kb minified gzipped JavaScript.

Football Analogy: I2P (Invisible Internet Project) and Bitcoin have proven the *<u>technical viability of fundamental Decentralised algorithms</u>*, produced many millionaires and billionaires – carrying football past half field. (25 years, *<u>beneficiaries are like Bankers</u>*)

Omni*Web plan to *<u>make more people rich – like McDonald's</u>* vs. Hongkong & Shanghai Bank – second half of football field – next 25 years.

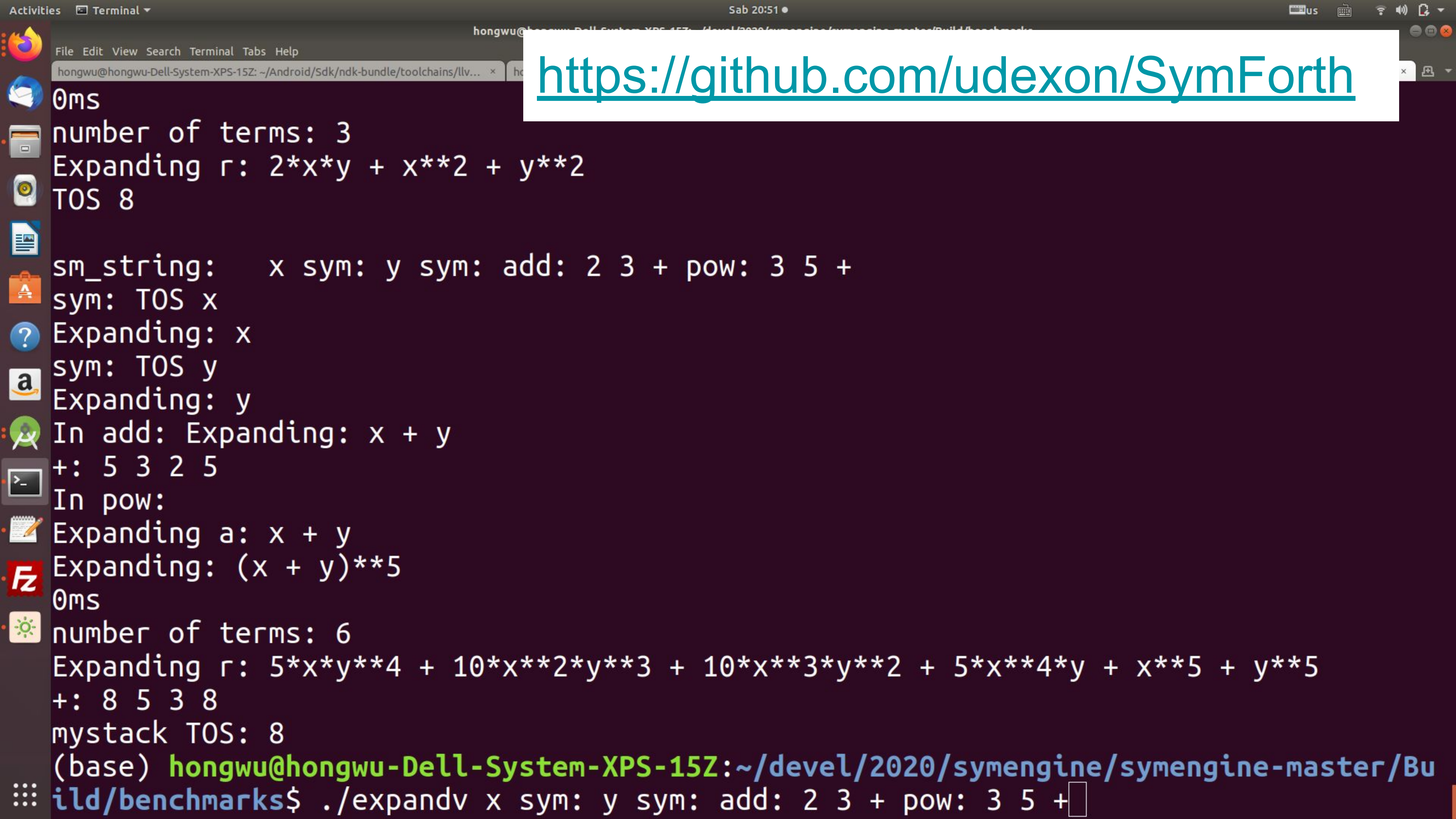
Existing web ecosystem – *dominated by a few huge social media applications* – difficult for small applications to grow – many reasons – *User Fragmentation.*

Solution: Omnihas User ID

Strategy: *"Trojan Horse Super-Shell"*, not the computer virus type – *security guaranteed by hash & public key cryptography*.

Omni*Shell in *JavaScript & Java* – cover most client applications in web & Android – *1000 users per open source project or commercial social media* – *100 projects → 100,000 Omni*Web users.*

Omni*Shell in *JavaScript & Java* – cover most client applications in web & Android – *easily ported to other programming languages* – Apple – *server side – PHP, Python, C++* – (show screenshots) https://github.com/udexon/SymForth



```
0ms
number of terms: 3
Expanding r: 2*x*y + x**2 + y**2
TOS 8


sm_string:   x sym: y sym: add: 2 3 + pow: 3 5 +
sym: TOS x
Expanding: x
sym: TOS y
Expanding: y
In add: Expanding: x + y
+: 5 3 2 5
In pow:
Expanding a: x + y
Expanding: (x + y)**5
0ms
number of terms: 6
Expanding r: 5*x*y**4 + 10*x**2*y**3 + 10*x**3*y**2 + 5*x**4*y + x**5 + y**5
+: 8 5 3 8
mystack TOS: 8
(base) hongwu@hongwu-Dell-System-XPS-15Z:~/devel/2020/symengine/symengine-master/Bu
ild/benchmarks$ ./expandv x sym: y sym: add: 2 3 + pow: 3 5 +
```

Activities Google Chrome Jum 15 Nov, 17:54 zh

SymForth/benchmarks/e ×

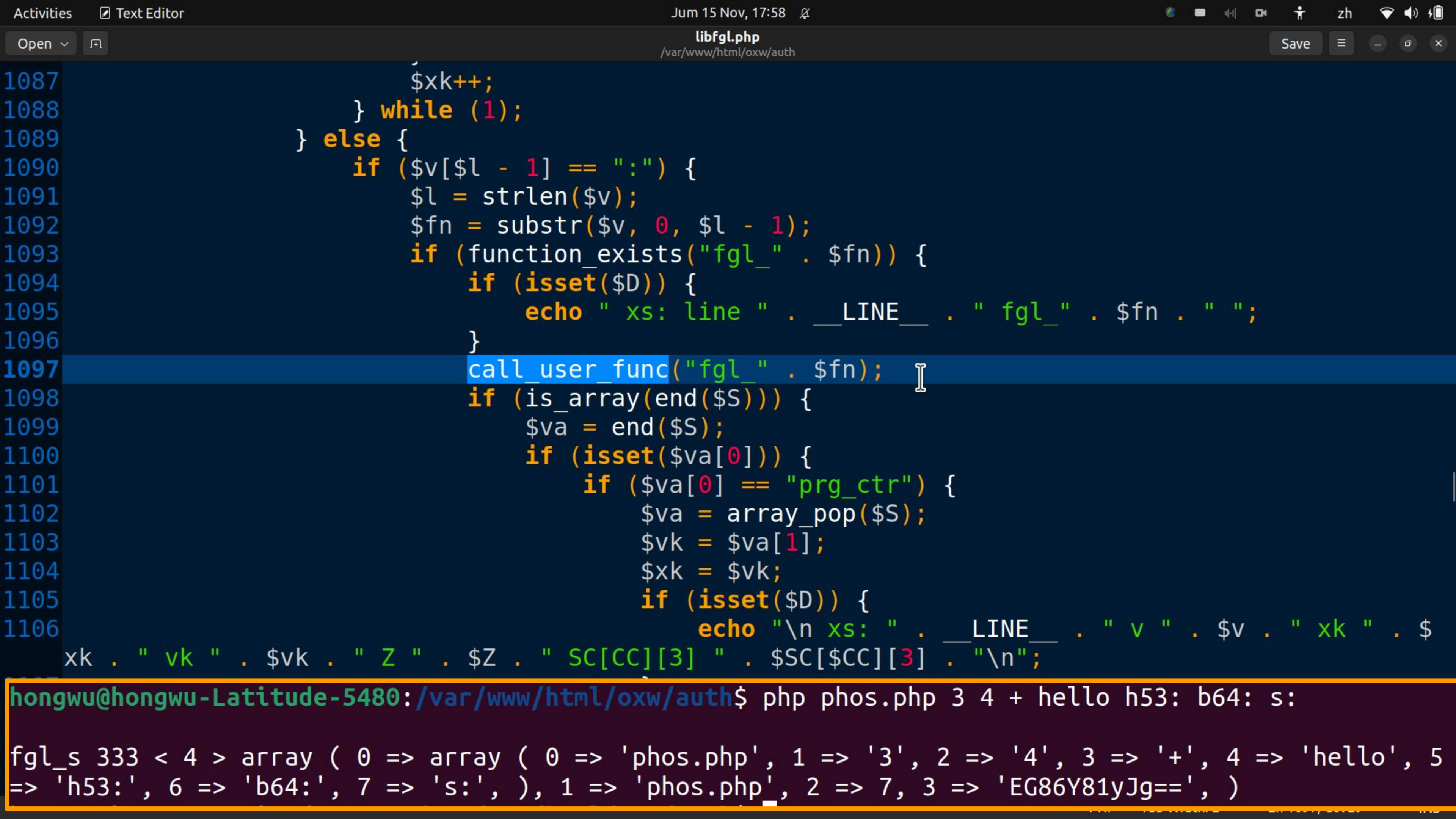
github.com/udexon/SymForth/blob/master/benchmarks/expandv.cpp

master SymForth / benchmarks / **expandv.cpp** Top

Code Blame Raw

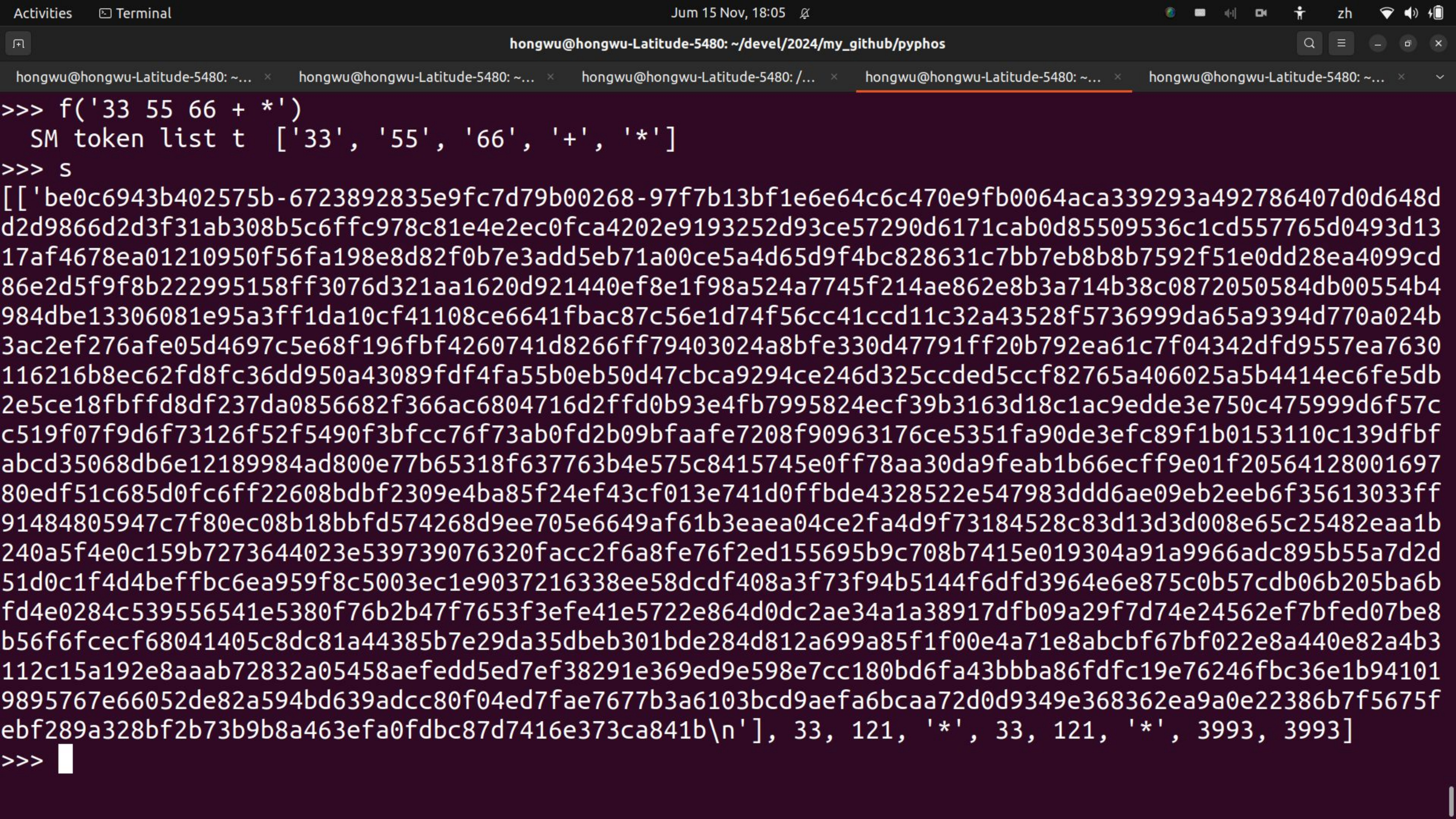
```
int sm_string( std::string str0 )
```

```
                                     }
                                     else if (token.compare("sym:") == 0) { // SymEngine::expand1 terms

                                       sprintf(sm_str, "%s", mystack.top().c_str() );
                                       printf("sym: TOS %s\n", sm_str);

                                       RCP_S.push( symbol( sm_str ) );

                                       RCP<const Basic> e, r, t;

                                       e = RCP_S.top();

                                       std::cout << "Expanding: " << *e << std::endl;

                                       mystack.pop();
```

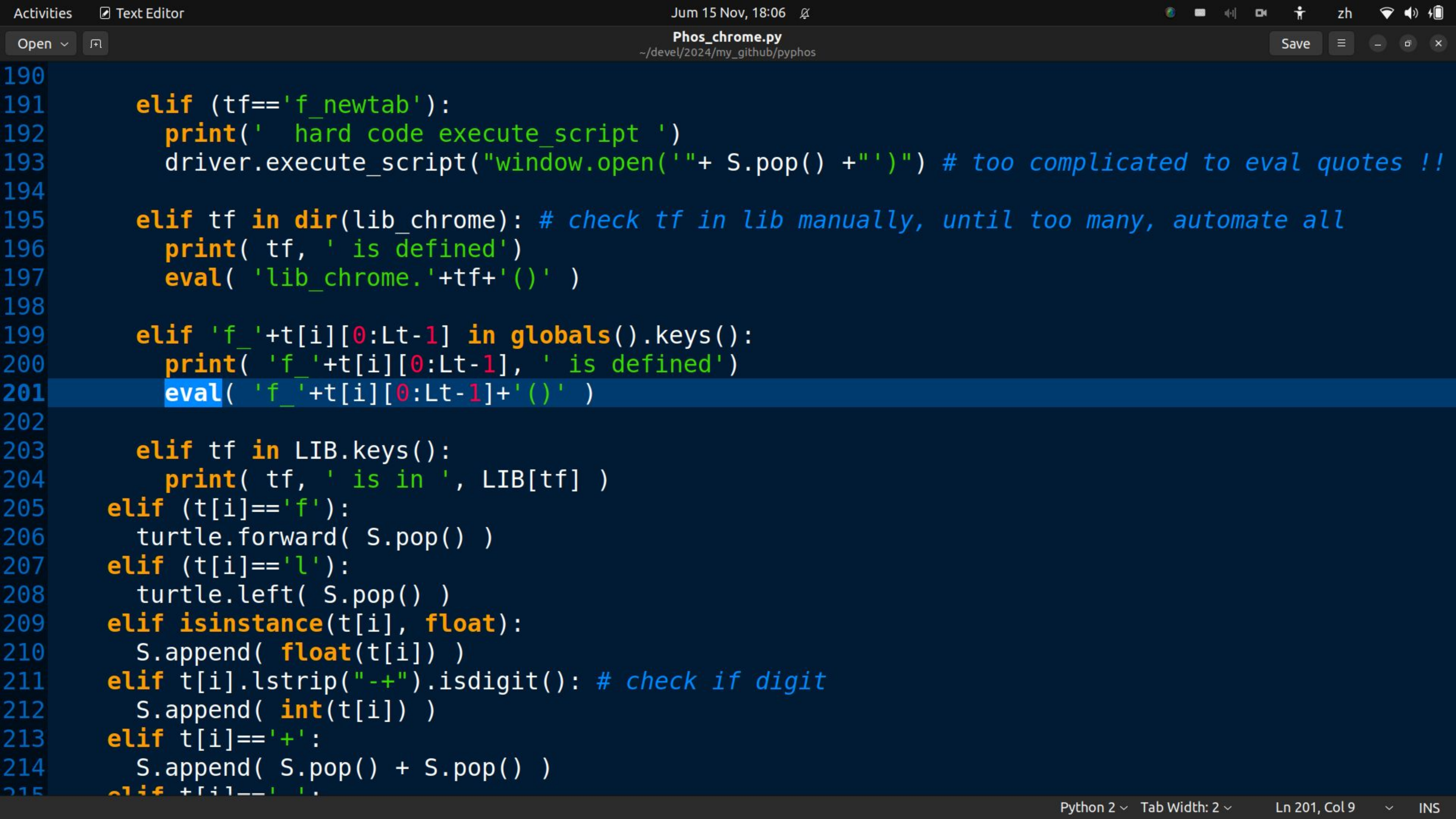


```
            $xk++;
          } while (1);
        } else {
          if ($v[$l - 1] == ":") {
            $l = strlen($v);
            $fn = substr($v, 0, $l - 1);
            if (function_exists("fgl_" . $fn)) {
              if (isset($D)) {
                echo " xs: line " . __LINE__ . " fgl_" . $fn . " ";
              }
              call_user_func("fgl_" . $fn);
              if (is_array(end($S))) {
                $va = end($S);
                if (isset($va[0])) {
                  if ($va[0] == "prg_ctr") {
                    $va = array_pop($S);
                    $vk = $va[1];
                    $xk = $vk;
                    if (isset($D)) {
                      echo "\n xs: " . __LINE__ . " v " . $v . " xk " . $xk . " vk " . $vk . " Z " . $Z . " SC[CC][3] " . $SC[$CC][3] . "\n";
```

```
hongwu@hongwu-Latitude-5480:/var/www/html/oxw/auth$ php phos.php 3 4 + hello h53: b64: s:

fgl_s 333 < 4 > array ( 0 => array ( 0 => 'phos.php', 1 => '3', 2 => '4', 3 => '+', 4 => 'hello', 5 => 'h53:', 6 => 'b64:', 7 => 's:', ), 1 => 'phos.php', 2 => 7, 3 => 'EG86Y81yJg==', )
```


Activities
Terminal
Jum 15 Nov, 18:05
zh
hongwu@hongwu-Latitude-5480: ~/devel/2024/my_github/pyphos
hongwu@hongwu-Latitude-5480: ~...
hongwu@hongwu-Latitude-5480: ~...
hongwu@hongwu-Latitude-5480: /...
hongwu@hongwu-Latitude-5480: ~...
hongwu@hongwu-Latitude-5480: ~...

```
>>> f('33 55 66 + *')
  SM token list t  ['33', '55', '66', '+', '*']
>>> s
[['be0c6943b402575b-6723892835e9fc7d79b00268-97f7b13bf1e6e64c6c470e9fb0064aca339293a492786407d0d648d
d2d9866d2d3f31ab308b5c6ffc978c81e4e2ec0fca4202e9193252d93ce57290d6171cab0d85509536c1cd557765d0493d13
17af4678ea01210950f56fa198e8d82f0b7e3add5eb71a00ce5a4d65d9f4bc828631c7bb7eb8b8b7592f51e0dd28ea4099cd
86e2d5f9f8b22995158ff3076d321aa1620d921440ef8e1f98a524a7745f214ae862e8b3a714b38c0872050584db00554b4
984dbe13306081e95a3ff1da10cf41108ce6641fbac87c56e1d74f56cc41ccd11c32a43528f5736999da65a9394d770a024b
3ac2ef276afe05d4697c5e68f196fbf4260741d8266ff79403024a8bfe330d47791ff20b792ea61c7f04342dfd9557ea7630
116216b8ec62fd8fc36dd950a43089fdf4fa55b0eb50d47cbca9294ce246d325ccded5ccf82765a406025a5b4414ec6fe5db
2e5ce18fbffd8df237da0856682f366ac6804716d2ffd0b93e4fb7995824ecf39b3163d18c1ac9edde3e750c475999d6f57c
c519f07f9d6f73126f52f5490f3bfcc76f73ab0fd2b09bfaafe7208f90963176ce5351fa90de3efc89f1b0153110c139dfbf
abcd35068db6e12189984ad800e77b65318f637763b4e575c8415745e0ff78aa30da9feab1b66ecff9e01f20564128001697
80edf51c685d0fc6ff22608bdbf2309e4ba85f24ef43cf013e741d0ffbde4328522e547983ddd6ae09eb2eeb6f35613033ff
91484805947c7f80ec08b18bbfd574268d9ee705e6649af61b3eaea04ce2fa4d9f73184528c83d13d3d008e65c25482eaa1b
240a5f4e0c159b7273644023e539739076320facc2f6a8fe76f2ed155695b9c708b7415e019304a91a9966adc895b55a7d2d
51d0c1f4d4beffbc6ea959f8c5003ec1e9037216338ee58dcdf408a3f73f94b5144f6dfd3964e6e875c0b57cdb06b205ba6b
fd4e0284c539556541e5380f76b2b47f7653f3efe41e5722e864d0dc2ae34a1a38917dfb09a29f7d74e24562ef7bfed07be8
b56f6fcecf68041405c8dc81a44385b7e29da35dbeb301bde284d812a699a85f1f00e4a71e8abcbf67bf022e8a440e82a4b3
112c15a192e8aaab72832a05458aefedd5ed7ef38291e369ed9e598e7cc180bd6fa43bbba86fdfc19e76246fbc36e1b94101
9895767e66052de82a594bd639adcc80f04ed7fae7677b3a6103bcd9aefa6bcaa72d0d9349e368362ea9a0e22386b7f5675f
ebf289a328bf2b73b9b8a463efa0fdbc87d7416e373ca841b\n'], 33, 121, '*', 33, 121, '*', 3993, 3993]
>>> 
```



```python
      elif (tf=='f_newtab'):
        print('  hard code execute_script ')
        driver.execute_script("window.open('"+ S.pop() +"')") # too complicated to eval quotes !!

      elif tf in dir(lib_chrome): # check tf in lib manually, until too many, automate all
        print( tf, ' is defined')
        eval( 'lib_chrome.'+tf+'()' )

      elif 'f_'+t[i][0:Lt-1] in globals().keys():
        print( 'f_'+t[i][0:Lt-1], ' is defined')
        eval( 'f_'+t[i][0:Lt-1]+'()' )

      elif tf in LIB.keys():
        print( tf, ' is in ', LIB[tf] )
    elif (t[i]=='f'):
      turtle.forward( S.pop() )
    elif (t[i]=='l'):
      turtle.left( S.pop() )
    elif isinstance(t[i], float):
      S.append( float(t[i]) )
    elif t[i].lstrip("-+").isdigit(): # check if digit
      S.append( int(t[i]) )
    elif t[i]=='+':
      S.append( S.pop() + S.pop() )
```

Python 2 Tab Width: 2 Ln 201, Col 9 INS

Probability of success?

Cloning: GNU & Linux began by cloning UNIX.

Now most commercial software companies use unspecified amount of free software.

Modify free software licence – separation of disclosure & royalties.

Track usage (execution) of free software using hashcodes – claim payments.

BitTorrent used to be carrying >30% of all Internet traffic.

Heavy censorship on popular platforms such as Reddit & TikTok – decentralised social media – impossible to censor – *FILTER model – users may choose various types of filters to filter unwanted text, videos, photos or user IDs* – data stored on devices owned by users – or access control belongs to users, not platform.

Part E: FORTH roles in future

Bitcoin & cryptocurrencies actually made FORTH (variant) *the most valuable and popular* (by number of devices, nodes, wallets) in the world – (need decentralised systems to gather statistics!!)

Omni*Web "trojan horse" strategy will (hopefully) attract more programmers to use FORTH (Phoscript or dialects) as interface programming language, if Omni*Web plan of making *Omnihash User ID as the universal decentralised ID* works.

Metaprogramming & Mathematics are important – SymForth example.

Decentralised AI – ideas threatening commercial interests are filtered – *hash based indices allowed AI training data to be distributed in users computers & mobile devices* – much bigger (?) than commercially owned assets?

Same dilemma in history – *land grab from farmers* – now land owned by nation state or corporation – do we want this to happen again – *now we know we can prevent it?*

Although we have used ideological terms such as *techno feudalism* and *techno socialism* to express our views of reducing global rich poor gaps through Decentralised systems, the Chinese or Asian *principle of Taichi views conflicts* not as binary black versus white, but *as forces in opposition feeding on each other*, as quoted by Danish Physicist Niels Bohr in Latin as *Contraria sunt complementa* or in English *"opposites are complementary".*

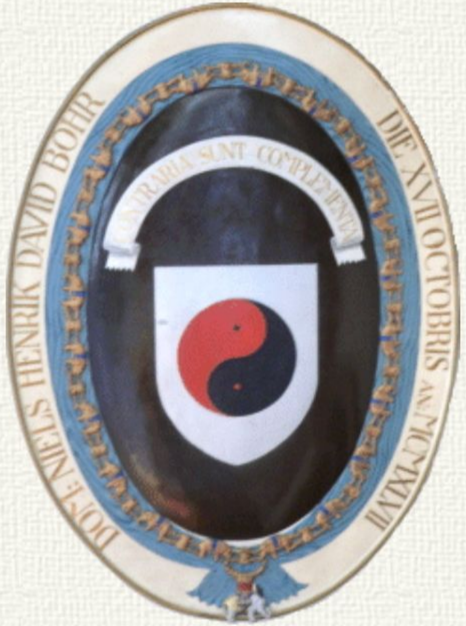
This philosophy can be seen in the *Chinese Weiqi or Go chess*, where on a board with 19 by 19 lines namely 361 spots, the goal of the game is to simply *outdo the opponent by occupying at least one spot more than the opponent* with the chosen black or white pieces, *instead of killing the King* of the opponent in the European chess.

Technosocialism = *"better pay for some programmers"*, *not threatening Washington & Wall Street.*

Metanarchy = Metaverse + (Mon)-archy = self governance through metaverse

- Trispecies Monetary Payments: Fiat currency, Cryptocurrency, Bullions (Gold/Silver)

Escutcheons of Science

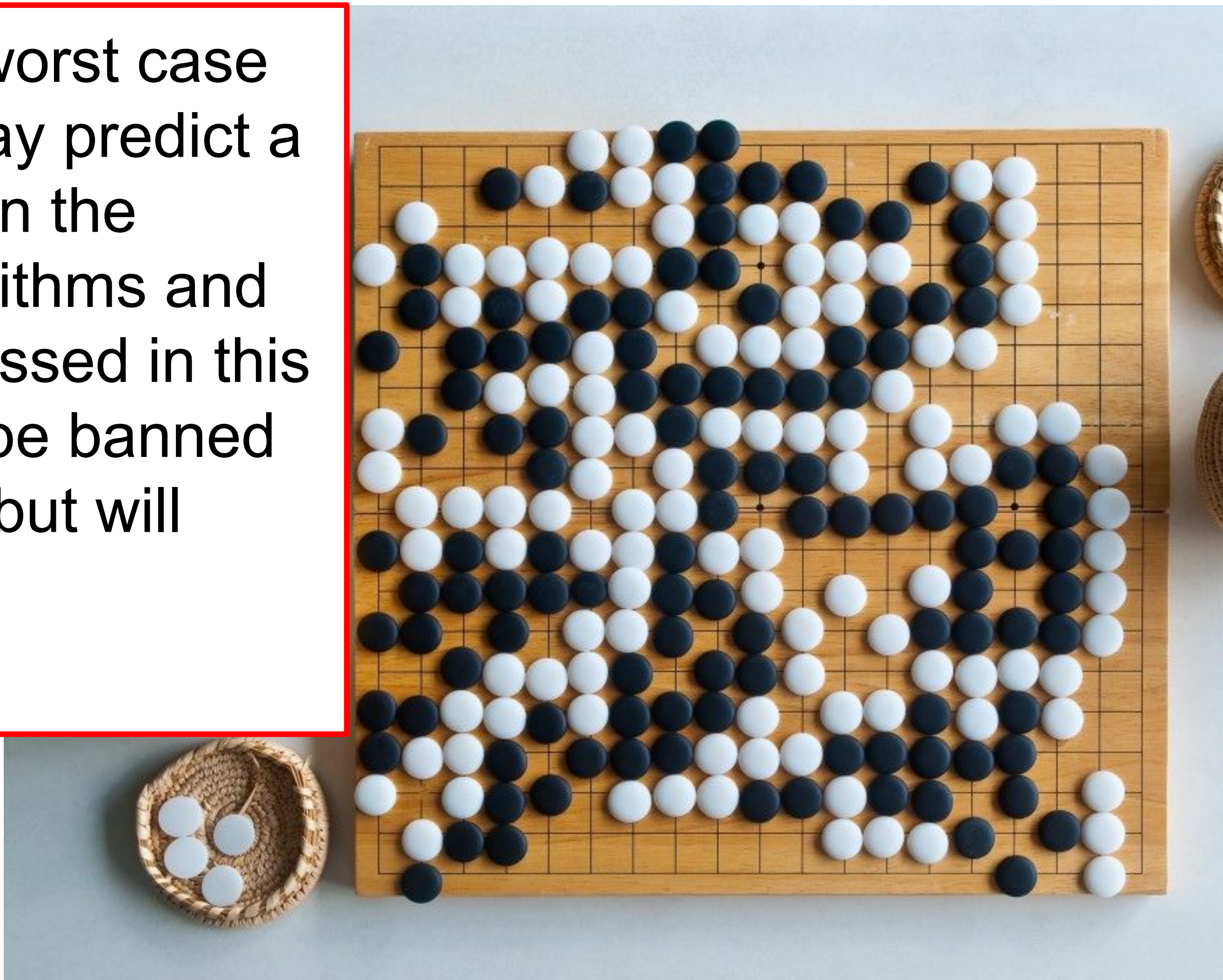

CONTRARIA SUNT COMPLEMENTA
DOCT: NIELS HENRIK DAVID BOHR
DIE XVII OCTOBRIS AN. MCMXLVII

Niels Bohr (1885-1962)
Danish physicist. Nobel laureate in 1922.
Knighted (Order of the Elephant) in 1947.

Argent, a taijitu Gules and Sable.
Motto : *Contraria sunt complementa* (opposites are complementary).

http://www.numericana.com/arms/bohr.htm

Mixture of best & worst case scenario: Weiqi may predict a fragmented world in the future. Some algorithms and technologies discussed in this presentation may be banned in some countries but will thrive in others.
Which?



Science Fiction & Politics

American Russian author Isaac Asimov – Foundation – Psychohistory – Mathematics, Algorithms, Metaprogramming – changing the course of the Galactic Empire.

Recent science fiction on cryptography changing future of mankind? If not, do it in Omni*Web.

Omni*Web Simulation – all hypotheses (plural) can be proven (or disproved) using FORTH (like) metaprogramming languages (plural).