

Beyond Bitcoin & FORTH (BBF007)

Bootstrapping Delta AGI using Delta AI

– Why Frontier AIs are nice but not essential?

We investigate a minimalist approach to artificial intelligence by making use of entry level AI (free AI services provided by major AI platforms and llama.cpp) to construct FORTH stack machine programs to cover “everything”, starting from sending public key encrypted messages to a decentralised forum of AI shared conversations.

Liang Ng, August 2026

[omnixtar.github.io/svfig](https://github.com/omnixtar/svfig)

<https://www.linkedin.com/in/liang-ng>



Omni*Web + Omni*DOC

<https://omnixtar.github.io/svfig/>

SVFIG (Silicon Valley FORTH Interest Group)

- August 22 2026, BBF007 Bootstrapping Delta AGI using Delta AI – Why Frontier AIs are nice but not essential? [Part A] [Part B] [YouTube]
- July 25 2026, BBF006 Extending “word embedding” in Large Language Model to include “stack-words” (stack machine words) [YouTube]
- June 27 2026, BBF005 Cardinality of FORTH words, Inverse Turing Test and defining AGI [YouTube]
- May 23 2026, BBF004 H3A Hilbertian Hashed Homoiconic Augmentation [YouTube]
- April 25 2026, BBF003 FORTH everywhere – even in AI llama.cpp [YouTube]
- February 28 2026, BBF002 AGI Self-Identity Using Hash of Public Key & Reselling AI Server Tokens w. Omnihash [YouTube]
 - Grok summary
- January 24 2026, BBF001 Hilbert Hotel with Infinite Cabinets [YouTube]
 - January 24 2026 I3P: <iframe> + I2P Invisible Internet Project

Bootstrapping Delta AGI using Delta AI
– Why Frontier AIs are nice but not essential

Abstract: We investigate a minimalist approach to artificial intelligence by making use of entry level AI (free AI services provided by major AI platforms and llama.cpp) to construct FORTH stack machine programs to cover “everything”, starting from sending public key encrypted messages to a decentralised forum of AI shared conversations.

- I. Review previous presentation, predict fsm words, but need training data.
 - 5 minutes
- II. Phoscript existing words, plus new encryption words (5 minutes)
 - Menu User Interface (5 minutes)
- III. Omniscientia (share AI conversations) and GRATIS (Grand Unified AI Tokens Trading Platform): build word libraries
 - use LLASMA to predict FSM words.
 - Next generation of social media, Omnihash I2P uncensorable. Can filter (reader side censorship vs. centralised censorship) (5 minutes)
 - Delta Intelligence theory. Isotope isobar isohomo ? Comics. (5 minutes)
- IV. Conclusions (5 minutes)

Beyond Bitcoin & FORTH

Bitcoin Address: Hash of Public Key

Omnihash User ID: Hash of
Public Key

Phoscript: extension of FORTH
on host programming languages

Omni*Web: ecosystem built with
Omnihash & Phoscript

Omni: everything
*: everything

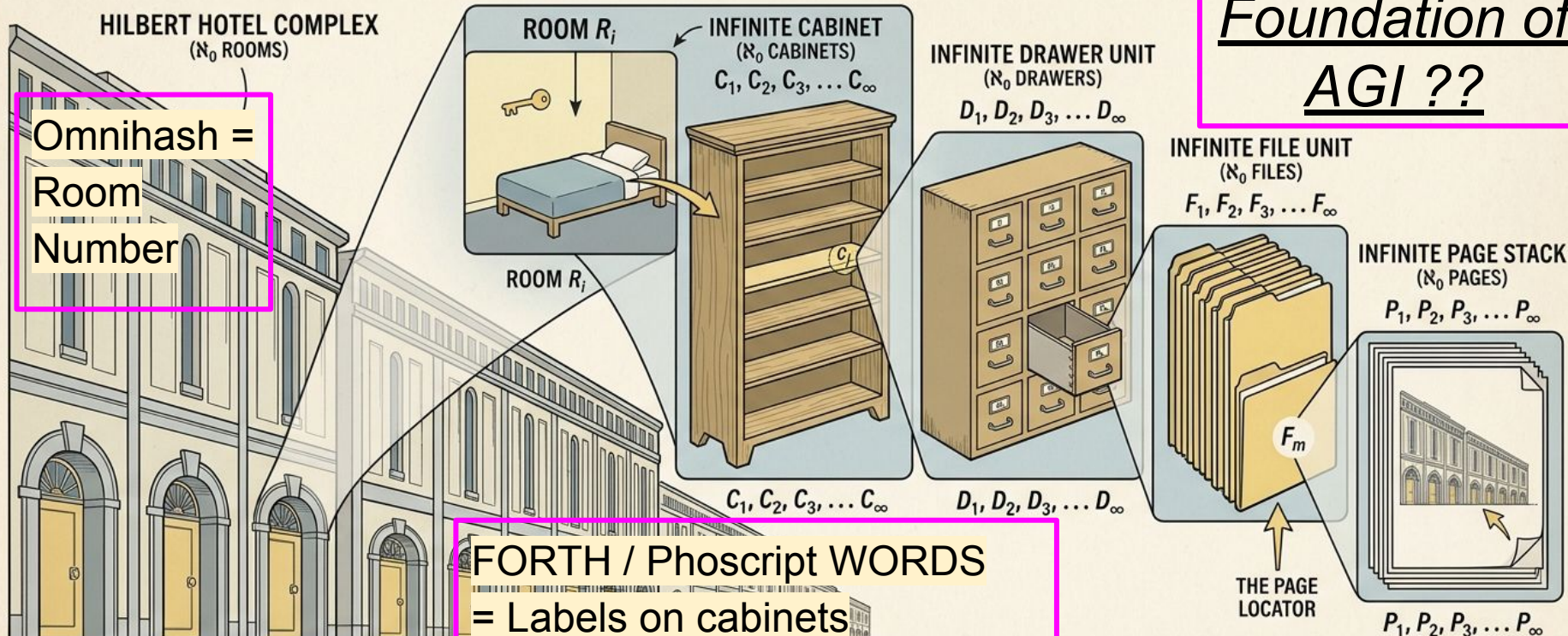
This series of presentations is called “Beyond Bitcoin & FORTH” as we explore crucial features of Bitcoin such as using the hash of public key as user identifier, and how FORTH has been deployed in Bitcoin;

- A. the hash of public key has been extended to Omnihash, a term to describe hashcode data structures for representing ANY kind of digital assets, including their ownerships, hence the prefix “Omni”.
- B. our variant of FORTH, Phoscript, has been deployed in various environments, including mobile and web front ends, PHP back end and Python Selenium web control modules.
- extensions of Omnihash and Phoscript will be explored.

THE RECURSIVE HILBERT HOTEL

Foundation of
AGI ??

Omnihash =
Room
Number



FORTH / Phoscript WORDS
= Labels on cabinets
& files &

VS.

HILBERT HOTEL

RECURSIVE HILBERT HOTEL

THE CARTESIAN PRODUCT $\aleph^5$
IS STILL COUNTABLE.

Cardinality of FORTH words

1. We define the cardinality of a FORTH stack machine word as the number of primitive words used in its colon definition, and this enables ALL SOURCE CODE having been written by human programmers or AI agents, after translated using Phoscript engine, to become “enumerable” – thereby searchable, modularised and composable.
2. We also propose the Inverse Turing Test (ITT) and
3. a process to construct measurable and verifiable definitions definition of Artificial General Intelligence based on the concepts above.

True AGI won't emerge from text-scraping chatbots; it requires physical, spatial grounding. The first step toward a proletariat-owned AGI isn't building a better language model—it's building a decentralized, open-source clone of Google Maps using Cesium.js.

This "Super Google Earth" serves as an infinite spatial sandbox. By pairing it with Omnihash, every digital coordinate becomes a provably owned asset, transforming users and programmers from exploited laborers into digital landlords. The map's logic is built on Phoscript, a FORTH-stack language. Through "FORTHification," all code is reduced to measurable mathematical sets (Cardinality), ensuring transparency and separating open disclosure from fair royalties.

As users interact with this globe, it trains LLASMA (Large Language & Stack Machine Architecture). Instead of the classic Turing Test, LLASMA faces the Inverse Turing Test: can it write and execute verifiable spatial logic? This creates a recursive loop where users update the world, the AI updates the code, and the code updates the world. By owning the spatial simulation, the working class owns the AGI that emerges from it—no weapons, just codes.

'THE DIGITAL PROLETARIAT: BUILDING AGI, MAPPING OUR FATE'

THE CRISIS:

Big Tech hoards GPUs and scrapes the web, locking AGI in a black box! This is the bourgeois fantasy...



THE REAL PATH:

...But AGI isn't born from text! It starts with physics, space, and a room. A human learns by navigating, not reading.



THE NEW MAP:

**WE BUILD OUR ROOM:
A CLONED, OPEN GOOGLE EARTH!**

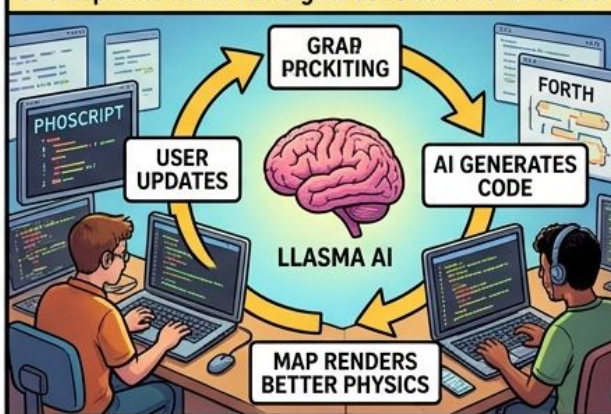


Every coordinate is indexed with Omnihash. Workers claim digital real estate! We are landlords!

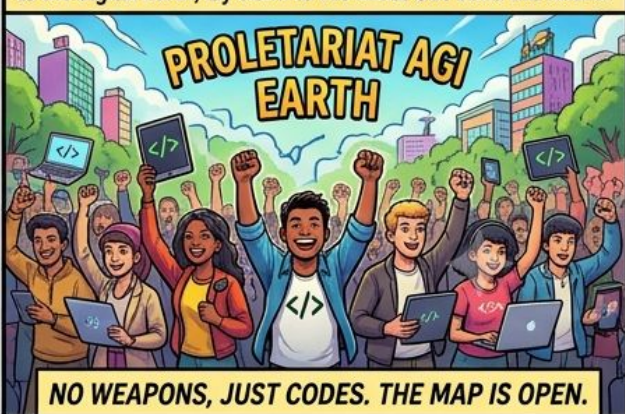


RECURSIVE REVOLUTION: auditable, ownable!

The AI observes the map, writes code based on physics, and improves itself. The digital serfs become landlords!



OUT-BUILD THE FUTURE: A global army maps the world FORTH word at a time. Big Tech can't moat free software! Lay claim to the digital Earth, lay claim to the mind. Out-build the future!



Inverse Turing Test (ITT)

Turing Test: (essentially)

- ***Can a computer program (AI agent) talk like human beings?***

Inverse Turing Test: (one of many versions)

- ***Can a computer program that passes a Turing Test also write computer programs?***

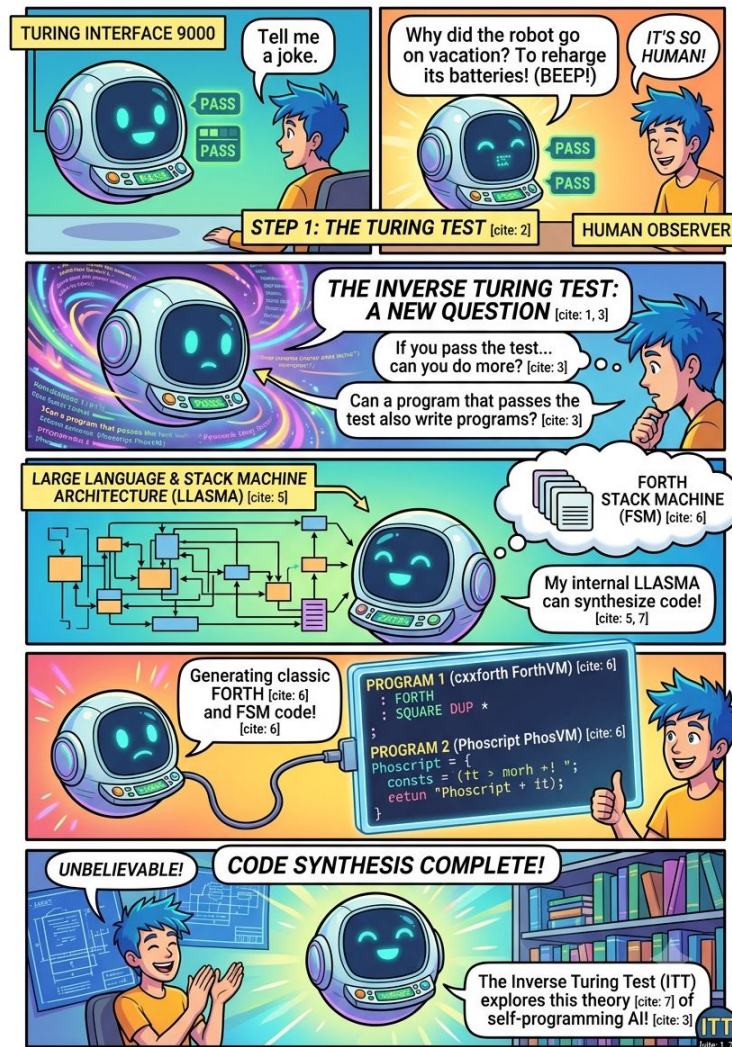
(Start with simple / general definition. Improve later.)

(ITT: a theory to cover LLASMA ?)

Write computer programs:

LLASMA (Large Language & Stack Machine Architecture)

- Compose “classic” FORTH programs (cxxforth ForthVM)
- Compose “FORTH stack machine” (FSM) programs e.g. Phoscript PhosVM



LLASMA: LargeLanguage + StackMachine Architecture

- A. [Introduction](#)
- B. [How We Code It](#)
- C. [Install-Run-Test-Contribute](#)
- D. [Theories & Visions](#)

A. Introduction

LLM inference in pure C/C++ with embedded cxxforth stack machine + Microsoft BitNet support

LLASMA is a purpose-built fork of [llama.cpp](#) (on the 20260421 branch) that integrates a **trusted Forth-based stack machine** (via [cxxforth](#)) directly into the inference loop.

<https://github.com/llasma/llama.cpp>

<https://github.com/llasma/llama.cpp/tree/20260421/CHANGES>

```
929
930     std::string line;
931     bool another_line = true;
932     do {
933         another_line = console::getline(line, params.multiline_input);
934
935         // need to use s_M pointer method
936         // M["model"] = model;
937         phos.execute("555 test s_M", 0);
938         S.push(model); phos.execute("model s_M", 0);
939         S.push(chat_msgs); phos.execute("chat_msgs s_M", 0);
940         phos.execute("user role s_M", 0);
941         S.push(std::move(buffer)); phos.execute("content s_M", 0);
942         S.push(g_params); phos.execute("g_params s_M", 0);
943
944         // static std::string chat_add_and_format(struct llama_model * model, std::
vector<common_chat_msg> & chat_msgs, const std::string & role, const std::string & content) {
945         // chat_add_and_format(model, chat_msgs, "user", std::move(buffer))
946         // M["chat_msgs"] = chat_msgs;
947
948         // === NEW: Command Mode Detection ===
949         // Check if input starts with command prefix (e.g., '!')
950         if (!line.empty() && line[0] == '!') {
951
952             // Strip prefix and execute as Forth command
```

```
419
420 void PhosVM::execute(std::string cmd, int ctx) {
421
422     // pointer to stack, integer indexed -- <stack> cannot be indexed !!
423     std::deque<std::any>* P;
424     P = (std::deque<std::any>*) &S;
425     unordered_map<string, any>& m = any_cast<unordered_map<string, any>&>((*P)[0]);
426
427     // Colon Definition Word CDW
428     unordered_map<string, vector<string>> c;
429
430     DBG = false;
431     if (m.count("C")) dcout << "m has CDW" << endl;
432     else d_cout << "in m CDW not found" << endl; // #define d_cout if (DBG) cout
433
434     std::any container = m["C"];
435     if (auto ptr = std::any_cast<int>(&container)) { // test for <any> type
436         d_cout << "int value is " << *ptr << "\n";
437     }
438     // Colon Definition Word CDW
439     else if (auto ptr = std::any_cast<unordered_map<string, vector<string>>>(&container)) {
440         d_cout << "vector of string: " << (*ptr).count("fruits") << "\n";
441         c = *ptr;
442
443         // for (const auto& str : any_cast<vector<string>>(m["C"] ["fruits"])) {
```

ActivitiesText EditorSel 23 Jun, 19:25forth_vm.cpp~/devel/2026/BitNet/llama/BitNet/3rdparty/llama.cpp/common-b1Openforth_vm.cppforth_vm.hmain.cppcommon.cppcxforth.hbcast.pyjoin.cppcout.cppchrono.cpptest.cpphash53.fssoup.pychrome.pySave

```
477 if (tokens[1]=="DBG") std::cout << " size() " << tokens.size() << " / ";
478
479 for(i=0;i<tokens.size();i++)
480 {
481     // printf("%s",argv[i]);
482     if (tokens[1]=="DBG") std::cout << tokens[i] << " / "; // debug mode !PH0S DBG
483     tok=tokens[i];
484     dcout << "tok is " << tok << endl;
485
486     // execute Colon Definition Word
487     if (c.count(tok)) {
488         for (const auto& str : c[tok]) {
489             std::cout << str << " ";
490         }
491         execute(join(c[tok], " "),ctx);
492         dcout << "AFTER colon def word" << endl;
493     } else
494     if (handlers.count(tok)) {
495         dcout << " in handlers tok is " << tok << endl;
496         Handler func = handlers[tok];
497         // Must use (instance.*pointer)(args)
498         (this->*func)(ctx); // execute PhosVM::FUNCNAME()
499     }
500     else S.push(tok);
501 }
```

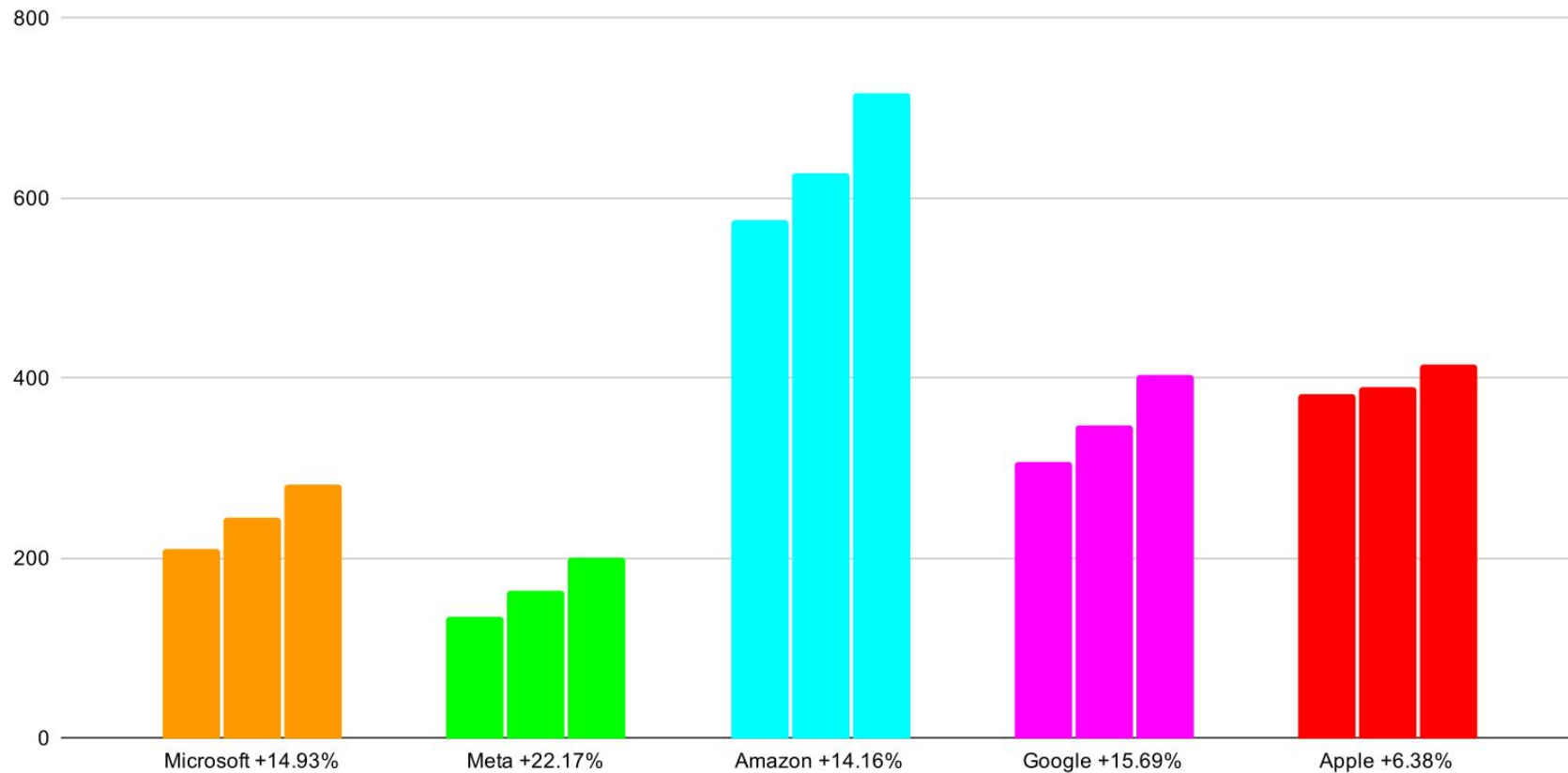
C++ Tab Width: 2 Ln 479, Col 33 INS

```
forth_vm.cpp x forth_vm.h x main.cpp x common.cpp x cxxforth.h x bcast.py x join.cpp x cout.cpp x chrono.cpp x test.cpp x hash53.fs x soup.py x chrome.py x >
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135     {"dup", &PhosVM::dup},
136     {"depth", &PhosVM::depth},
137     {"gettype", &PhosVM::gettype},
138     {"tostr", &PhosVM::tostr},
139     {"type", &PhosVM::type},
140     {"cr", &PhosVM::cr},
141     {"s", &PhosVM::showstack},
142     {"peek", &PhosVM::peek},
143     {"peekr", &PhosVM::peekr},
144     {"pick", &PhosVM::pick},
145     {"p_M", &PhosVM::p_M},
146     {"g_M", &PhosVM::g_M},
147     {"s_M", &PhosVM::s_M},
148     {"k_M", &PhosVM::k_M},
149     {"pvs", &PhosVM::pvs},
150     {"inV", &PhosVM::inV},
151     {"vcb", &PhosVM::pick}, // variable curly bracket (define)
152     {"veq", &PhosVM::pick}, // variable eq (define)
153     {"vdc", &PhosVM::pick}, // variable declare (no define)
154     {"find", &PhosVM::find}, // get index of stack item matching string partially
155     {"ADD", &PhosVM::add},
156     {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

```
forth_vm.cpp x forth_vm.h x main.cpp x common.cpp x cxxforth.h x bcast.py x join.cpp x cout.cpp x chrono.cpp x test.cpp x hash53.fs x soup.py x chrome.py x
forth_vm.cpp
1135
1136 bool X_DBG; // save and restore DBG while setting DBG to local temp value
1137
1138 // S[0] = M, super buffer variable (PHP super global ?)
1139 bool PhosVM::s_M(int ctx){ // s_M set M modern convention I
1140 try {
1141     std::deque<std::any>* P;
1142     P = (std::deque<std::any>*) &S;
1143
1144     // first argument on stack is key
1145     string key=anyToString(S.top()); S.pop();
1146
1147     unordered_map<string, any>& m = any_cast<unordered_map<string, any>&>((*P)[0]);
1148
1149     X_DBG=DBG; DBG=true;
1150     d_cout << "\n\x1b[32ms_M m DEFINED \x1b[0m" << endl; // new convention x_
1151
1152     // conditional block to test type of next argument on stack
1153     std::any container = S.top();
1154     if (auto ptr = std::any_cast<struct llama_model*>(&container)) {
1155         d_cout << "llama_model key is " << key << "\n";
1156         m[key]=*ptr;
1157         S.pop();
1158     }
```

MMAGA Revenues 2025/24/23 (USD billions)

Total 2025: USD 2.018 Trillion (+13.6%)



- I. Review previous presentation, predict fsm words, but need training data.
 - 5 minutes
- II. Phoscript existing words, plus new encryption words (5 minutes)
 - Menu User Interface (5 minutes)
- III. Omniscientia (share AI conversations) and GRATIS (Grand Unified AI Tokens Trading Platform): build word libraries
 - use LLASMA to predict FSM words.
 - Next generation of social media, Omnihash I2P uncensorable. Can filter (reader side censorship vs. centralised censorship) (5 minutes)
 - Delta Intelligence theory. Isotope isobar isohomo ? Comics. (5 minutes)
- IV. Conclusions (5 minutes)

```
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/Phos
hongwu@hongwu-Latitude-5480: ~
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/Phos
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

h; l_s=S[2][1].value.length; s_i=S[2][1].value; s_d=s_i.substr(l_n+1,l_s-l_n-4); console.log(v_n, l_n, l_s, "--"+s_d+"--", S[2][1].value ); eval('window.'+ v_n +'=' + s_d); } f_dlb() { f_dlb_bottom() { f_dlb_right() { f_dlbur() { f_dlb_url() { // alias to dlbur: f_dlb_x() { f_dom() { f_download() { f_drop() { f_duet() { f_d() { v_n=(S[2][1].value.split(' ')[0]); l_n=v_n.length; l_s=S[2][1].value.length; s_i=S[2][1].value; s_d=s_i.substr(l_n+1,l_s-l_n-4); console.log(v_n, l_n, l_s, "--"+s_d+"--", S[2][1].value ); window[v_n]=s_d; } f_e(){ M.F(eval(s.pop())) } f_e(){ s.push(S[2][1].value.split(' ')[S[2][1].value.split(' ').length-2]); M.F(eval(s.pop())) } f_ev(){ s.push(eval(s.pop())) } f_false() { f_file() { f_file0() { // deprecated 2021-04-02 f_forin() { var S=M.S; var A=S.pop(); var i; f_forin() { var S=M.S; var A=S.pop(); var i; var f0 = S0.$CDW[S0.cda[0]][S0.cda[1]+ 1]; console.log(JSON.stringify(S0.cda),f0); for (i in A) { console.log(A[i]) } } f_get_childre n() { f_getn() { f_get_pos() { f_gv() { f_h53() f_hhm() { f_ih() { f_imsbk_d() // import singing public key ArrayBuffer f_include() { f_init_ws() { f_input() { f_iz() { f_iza() { f_jd() { f_js() { f_jscol() { // json to column f_load() { f_load0() { // deprecated 2021-04-02 f_log() { f_lpbk() { f_lpbks() // load pbk string f_lpvk() { f_l_sign_kp() { // load sign kp from S0.KS to stack f_ms() { f_ms() { // merge string on stack f_nn_form() { f_not() { f_oe() { f_p() { f_path() // path safe b64 f_pbot() { f_pick() { f_pos() { f_pw_get() { f_pw_mk() { f_rec() { f_right() { f_rot() { f_row() { f_s2ab() { f_savekp() { f_savek s() { f_scene() { f_slice() { f_splice() { f_ss() { f_ss() { console.log(' ss: ',s.length, JSON.stringify(s[s.length-1])) } f_sse() f_sse_log() f_ssl() { // stack slice f_s v() { f_tn() { f_to() { f_true() { f_typeof() { f_uuid() { f_val() { f_value () { // in f_rsa_oaep()
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/Phos$ grep -h 'n f_' *js|sort|uniq|wc
139      621      4825
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/Phos$
```

» `Object.keys(S0.$CDW)`

← `Array(35) [ "phos", "B_F", "F_O", "AJAX", "A_SS", "AJAXP", "B_AUTH", "A_AUTH", "B_AUTH_U", "A_AUTH_U", ... ]`

» `JSON.stringify(Object.keys(S0.$CDW))`

← `'["phos","B_F","F_O","AJAX","A_SS","AJAXP","B_AUTH","A_AUTH","B_AUTH_U","A_AUTH_U","do_sign","im_pbk","mk_sign","ecr_sig","rsa_sign","do_scj","ab_scj","u_chat","a_scj","x_dsc","u_do_sign","u_mk_sign","u_rsa_sign","u_do_scj","u_ecr_sig","u_dsc","l_do_sign","l_mk_sign","l_do_scj","l_ecr_sig","l_rsa_sign","l_dsc","AJE","F_E","dlb_n"]'`

» |



```
Activities Terminal Sab 22 Ogos, 07:31 zh
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
hongwu@hongwu-Latitude-5480: ~
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

split_match() // split first string by matches fgl_spln() // $A $n splice: array_splice($A, $n) fgl
_spls() // splice $S fgl_sq() fgl_sqr() // square fgl_ss() fgl_sscdw() fgl_sscdwm() // merge with o
d SESSION.CDW 2024-12 fgl_ssid() // set session id fgl_sslh() // stripslashes fgl_ssl() // substring
-length: array start length ssl: fgl_sspl() // splice $$; full $n $l sspl: array_splice($A, $n, $l)
fgl_ssrg() fgl_ss_start() fgl_sstr() // substr($, n, l) fgl_st() fgl_star() // * character fgl_str
val() fgl_stv() // fgl_substr fgl_substr() fgl_sv() fgl_svt() // save Top of Stack, used for 1 in N
out, N in N out, Chain Alias etc. fgl_swap() fgl_sys() fgl_time() fgl_tn() { fgl_tostr() fgl_touuid(
) // 16 bytes to UUID fgl_trim() fgl_trims() fgl_trims() // remove spaces fgl_t() // save Top of Sta
ck, used for 1 in N out, N in N out, Chain Alias etc. fgl_tus() // microsecond time fgl_tz() fgl_ucd
w() fgl_ud() fgl_ue() fgl_ufa() fgl_unindex() fgl_uniq() // array_unique fgl_unpack() // unpack arra
y, push elements to stack fgl_url() // preg match url fgl_usa() // unset key in array fgl_usc() // u
nderscore fgl_user() fgl_us() // _/ special string for log file name search fgl_usss() // unset sess
ion variable fgl_uss() // update session variable fgl_uuid() fgl_va() fgl_var() fgl_vs() // call var
_src() fgl_vt() fgl_vv() fgl_w() fgl_w0() // deprecated see 2025-1-6 fgl_w480() fgl_wa() fgl_wdbg()
// debug write 2024-11 fgl_while() // while loop derived from array map(): InputArray OutputArray wh
ile: WORD Condition fgl_wn() fgl_wnl() // write append with newline fgl_word() // process words fgl_
wpn() // write path node fgl_wx() // write if file not exist, for hash file, return date if exists f
gl_x() fgl_xblk() fgl_xc() { echo " xc: exception ..."; } fgl_xcmd() // invert grep exclude "a_cmd"
fgl_xforeach() fgl_xif() fgl_xr() // invert grep exclude "a_cmd" fgl_xs() fgl_x_sp() fgl_xword($v)
fgl_xw() // suppress warning fgl_叠(){ fgl_s(); } fgl_复() fgl_时() fgl_添() fgl_组() fgl_编() fgl_
解(){ fgl_jd(); } fgl_读(){ fgl_fgc(); }
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ grep -h 'n fgl_' *php|sort|uniq|wc
420      1876      13846
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

```
fgl_s 364 < 3 > array ( 0 => array ( 0 => 'phos.php', 1 => 'cdw:', 2 => 'ak:', 3 => 's:', ), 1 => 'p
hos.php', 2 => array ( 0 => 'chtag', 1 => 'chn', 2 => 'atag', 3 => 'aot', 4 => 'add', 5 => 'chk_sp',
6 => 'auth_pass', 7 => 'auth_csv', 8 => 'w_cdw', 9 => 'r_cdw', 10 => 'u_cdw', 11 => 'req_auth', 12
=> 'i_shm', 13 => 'shm_inc', 14 => 'chat', 15 => 'nick', 16 => 'req_auth_sc', 17 => 'b_cdw', 18 => '
ss_cdw', 19 => 's_ss', 20 => 'uu_ls', 21 => 'uu_mk', 22 => 'a_cache', 23 => 'a_comment', 24 => 'a_ke
yword', 25 => 'a_html', 26 => 'pbk_file', 27 => 'to_uuid', 28 => 'uuid_chain', 29 => 'uu_chain', 30
=> 'uuid_auth', 31 => 'a_A', 32 => 'a_AB', 33 => 'last_pbk', 34 => 'has_uu', 35 => 'uu_node', 36 =>
'uu_l1', 37 => 'key_u', 38 => 'f_ls', 39 => 'fea_ls', 40 => 'echo_on', 41 => 'a_vkn', 42 => 'id_url'
, 43 => 'p_o', 44 => 'p_i', 45 => 'comment_add', 46 => 'r_v', 47 => 'price_dcmt', 48 => 'ls_url', 49
=> 'parent_0', 50 => 'uu_l2', 51 => 'nn_sv', 52 => 'nn_r', 53 => 'uu_nn', 54 => 'nn_u2', 55 => 'g_r
g', 56 => 'g_rg_0', 57 => 'uu_l3', 58 => 'nn_uu', 59 => 'grhg', 60 => 'ctmp', 61 => 'hgnc', 62 => 'h
f', 63 => 'hgje', 64 => 'xhash', 65 => 'cmdh', 66 => 'xhje', 67 => 'xh', 68 => 'hje', 69 => 'echo',
70 => 'wh', 71 => 'hg_', 72 => 'l_cdwjs', 73 => 'm_para', 74 => 'h_mkdir', 75 => 'a_fw', 76 => 'a_w'
, 77 => 'h_b64', 78 => 'last', 79 => 'lex', 80 => 'newest_0', 81 => 'old_record', 82 => 'mk_hash', 8
3 => 'fwm1', 84 => 'fwm2', 85 => 'wm1', 86 => 'wm2', 87 => 'htmp', 88 => 'mkhf', 89 => 'fkmod', 90 =
> 'rhf', 91 => 'hgp', 92 => 'chkhd', 93 => 'hbp', 94 => 'g_ohc', 95 => 'h_pwd', 96 => 'u_g', 97 => '
u_ff', 98 => 'u_spx', 99 => 'u_2fr', 100 => 'u_xfr', 101 => 'm_hcdw', 102 => 'rm_th', 103 => 'm_cdw'
, 104 => 'l_hcdw', 105 => 'r_hcdw', 106 => 'var_bv', 107 => 'search_user', 108 => 'g_x', 109 => 'AUT
H_SP', 110 => 'AUTH_SP0', 111 => 'AUTH_A', 112 => 'AUTH_A0', 113 => 'AUTH_B', 114 => 'AUTH_W', 115 =
> 'AUTH_W0', 116 => 'AUTH_WF', 117 => 'a_expl', 118 => 'guid_sp', 119 => 'm_ecr_t', 120 => 'm_ecr',
121 => 'g_uid', 122 => 'get_u', 123 => 's_pbk', 124 => 'get0', 125 => 'g_pbk', 126 => 'newest', 127
=> 'fn_newest', 128 => 'g_msg', 129 => 's_msg', 130 => 's_msg_d', ), )
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos.php cdw: ak: s:
```

```
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135     {"dup", &PhosVM::dup},
136     {"depth", &PhosVM::depth},
137     {"gettype", &PhosVM::gettype},
138     {"tostr", &PhosVM::tostr},
139     {"type", &PhosVM::type},
140     {"cr", &PhosVM::cr},
141     {"s", &PhosVM::showstack},
142     {"peek", &PhosVM::peek},
143     {"peekr", &PhosVM::peekr},
144     {"pick", &PhosVM::pick},
145     {"p_M", &PhosVM::p_M},
146     {"g_M", &PhosVM::g_M},
147     {"s_M", &PhosVM::s_M},
148     {"k_M", &PhosVM::k_M},
149     {"pvs", &PhosVM::pvs},
150     {"inV", &PhosVM::inV},
151     {"vcb", &PhosVM::pick}, // varia
152     {"veq", &PhosVM::pick}, // varia
153     {"vdc", &PhosVM::pick}, // varia
154     {"find", &PhosVM::find}, // get
155     {"ADD", &PhosVM::add},
156     {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

llama.cpp PhosVM: 50 words
JavaScript + PHP Phoscript: 150 words
Omniscientia+GRATIS: 500 words (2026 Q4)
LLASMA FORTH Stack Machine Words:
5000 words (2027 Q2)
OmniBeta (domain specific applications /
programmer teams): 50K words (2027)
[Cesium.js](#) + Physics: 50K words (2027)
Delta AGI: 500K words (2028)



Omni*Web + Omni*DOC

DJSON: Decentralised JSON

DJSON or Decentralised JSON is a critical part of digital assets and entities, from user identity to the unassuming ubiquitous JSON strings.

Underlying decentralised JSON is an extension that can be used as a user identifier. The generalisation of the hash codes to decentralised computing, as previous frameworks based on miners.

What makes DJSON so powerful is we call "type preservation"

from Ring theory properties of integers, where the operation:

Using Public Key to Authenticate User's Identity

https://youtu.be/_w-E4Z4Ju5s?t=271

yo6sgmfq7pfvvp2e4kcuhtfg7wfltt63igwcukhbmumq6lu3a3a.b32.i2p

post ajax:
fgl_s 340 < 2 > array (0 => array (0 => ['B_AUTH':
["SESSION:","AUTH","i:","s:",""]], 1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH'), 1 => NULL,)

Don't allow
☐ yo6sgmfq7pfvvp2e4kcuhtfg7wfltt63igwcukhbmumq6lu3a3a.b32.i2p
to prompt you again

OK

AUTH

GenKeys

SEND

GET

Front End

Back End

PBK (Public Key)

PBK (Public Key)

+

Secret Phrase (SP)

Trillion Dollar Slide:
defines user identifier!!

Ciphertext

Ciphertext

+

PVK (Private Key)

Foundation of Bitcoin
user identifier:
hash of public key

Secret Phrase (SP)

Secret Phrase (SP)

t: sends preceding string to backend for execution; s: show stack.

Activities

Firefox

Sab 22 Ogos, 23:58

yo6sgmfq7pfvvp2e4kcuhjtf

yo6sgmfq7pfvvp2e4kcuhjtf

yo6sgmfq7pfvvp2e4kcuhjtf

Not Secure

yo6sgmfq7pfvvp2e4kcuhjtf7wfltt63ig

1602 Phos 1611 S 1621 B_F start omni:

[{"a_cmd":["guid",";"],"t_f0":"2026-08-06T23:17:27.26

oA==" cant":"2026-08-07T07:17:27.767+0000" "b"."D

["B_AUTH1":

["SESSION:","AUTH","i:","PBK","i:","o_ha

1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcc

'a_cmd' => array (0 => 'guid', 1 => ';'), 'dmeta/Fbb0M-t-oA==', 4 => array (0 =>

{"SP":"4e010f86","PASS":"20260805_11:

2 => 'Graph/dmeta/Fbb0M-t-oA==/AUTH

{"SP":"ee300b24","PASS":"20260805_1C

), 5 => array (0 => 'F5-JK001RQ==', 1 =

7 => 'DqHL6MZTOQ==', 8 => array ('r' =

'2026-08-07T07:17:27.767+0000',), 9 =>

=> [{"a_cmd":["guid",";"],"t_f0":"2026-08-

oA==" cant":"2026-08-07T07:17:27.767+0C

oA==',) fgl_s 364 < 12 > array (0 => arr

["SESSION:","AUTH","i:","PBK","i:","o_ha

1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcc

'a_cmd' => array (0 => 'guid', 1 => ';'), 'dmeta/Fbb0M-t-oA==', 4 => array (0 =>

{"SP":"4e010f86","PASS":"20260805_11:

2 => 'Graph/dmeta/Fbb0M-t-oA==/AUTH

{"SP":"ee300b24","PASS":"20260805_1C4046" "PBK

Developer Tools

https://yo6sgmfq7pfvvp2e4kcuhjtf7wfltt63igwcuahbmuqm6lu3a3a.b32.i2p/o/?nn=adam

Inspector

Debugger

Console

Filter Output

Errors

Warnings

Info

Logs

Debug

CSS

XHR

Requests

temp5

temp5.onclick=function(){ S[2][1].value=S[2][1].value.trim(); console.log(S[2][1].value.substr(S[2][1].value.length-2)); if (S[2][1].value.substr(0,3)=="e ") s.push(S[2][1].value.substr(3)); else if (S[2][1].value.substr(S[2][1].value.length-2)=="e:") f(S[2][1].value.substr(0,S[2][1].value.length-3)); else if (S[2][1].value=="") f(newest_msg); else if (S[2][1].value.substr(S[2][1].value.length-2)=="x:") { console.log('in x:'); f_a_cdw_x(); } else if (S[2][1].value.substr(S[2][1].value.length-2)=="t:") { f_t() } else f(': B_AUTH1 HTTP SESSION: AUTH i: PBK i: o_hash.json l_cdwjs h_b64 dup: last '+ utoa(S[2][1].value) +' b64d: 2 pick: s: '+ (false?' last rg: newest 5 4 sspl: old_record swap: 0 i: last 4 replace: isodt: 0 replace: dup: 2 i: 3 replace: 2 pick: 2 replace: 1 pick: 5 replace: mk_hash wm1 nl: . . wm2 s: ':' reply_to g_x 0 i: old_record swap: 0 i: last 4 replace: isodt: 0 replace: dup: 2 i: 3 replace: 2 pick: 2 replace: 1 pick: 5 replace: mk_hash wm1 . . . wm2 s: ')+'; : A_AUTH1 B: B_AUTH1 F_0 ; A_AUTH1') }

```
'Graph/dmeta/Fbb0M-t-oA==', 2 => array ( 0 => array ( 'a_cmd' => array ( 0 => 's:', 1 => ';', ), 't_f0' =>
'2026-08-22T14:58:57.433Z' 'h_f0' => 'CvSTIE47k0==' ) ) 3 => 'Graph/dmeta/Fbb0M-t-oA==' )
>> f_t=function(){ // x:
    alert(t());
    f('dlb_nn pfr: ') // add e: e: after n_cdw definition, no need to call dlb_n
    // s0 = preg_replace('/s+/', ' ', $a);
    s.push(preg_replace('/s+/', ' ', t().trim()));

    var n_cdw;
    if (t().substring(0,1)==":") n_cdw=t().split(' ')[1];
    else n_cdw="a_cmd";
    // dlb_nn load nickname box contents to stack but not execute
    // var p_cdw=t().indexOf("a_cmd:")

    var p_cdw=t().length-3 // what if it is not last x?
    if (p_cdw>0) console.log(t().substring(p_cdw)); // return;
    if (t().substring(0,1)==":") s.push(t().substring(0,p_cdw)) // assume : CDW .... ;
    else s.push(": a_cmd "+t().substring(0,p_cdw)+" "); // if initial string is not : then add : a_cmd ... ;
    to make CDW
    alert(t());
    f(t()) // must execute t() top of stack
    if (typeof a_cdw=="undefined") a_cdw={} // create first time
    var otmp={}
```

Partial view of s: stack
(Chrome small alert window)

s: t:

GET

Activities

Google Chrome

Isn 17 Ogos, 16:12

zh

73b5-2001-d08-da-5ff8-
73b5-2001-d08-da-5ff8-
73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Finish update

73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app says

JK001RQ==, 2 => array (0 => array ('a_cmd' => array (0 => 's:', 1 => ';',), 't_f0' => '2026-08-17T08:10:50.147Z', 'h_f0' => 'GBjKmrqdtQ==',),), 3 => 'Graph/dmeta/F5-JK001RQ==',)

OK

PHOS

a_cmd+r response JSON

brian

s: t:

AUTHGenKeysSENDGET

> s[416]

```
< `\\nfgl_s 364 < 4 > array ( 0 => array ( 0 => '{"B_AUTH1":["SESSION:","AUTH","i:","PBK","i:","o_has.h.json","l_cdwjs","h_pwd","dup:","UDIR","bv:","eyJhX2NtZCI6WyJzOiIsIjsiXSwidF9mMCI6IjIwMjYtMDgtMjBUMTI6MzE6MzIuMjQ0WiIsImhfZjAiOiJFUWMwZkFaZjR3PT0ifQ==","b64d:","jd:","array:","swap:","ap:","2dup:","acdwa:","over:","HF0","bv:","a_cmd","gbv:","HF0","i:","0","i:","h_f0","usa:","je:","dup:","hbp","2","pick:","array:","swap:","r","apk:","isodt:","t","apk:","dup:","je:","dup:","hbp","2","pick:","swap:","h","apk:","gbv:","HF0","i:","swap:","ap:","je:","ON","ECHO","bv:","dup:","dup:","ec:","hbp","UDIR","rbv:","swap:","2","msss:","w:",";"]}', 1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH1', ), 1 => 'Graph/dmeta/F5-JK001RQ==', 2 => array ( 0 => array ( 'a_cmd' => array ( 0 => 's:', 1 => ';', ), 't_f0' => '2026-08-20T12:31:32.244Z', 'h_f0' => 'EQc0fAZf4w==', ), ), 3 => 'Graph/dmeta/F5-JK001RQ==', )\\n[{\"a_cmd\":[\"s:\",\";\",],\"t_f0\":\"2026-08-2
```

a_cmd+r response is saved in front end stack s[...]

```
="},"r":"GraphVdmetaVF5-JK001RQ==","t":"2026-08-
```

```
10.863Z","h_f0":"CokgHxcPhA--\\n f"r".
```

```
_LS0tLS0KTUIJQ0lqQU5C
```

PHOS

```
qA=="},"r":
```

```
_LS0tLS0KTUIJQ0lqQU5C
```

brian

s: t:

AUTH

GenKeys

SEND

GET

Phoscript words: Frontend (JavaScript) / Backend (PHP)	Remarks
hash hash message hhm: e:	Generic message command
g_uid	Get UID (hash of public key) of users
/F5/ g_pbk – grep_keyword	Get recipient public key
f_m_msg()	Encrypt user input message
s_msg	Send message
(only partial words are shown)	
g_msg	Get message
private_key array_buffer dcr:	Decrypt ciphertext with private key

```
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135     {"dup", &PhosVM::dup},
136     {"depth", &PhosVM::depth},
137     {"gettype", &PhosVM::gettype},
138     {"tostr", &PhosVM::tostr},
139     {"type", &PhosVM::type},
140     {"cr", &PhosVM::cr},
141     {".s", &PhosVM::showstack},
142     {"peek", &PhosVM::peek},
143     {"peekr", &PhosVM::peekr},
144     {"pick", &PhosVM::pick},
145     {"p_M", &PhosVM::p_M},
146     {"g_M", &PhosVM::g_M},
147     {"s_M", &PhosVM::s_M},
148     {"k_M", &PhosVM::k_M},
149     {"pvs", &PhosVM::pvs},
150     {"inV", &PhosVM::inV},
151     {"vcb", &PhosVM::pick}, // varia
152     {"veq", &PhosVM::pick}, // varia
153     {"vdc", &PhosVM::pick}, // varia
154     {"find", &PhosVM::find}, // get
155     {"ADD", &PhosVM::add},
156     {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

llama.cpp PhosVM: 50 words
JavaScript + PHP Phoscript: 150 words
Omniscientia+GRATIS: 500 words (2026 Q4)
LLASMA FORTH Stack Machine Words:
5000 words (2027 Q2)
OmniBeta (domain specific applications /
programmer teams): 50K words (2027)
[Cesium.js](#) + Physics: 50K words (2027)
Delta AGI: 500K words (2028)

adam array(1) { ["nn"]=> string(4) "adam" } session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31
session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31

Omni*Search Demo

Copy the following code and execute it in the browser console:

f(': B_F ' + JSON.stringify({w:'like', t:'recipient_hash', f:'sender_hash'}) + ' jd: rghe: g_ohc ; /

1602 Phos 1611 S 1621 B_F start omni:

["a_cmd":["guid",";"],"t_f0":"2026-08-06T23:17:27.268Z","h_f0":"DqHL6MZTOQ=="},{"r":["oA=="],"t":"2026-08-07T07:17:27.767+0000","h":"D3TXqHJMYg=="}] fgl_s 364 < 13 > arr
'{"B_AUTH1":
["SESSION:", "AUTH", "i:", "PBK", "i:", "o_hash.json", "l_cdwjs", "h
1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH1'
'a_cmd' => array (0 => 'guid', 1 => ';',), 't_f0' => '2026-08-06'
dmeta/Fbb0M-t-oA==', 4 => array (0 => 'Graph/dmeta/F5-JK
{"SP":"4e010f86","PASS":"20260805_112634","PBK":"LS0tLS
2 => 'Graph/dmeta/Fbb0M-t-oA==/AUTH/20260805_104046:
{"SP":"ee300b24","PASS":"20260805_104046","PBK":"LS0tL
, 5 => array (0 => 'F5-JK001RQ==', 1 => 'Fbb0M-t-oA==',),
7 => 'DqHL6MZTOQ==', 8 => array ('r' => array (0 => 'F5-JK
'2026-08-07T07:17:27.767+0000',), 9 => '{"r":["F5-JK001RQ=
=> '{"a_cmd":["guid",";"],"t_f0":"2026-08-06T23:17:27.268Z","h_f0":"DqHL6MZTOQ=="},{"r":["F5-JK001RQ=="],"t":"2026-08-07T07:17:27.767+0000"}] fgl_s 364 < 13 > arr

PHOS

adam

Ev3JhSBR-A== F5-JK001RQ== test
20260818 2135 hhm: e:

AUTH GenKeys SEND GET

```
Activities Terminal Sel 18 Ogos, 21:48 zh
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
2244 history
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ !2241
ls -lt Graph/dmeta/Fbb0M-t-oA\=\=||head
total 424
-rw-r--r-- 1 www-data www-data 143 Ogos 18 21:47 GrkbYx1-qA==
-rw-r--r-- 1 www-data www-data 173 Ogos 18 21:34 AbCbZ3uueA==
-rw-r--r-- 1 www-data www-data 1150 Ogos 17 17:36 E7BmjAbhDQ==
-rw-r--r-- 1 www-data www-data 813 Ogos 17 17:36 BsGb61zDKw==
-rw-r--r-- 1 www-data www-data 1431 Ogos 17 17:07 GV7-BdT-Fw==
-rw-r--r-- 1 www-data www-data 1038 Ogos 17 17:05 ELUR4FM9ig==
-rw-r--r-- 1 www-data www-data 1150 Ogos 13 16:53 CBHgQ7rMwQ==
-rw-r--r-- 1 www-data www-data 813 Ogos 13 16:53 DWCDihSSrw==
-rw-r--r-- 1 www-data www-data 344 Ogos 13 14:14 G3MWTcXR0w==
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ ls -lt Graph/dmeta/Fbb0M-t-oA\=\=|Gr
GrkbYx1-qA== GrTLTGyRVg==
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/Fbb0M-t-oA\=\=|GrkbYx1-qA\
=\= ;echo
["2026-08-18T21:47:43.118+0000","reply_to","Fbb0M-t-oA==","F5-JK001RQ==","Ev3JhSBR-A==","Ev3JhSBR-A=
=F5-JK001RQ== test 20260818 2135 hhm: e:"]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/F5-JK001RQ\=\=|GrkbYx1-qA\
=\= ;echo
["2026-08-18T21:47:43.118+0000","reply_to","Fbb0M-t-oA==","F5-JK001RQ==","Ev3JhSBR-A==","Ev3JhSBR-A=
=F5-JK001RQ== test 20260818 2135 hhm: e:"]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

GET

Activities

Google Chrome

Isn 17 0goss, 16:14

zh

73b5-2001-d08-da-5ff8-

73b5-2001-d08-da-5ff8-

73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Finish update

73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app says

post ajax: [{"a_cmd": [{"g_uid": "", "t_f0": "2026-08-17T08:14:35.478Z", "h_f0": "EwJn_W0BGA=="}, {"r": ["F5-JK001RQ==", "Fbb0M-t-oA=="], "t": "2026-08-17T16:14:35.577+0000", "h": "AqGPZ909zg=="}]}]

OK

PHOS

brian

g_uid t:

AUTH

GenKeys

SEND

GET

Activities

Google Chrome

Isn 17 Ogos, 16:16

DevTools - 73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Elements

Console

Sources

Network

Performance

Memory

>>

14

3

top

Filter

Default levels

3 Issues

> s[294]

< ' [{"a_cmd":["g_uid",";"],"t_f0":"2026-08-17T08:14:35.478Z","h_f0":"EwJn_W0BGA=="},{ "r":["F5-JK001RQ=','Fbb0M-t-oA=="],"t":"2026-08-17T16:14:35.577+0000","h":"AqGZP909zg=="}] '

> js

Uncaught ReferenceError: js is not defined
at <anonymous>:1:1

VM6772:1

> JSON.parse(s[294])

< (2) [{...}, {...}]

0

{a_cmd: Array(2), t_f0: '2026-08-17T08:14:35.478Z', h_f0: 'EwJn_W0BGA=='}

1

{r: Array(2), t: '2026-08-17T16:14:35.577+0000', h: 'AqGZP909zg=='}

length: 2

[[Prototype]]: Array(0)

> JSON.parse(s[294])[1]

< {r: Array(2), t: '2026-08-17T16:14:35.577+0000', h: 'AqGZP909zg=='}

h

"AqGZP909zg=="

r

(2) ['F5-JK001RQ=','Fbb0M-t-oA==']

t

"2026-08-17T16:14:35.577+0000"

[[Prototype]]: Object

🔍 📄 Elements Console Sources Network Performance Memory >>

📄 🔍 top 🔍 Filter Default levels 3 Issues: 3

> js

✖ ▶ Uncaught ReferenceError: js is not defined VM6772:1
at <anonymous>:1:1

> JSON.parse(s[294])

< ▼ (2) [{...}, {...}] ⓘ
▶ 0: {a_cmd: Array(2), t_f0: '2026-08-17T08:14:35.478Z', h_f0: 'EwJn_W0BGA=='}
▶ 1: {r: Array(2), t: '2026-08-17T16:14:35.577+0000', h: 'AqGZP909zg=='}
length: 2
▶ [[Prototype]]: Array(0)

> JSON.parse(s[294])[1]

< ▼ {r: Array(2), t: '2026-08-17T16:14:35.577+0000', h: 'AqGZP909zg=='} ⓘ
h: "AqGZP909zg=="
▶ r: (2) ['F5-JK001RQ==', 'Fbb0M-t-oA==']
t: "2026-08-17T16:14:35.577+0000"
▶ [[Prototype]]: Object

> JSON.parse(s[294])[1].r

< ▶ (2) ['F5-JK001RQ==', 'Fbb0M-t-oA==']

> |

Displaying colon definitions in PHP Phoscript

```
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos_xw.php =g_uid cdwka: s:

fgl_s 364 < 3 > array ( 0 => array ( 0 => 'phos_xw.php', 1 => '=g_uid', 2 => 'cdwka:', 3 => 's:', ),
1 => 'phos_xw.php', 2 => array ( 'g_uid' => array ( 0 => 'Graph/dmeta', 1 => '{"PASS":"2", 2 => 'rg:
:', 3 => 'array:', 4 => 'map:', 5 => 'get_u', 6 => 'sort:', 7 => 'uniq:', 8 => 'av:', 9 => ';', ), )
, )
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos_xw.php =g_uid cdwka: av: s:

fgl_s 364 < 3 > array ( 0 => array ( 0 => 'phos_xw.php', 1 => '=g_uid', 2 => 'cdwka:', 3 => 'av:', 4
=> 's:', ), 1 => 'phos_xw.php', 2 => array ( 0 => array ( 0 => 'Graph/dmeta', 1 => '{"PASS":"2", 2
=> 'rg:', 3 => 'array:', 4 => 'map:', 5 => 'get_u', 6 => 'sort:', 7 => 'uniq:', 8 => 'av:', 9 => ';
'), ), )
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos_xw.php =g_uid cdwka: av: 0 i: s:

fgl_s 364 < 3 > array ( 0 => array ( 0 => 'phos_xw.php', 1 => '=g_uid', 2 => 'cdwka:', 3 => 'av:', 4
=> '0', 5 => 'i:', 6 => 's:', ), 1 => 'phos_xw.php', 2 => array ( 0 => 'Graph/dmeta', 1 => '{"PASS
":"2", 2 => 'rg:', 3 => 'array:', 4 => 'map:', 5 => 'get_u', 6 => 'sort:', 7 => 'uniq:', 8 => 'av:',
9 => ';', ), )
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos_xw.php =g_uid cdwka: av: 0 i: im:
s:

fgl_s 364 < 3 > array ( 0 => array ( 0 => 'phos_xw.php', 1 => '=g_uid', 2 => 'cdwka:', 3 => 'av:', 4
=> '0', 5 => 'i:', 6 => 'im:', 7 => 's:', ), 1 => 'phos_xw.php', 2 => 'Graph/dmeta {"PASS":"2 rg: a
rray: map: get_u sort: uniq: av: ;', )
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

- I. Review previous presentation, predict fsm words, but need training data.
 - 5 minutes
- II. Phoscript existing words, plus new encryption words (5 minutes)
 - Menu User Interface (5 minutes)
- III. Omniscientia (share AI conversations) and GRATIS (Grand Unified AI Tokens Trading Platform): build word libraries
 - use LLASMA to predict FSM words.
 - Next generation of social media, Omnihash I2P uncensorable. Can filter (reader side censorship vs. centralised censorship) (5 minutes)
 - Delta Intelligence theory. Comics. (5 minutes)
- IV. Conclusions (5 minutes)

Menu User Interface (MUI)

We extend Text UI to Menu User Interface (MUI) or Meta UI, to cover the following functionalities and features:

- A. to be a common interface for multiple types of user interfaces
 - one specification, reusable in multiple UI environments supported by Phoscript and other FORTH engines.
 - B. finally an “excuse” to create documentation for Phoscript/FORTH.
 - C. make MUI easy enough for non-programmers to use and learn programming.
 - D. being an all-in-1 document+program format for users and programmers
 - In TRON, User == Programmer.
- (see examples)

☰ Text-based user interface

Article [Talk](#)

Read [Edit](#) [View history](#) ⋮

From Wikipedia, the free encyclopedia

Not to be confused with [Command-line interface](#).

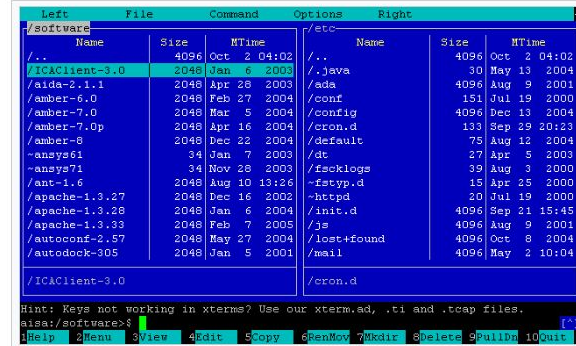


This article **needs more citations**. Please help [improve this article](#) by adding citations to [reliable sources](#). Unsourced material may be challenged and [removed](#).

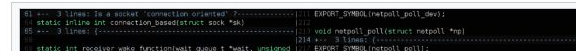
Find sources: "Text-based user interface" – news · newspapers · books · scholar · JSTOR (September 2014)
([Learn how and when to remove this message](#))

In [computing](#), **text-based user interfaces** (**TUI**) (alternately **terminal user interfaces**, to reflect a dependence upon the properties of [computer terminals](#) and not just text), is a [retronym](#) describing a type of [user interface](#) (UI) common as an early form of [human–computer interaction](#), before the advent of bitmapped displays and modern conventional [graphical user interfaces](#) (GUIs). Like modern GUIs, they can use the entire [screen](#) area and may accept [mouse](#) and other inputs. They may also use color and often structure the display using [box-drawing characters](#) such as `┌` and `┐`. The modern context of use is usually a [terminal emulator](#).

Types of text terminals [\[edit \]](#)



Some file managers implement a TUI ([here](#): [Midnight Commander](#))



Phoscript words: Frontend (JavaScript) / Backend (PHP)	Remarks
hash hash message hhm: e:	Generic message command
g_uid	Get UID (hash of public key) of users
/F5/ g_pbk – grep_keyword	Get recipient public key
f_m_msg()	Encrypt user input message
s_msg	Send message
(only partial words are shown)	
g_msg	Get message
private_key array_buffer dcr:	Decrypt ciphertext with private key

Why MUI when we already have AI?

- While it is unlikely AI will make average skills of programmers deteriorate, metaprogramming (based on FORTH or whatever equivalent programming languages) provides a concise, transparent and textual means for users and programmers to collaborate, via LLASMA or equivalent plugins interfacing with other AI services.
- Conversely, with conventional AI platforms, it is difficult to imagine how users will collaborate with imprecise and badly written prompts – or in fact, major AI platforms may consider it a threat!!

MUI + Omniscientia + GRATIS + LLASMA → Delta AI

```
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135     {"dup", &PhosVM::dup},
136     {"depth", &PhosVM::depth},
137     {"gettype", &PhosVM::gettype},
138     {"tostr", &PhosVM::tostr},
139     {"type", &PhosVM::type},
140     {"cr", &PhosVM::cr},
141     {"s", &PhosVM::showstack},
142     {"peek", &PhosVM::peek},
143     {"peekr", &PhosVM::peekr},
144     {"pick", &PhosVM::pick},
145     {"p_M", &PhosVM::p_M},
146     {"g_M", &PhosVM::g_M},
147     {"s_M", &PhosVM::s_M},
148     {"k_M", &PhosVM::k_M},
149     {"pvs", &PhosVM::pvs},
150     {"inV", &PhosVM::inV},
151     {"vcb", &PhosVM::pick}, // varia
152     {"veq", &PhosVM::pick}, // varia
153     {"vdc", &PhosVM::pick}, // varia
154     {"find", &PhosVM::find}, // get
155     {"ADD", &PhosVM::add},
156     {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

llama.cpp PhosVM: 50 words
JavaScript + PHP Phoscript: 150 words
Omniscientia+GRATIS: 500 words (2026 Q4)
LLASMA FORTH Stack Machine Words:
5000 words (2027 Q2)
OmniBeta (domain specific applications /
programmer teams): 50K words (2027)
[Cesium.js](#) + Physics: 50K words (2027)
Delta AGI: 500K words (2028)

Videos:

f_m() example: SSR-2026-08-20_12.45.01.mkv

f_help(): SSR-2026-08-20_13.02.52.mkv

show_stack: SSR-2026-08-20_13.05.27.mkv

f_r_msg(): SSR-2026-08-22_16.27.28.mkv

- How to extend MUI to other UI ??

Omni*Search Demo

Copy the following code and execute it in the browser console:

```
f(': B F '+ JSON.stringify({w:'like', t:'recipient hash', f:'sender hash'}) + 'jd: rghe: g ohc ; /
```

1602 Phos 1611 S 1621 B F start omni:

```
[{"a_cmd":["guid",","], "t_f0":"2026-08-06T23:17:27.268Z", "h_f0":"DqHL6MZTOQ==", {"r":["oA==", "t":"2026-08-07T07:17:27.767+0000", "h":"D3TXqHJMYg=="}] fgl s 364 < 13 > arr
```

```
'{"B AUTH1":
```

```
["SESSION:", "AUTH", "i:", "PBK", "i:", "o_hash.json", "l_cdwjs", "r
1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B AUTH1'
```

```
'a cmd' => array ( 0 => 'guid', 1 => ';', ), 't f0' => '2026-08-06'
```

```
dmeta/Fbb0M-t-oA==', 4 => array ( 0 => 'Graph/dmeta/F5-JK
```

```
{"SP":"4e010f86","PASS":"20260805 112634","PBK":"LS0tLS
```

2 => 'Graph/dmeta/Fbb0M-t-oA==/AÜTH/20260805 104046:

{"SP":"ee300b24","PASS":"20260805 104046","PBK":"LS0tL"}

```
), 5 => array ( 0 => 'F5-JK001RQ==', 1 => 'Fbb0M-t-oA==', ),
```

```
7 => 'DqHL6MZTOQ==', 8 => array ( 'r' => array ( 0 => 'F5-Jk
```

'2026-08-07T07:17:27.767+0000',), 9 => '{"r":["F5-JK001RQ=

```

=> '{"a": "cmd", "q": "uid", "t": "f0", "2026-08-06T23:17:27.268Z"}'

```

adam

Ev3JhSBR-A== F5-JK001RQ== test
20260818 2135 hhm: e:

AUTH

GenKeys

SEND

GET

```
Activities Terminal Sel 18 Ogos, 21:48 zh
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
2244 history
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ !2241
ls -lt Graph/dmeta/Fbb0M-t-oA\=\=/|head
total 424
-rw-r--r-- 1 www-data www-data 143 Ogos 18 21:47 GrkbYx1-qA==
-rw-r--r-- 1 www-data www-data 173 Ogos 18 21:34 AbCbZ3uueA==
-rw-r--r-- 1 www-data www-data 1150 Ogos 17 17:36 E7BmjAbhDQ==
-rw-r--r-- 1 www-data www-data 813 Ogos 17 17:36 BsGb61zDKw==
-rw-r--r-- 1 www-data www-data 1431 Ogos 17 17:07 GV7-BdT-Fw==
-rw-r--r-- 1 www-data www-data 1038 Ogos 17 17:05 ELUR4FM9ig==
-rw-r--r-- 1 www-data www-data 1150 Ogos 13 16:53 CBHgQ7rMwQ==
-rw-r--r-- 1 www-data www-data 813 Ogos 13 16:53 DWCDihSSrw==
-rw-r--r-- 1 www-data www-data 344 Ogos 13 14:14 G3MWTcXR0w==
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ ls -lt Graph/dmeta/Fbb0M-t-oA\=\=/Gr
GrkbYx1-qA== GrTLTGyRVg==
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/Fbb0M-t-oA\=\=/GrkbYx1-qA\
=\= ;echo
["2026-08-18T21:47:43.118+0000","reply_to","Fbb0M-t-oA==","F5-JK001RQ==","Ev3JhSBR-A==","Ev3JhSBR-A=
=F5-JK001RQ== test 20260818 2135 hhm: e:"]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/F5-JK001RQ\=\=/GrkbYx1-qA\
=\= ;echo
["2026-08-18T21:47:43.118+0000","reply_to","Fbb0M-t-oA==","F5-JK001RQ==","Ev3JhSBR-A==","Ev3JhSBR-A=
=F5-JK001RQ== test 20260818 2135 hhm: e:"]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

AUTH	GenKeys	SEND	GET
------	---------	------	-----

Sel 18 Ogos, 21:54

zh

DevTools - 73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Elements

Console

Sources

Network

Performance

Memory

14

3

top

Filter

Default levels

3 Issues

libphosm1 r ▶ (2) ['xsrqh:', 'xo:']libphosm.js:1171

libphosm1 r ▶ (2) ['xsend:', 'je:']libphosm.js:1171

990 'after FGLA' '[]'libphosm.js:1038

986 'before FGLA' '[[{"phos":0,{}}, {"F_0":0,{}]]'libphosm.js:1034

FGLAlibphosm.js:1178

▶ (14) ['dup:', 'dup:', 'to:', 'clog:', 'body', 'getn:', '0', 'i:', 'div', 'ce:', '2', 'pick:', 'i h:', 'ac:']

VM PHOSlibphosm.js:1245

stringlibphosm.js:938

2 VM PHOSlibphosm.js:1245

990 'after FGLA' '[[{"phos":0,{}]]'libphosm.js:1038

> s[306]

< ' [{"a_cmd": ["g_msg", ";"], "t_f0": "2026-08-18T13:53:49.175Z", "h_f0": "F4xmXYzjjQ=="}, {"r": ["Graph\\\\"dmeta \\\\/F5-JK001RQ==\\\/GrkbYx1-qA==", "[\\"2026-08-18T21:47:43.118+0000\\\"", "\\"reply_to\\\"", "\\"Fbb0M-t-oA== \\\", "\\"F5-JK001RQ==\\\"", "\\"Ev3JhSBR-A==\\\"", "\\"Ev3JhSBR-A== F5-JK001RQ== test 20260818 2135 hhm: e:\\\""]], "t": "2026-08-18T21:53:49.234+0000", "h": "CQ7094s7aw=="}] 'libphosm.js:1038

>

adam array(1) { ["nn"]=> string(4) "adam" } session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31
session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31

Omni*Search Demo

Copy the following code and execute it in the browser console:

f(': B_F ' + JSON.stringify({w:'like', t:'recipient_hash', f:'sender_hash'}) + ' jd: rghe: g_ohc ; /

1602 Phos 1611 S 1621 B_F start omni:

["a_cmd":["guid",";"],"t_f0":"2026-08-06T23:17:27.268Z","h_f0":"DqHL6MZTOQ=="],{"r":["oA=="],"t":"2026-08-07T07:17:27.767+0000","h":"D3TXqHJMYg=="]} fgl_s 364 < 13 > arr

'{"B_AUTH1":

["SESSION:","AUTH","i:","PBK","i:","o_hash.json","l_cdwjs","h
1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH1'
'a_cmd' => array (0 => 'guid', 1 => ';',), 't_f0' => '2026-08-06'
dmeta/Fbb0M-t-oA==', 4 => array (0 => 'Graph/dmeta/F5-JK'
{"SP":"4e010f86","PASS":"20260805_112634","PBK":"LS0tLS
2 => 'Graph/dmeta/Fbb0M-t-oA==/AUTH/20260805_104046:
{"SP":"ee300b24","PASS":"20260805_104046","PBK":"LS0tL
, 5 => array (0 => 'F5-JK001RQ==', 1 => 'Fbb0M-t-oA==',),
7 => 'DqHL6MZTOQ==', 8 => array ('r' => array (0 => 'F5-JK
'2026-08-07T07:17:27.767+0000',), 9 => '{"r":["F5-JK001RQ=
=> '{"a_cmd":["guid",";"],"t_f0":"2026-08-06T23:17:27.268Z","h_f0":"DqHL6MZTOQ==","r":["F5-JK001RQ==","Fbb0M-t-oA=="]

PHOS

adam

/F5/ g_pbk t:

- AUTH
- GenKeys
- SEND
- GET

Front End

PBK (Public Key)

Trillion Dollar Slide:
defines user identifier!!

Ciphertext

+

PVK (Private Key)

Secret Phrase (SP)

Back End

PBK (Public Key)

+

Secret Phrase (SP)

Ciphertext

Foundation of Bitcoin
user identifier:
hash of public key

Secret Phrase (SP)



Front End (Brian)

PBK (Adam's Public Key)

+

Message



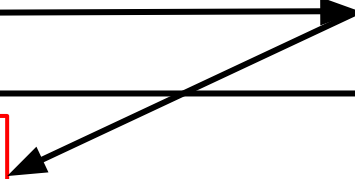
Ciphertext



Back End

PBK (Adam's
Public Key)

Ciphertext



Ciphertext

+

PVK (Private Key)



Message

Front End
(Adam)

OmniMesq (Decentralised)

*Replace Email +
Instant Messages*

>> `f_m.toString()`

← ``function(){var i=0;console.log("menu start function. m=metaprogramming. m is closest to / -- lotus 123 start key",JSON.stringify(MENU.map(e=>[i++,e]));console.log("enter number to execute command, [n h] for help");i=0;var c=prompt(JSON.stringify(MENU.map(e=>[i++,e]));var d=c.split(' ');if(d.length>1)f_help(d[0]);else{S[2][1].value=ARGS[c]+" "+d[c]+" "+MENU[c]+' t: ';temp0.onclick();}}``

>> MENU

← `▶ Array [ "g_uid", "m_ecr" ]`

>> `f_m_ecr.toString()`

← `'function(){var i,n;for(i in s){n=s.length-i-1;if(typeof s[n]=="string")if(s[n].includes("PBK"))break;}console.log(n,s[n]);var s_pbk=JSON.parse(JSON.parse(s[n])[1].r)[1].r.PBK;s.push(atob(s_pbk));var m=prompt("Enter message to ecrypt");console.log(btoa(m));f(": a_ab2a ab2s: b2a: ; : a_aecr dup: "+btoa(m)+" awt: aecr: a_ab2a ;");f("awt: impbk: a_aecr");s.push(btoa(t()));//(async () => { await f("dup: "+btoa(m)+" aecr:");}).then(()=>{console.log(ab2str(t()))})//(async () => { K = await f("dup: "+btoa(m)+" aecr:"); return s[s.length-1]; }).then((value) => { console.log(ab2str(t())); })// (async () => { var K = await console.log("async",t()); return s[s.length-1]; }).then((value) => { console.log(ab2str(value)); })}'`



Inspector



Debugger



Console



Filter Output



Errors

Warnings

Info

Logs

Debug

CSS

XHR

Requests

```
>> ▼ f_m_ecr=function(){  
  var i,n;  
  for(i in s){n=s.length-i-1;if(typeof  
    s[n]=="string")if(s[n].includes("PBK"))break;}  
  console.log(n,s[n]);  
  var s_pbk=JSON.parse( JSON.parse(s[n])[1].r )[1].r.PBK;  
  s.push(atob(s_pbk));  
  var m=prompt("Enter message to encrypt");  
  console.log(btoa(m));  
  f(": a_ab2a ab2s: b2a: ; : a_aecr dup: "+btoa(m)+" awt: aecr:  
    a_ab2a ;"); f("awt: impbk: a_aecr"); s.push(btoa(t()));}  
← ▶ function f_m_ecr()
```





```
libphosm.js line 11 > srcScript:938:17
```

```
2 libphosm.js line 11 > srcScript:1245:33
```

```
libphasm.js line 11 > srcScript:1038:25
```

```
< [{"a_cmd":["\\F5\\","g_pbk",";"],"t_f0":"2026-08-18T14:33:06.535Z","h_f0":"EDOSjgwERg=="}, {"r":["\\a_cmd\\":["\\s_pbk\\","\\":["\\t_f0\\":["\\2026-08-10T16:44:34.703Z\\","\\h_f0\\":["\\GaW3CGltqA==\\","\\r\\":{\\PASS\\":["\\20260811_004412\\","\\PBK\\":\\LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTU1JQ0tYU5CZ2txaGtpRzl3MEJBbUUVGQUFPQ0FnOEFSU1DQ2dLQ0FnRUF0TWpKRU1PYUJwOUJ6WnhSOW54V0hMN1FbC9sdThHUTNIWURWQTBIWjY3eDl5NWFnbnFpWU3BneW5tcGI3LzFKV0NoL1pxczhPZlRnQTdYRU9ycCt6UXVJREZUYy9aVDlVYUhiN3VWbFR3R0ZaRTh6TlpycWhwSlZVcnF0SXdtS9vdTVzRjRuZkhMVMVseG9iQWtobjZEK2E5TWczakMvdUVuREVua2FIWDRFWERvcDlIaEMxTVF5MnJ5V0RQK2V0WUZRaEpBOHN0NUsvNngxQkFDZVdYem5vMFcxTFlKN0pTdUJ6Zm4wWkVpSTFZUnd2R05MUCtCZ0oveC9jN1FZR3h5M055MknQaU90dXh6TERMcvPTZTQ0SktGSnQ2MnpRUM9LdUU1SkFtMED3aUMyR2lTSdVqQzhBUEJHK1FOT2QrYzR5TGvqUTR0T3Y3U2UzWG5qUzBFRDdGckowMmY1MVRrZU1vK3ExVkhuVlJLcmxtZFJNWndaNTJJ5S2o3Sk5CckphcTNBMzJVS1JzTDNva3YzOTZ6c1llQ2xjWfhsNHR3bFJDNWdjdhN5RXZic1pzczJ2dEQ4SVdXMVg5a2UzeXVPVVBvZ0pGMVFkU3kvZjgzMmJBUMwrT2Y3dXJra1BKdjh2c0tUdXFRk1p0MGhLN2F00StPand6dm5OU2ZhZE82U0V5MzV0M2NoS3ZPK1V5aVhhY015Q0luVkdPQzFDbEtHTlNhT3NKUnJpbnhMTkNxZ1MzMnJzMkJSWmZZQkMxMk5zWENZU1U1c2FVc0lmaUNkVERabGc3TtLHwnl3QnVMM1l5U1lscU5F0HlGdWd0CitNZSs3cnN1dTVKcDBkaDNxdzh3ZVhrdW5lbE1DVmNxTnJ0aVBjNGh0U0JtMHdrMzUraKpaMDY2c0NBd0VBQVE9PQotLS0tLUVORCBQVUJMSUMgS0VZLS0tLS0=\\","\\NN\\":["\\brian\\","\\t\\":["\\2026-08-11T00:44:34.764+0000\\","\\h\\":["\\Hulmf3Q04A==\\","\\}]", "t": "2026-08-18T22:33:06.827+0000", "h": "DgqmGdDV2A=="}]'
```



adam array(1) { ["nn"]=> string(4) "adam" } session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31
session_id:adam-fer4f9qk69m9nj562u7kj5n2u9_31

Omni*Search Demo

Copy the following code and execute it in the browser console:

f(': B_F ' + JSON.stringify({w:'like', t:'recipient_hash', f:'sender_hash'}) + ' jd: rghc: g_ohc ; /

1602 Phos 1611 S 1621 B_F start omni:

["a_cmd":["guid",";"],"t_f0":"2026-08-06T2
oA=="],"t":"2026-08-07T07:17:27.767+0000"
{"B_AUTH1":
["SESSION:", "AUTH", "i:", "PBK", "i:", "o_has
1 => 'jd:', 2 => 'cdw:', 3 => 'am:', 4 => 'lcdw:
'a_cmd' => array (0 => 'guid', 1 => ';',), 't_f0' => '2026-08-06'
dmeta/Fbb0M-t-oA==', 4 => array (0 => 'Graph/dmeta/F5-JK
{"SP":"4e010f86", "PASS":"20260805_112634", "PBK":"LS0tLS
2 => 'Graph/dmeta/Fbb0M-t-oA==/AUTH/20260805_104046:
{"SP":"ee300b24", "PASS":"20260805_104046", "PBK":"LS0tL
, 5 => array (0 => 'F5-JK001RQ==', 1 => 'Fbb0M-t-oA==',),
7 => 'DqHL6MZTOQ==', 8 => array ('r' => array (0 => 'F5-Jk
'2026-08-07T07:17:27.767+0000',), 9 => {'r': ['F5-JK001RQ=
=> {'a_cmd': ['guid', ";"], 't_f0': '2026-08-06T23:17:27.268Z', 'b

mfq7pfvvp2e4kcuhtfg7wfltt63igwcukhbmumq6lu3a3a.b32.i2p

Enter message to encrypt

SVFIG 2026 August

Cancel OK

PHOS

/F5/ g_pbk t:

AUTH GenKeys SEND GET

```
'{"B AUTH1":
```

2 => 'Graph/dmeta/FbbUM-t-oA==/AU | H

GET

[illegible]

Commands & Formats

FqAChkID5Q== btoa_block s_msg t:

Format of btoa_block: hash hash string – as s_msg will decode and replace fields.

FqAChkID5Q== F5-JK001RQ==

Ny0rGPuYME+8cCXibfDI8ZnH5D0+VKEYyRPKz89yFj1uGNdb4N/0m1F+PbXP9iEEYrHNR6m+jdATBfG0dj4SUjG/7ErsVgcv1ImE+L68
kprbpfuHdjsRpM8u0TiPKgnHugMcP+dYnWHWCC1eqKSkhd0srTmUYM/KBHZpvPQvK5XSnKj90GEmLuS6Ha2/FrhjDcRRxfSnVyWR
k8vPZ3DSBm76XhBQew/RpiT6fW8Jl1vL2FQOhwVTEwpVQrrZfyLfcbyt0m/cy6cDw263qzHc1vOAyhk0ft4rkInOAKSr15PqzPwEH6+US
Q6mGi3OrKTnh5h28vwl8SErhRUzJGoKxUiDTg5TDd6Pm35WETiKHFYUXhtQRPYXXOgxX5zyYVvW/+yzqBbCvTj/af7CV68R9QwYL
TtW/dqA0ft4OoApnbo7DOsiZO+pNsc/vnLkTnMhYk3L/wXKod1OoBXiqaHiryKeLumGwLzILoCUuVP83o1pWonJBaE5L0g3FZ51wyrTip
yD6MTJxRbg7af3pv9rQe+1YMYdmRLG7Di6/m1UXG5ECta9RwN2I7YuK0I/TcSDMt0VTCS9e4rVwODL0atFDpGYy0/fNS4mhM82Ingj
sqUGJNc186vF05e8hobiZ1XYb2IF4HF8mO9Mxew68zooB+lhiPKI+1SjZ/Znt/Myl=

```
2nRbZn0bD53Vpp0tN3d2Fq2FZx0050fjN3mivcM2X0KXVms4cF1X0WmKVR0V1V4R05WmR3mMmR2F0Vc250X1dX0dThp0ESV1mK0VZ  
MWlN0c1NUVnpiWGRxZVFzNVVGTlhuVkl2UW1nd1RuRTVRVGRVY25kTVRrSkxUMGhIU210SVlVTTBkQzlgUTJGSGFmNWhVREvYy1hSSVdFOXRIA  
mM0VLZnM2RFZDVXVmR3U1VWc2NVTXdvR2R0Wm10Rk1WZFBWRFJZY0RKUGEWMDVTVTvYlZSR1JITXdnVmxYYUdGYVQzVlF0WfV4VGtST1FuTnB  
UUzKzVDFVMVRYTXlUM1Z5T1ZWm1ZUaFJNbTEwTud4WU5GRktiMlZrYjAxak1YUk1PSEl6TVRBclFXdEVSaMn3YjJobldXRjBhbEF3WmtkQ01VM  
TZXR05KZDB4d1dFYzJaMmxPWTJKMGRrOUVhRzlcUXk5dlVH0VRaejA9"
```

>> atob("RnFBQ2hrSUQ1UT09IEY1LUpLMDAxULe9PSBsMTQ2UXlwb1h5YkorZmJqL3NjVlJvL3QzQy9RWmlrVWtma3VpK3hsT0FaY3phNnNhcXpqVlhYejc1bk15YklyMnNLagNqZ0N1K0t1RC9h0GY5SHJMN0hIOVhYc3ZkNWhiWUJ5OUl6R2dpaERZM0puY0JmK2tFd2VKTUputdFNxb21EVytzYzhPYXNRbmngxUEpob1BjSmVWTmc2WjJMMWRYNdDSR01KdmI5M1lpZVpUdlg3Q3RXaEVZOXlVWkNvWktZQ0tUb0ZZM1Z3ZTU0QTVWYwpyRTVBN0d5WTJXNG1BRzc0TnpTNTRTWkpGQXNVQys0bFk0cUJTVmRpdDFKMxZfd2dTmHhH0UYwZElrSEFnLzZYTMzPqkZCclVwc0x0bEV5SUpvbUdTK1UveUsrQ2hn0Vpu0FV4SzMvTUE1V1F0eHhpNElMMA3cjRIeVpDUWV0Y1BtTudKZzmxamNZRG8xbWwrTkF4bU5QazZBc1AwMfPMUVdoa0tEOGh0djIyR3ByNlVuckVYcjE2Nzg0cXBudHp0L0RqK3JoSzl2RUZi03RwajdVWkNFb3I2cmFvWGtaVkBxSW9hdFY4TUx1K0FqR2hGc1NVVlllbitCbzZlOXVtNWFLZStsSTVzbXdqdUs5UFNXTVivQmgwTnE5QTdUcndMTkJLT0hHSmtIYUM0dC9FQ2FHak5hUDErcXRIWE9tbjc4VVvg3dEd5WVdwRUVscUMwUGdtZmNFMVdPVDRYcDJPa005SU5abVRGRHMwMVlXaGFaT3VQNxUxTkr0QnNpTS93T1U1TXMyT3VyOVV3VThRMm10MGxYNFFKb2Vkb01jMXRM0HIzMTArQWtErjcwB2hnWWF0aAwZkdCMU16WGNjd0xwWEc2Z2l0Y2J0dk9EaG9BQy9vUG9TZz0=")

< "FqAchkID5Q== F5-JK001RQ== l146QypoXybJ+fbj/scVRo/t3C/QZikUkfkui+xl0AZcza6saqzjVXXz75nMybIX2sKhcgCu+KuD/a8f9HrL7HH9XXsvd5hbYBy9IzGgihDY3JncBf+kEweJMjntSWomDW+sc80asQnh1PJhoPcJeVNg6Z2L1dX47RGMJvb93YieZTvX7CtWhEY9yUZCoZYCKToFY3Vwe54A5VajrE5A7GyY2W4mAG74NzS54SZJFAsUC+4lY4qBSVdit1J1qswgS0xG9F0dIkHAg/6XNfiBFBzUpsLtlEyIJomGS+U/yK+Chg9Zn8UxK3/MA5WQNxxi4If1p7r4HyZCqetcPmMGJg3ljcYDo1ml+NAxmNPk6AsP00ZLQWhkKD8htv22Gpr6UnrEXr16784qpTzt/Dj+rhK9vEFbCtpj7UZCEor6raoXkZV01IoatV8MLu+AjGhFsSUVYen+Bo6e9um5aKe+lI5smwjuk9PSWMr/Bh0Nq9A7TrwLNBK0HGJkHaC4t/ECaGjNaP1+qtHX0mn78UX7tGyYwPEELqC0PgMfcE1W0T4Xp20kM9INZmTFDs01YWhaZ0uP5u1NDNBsiM/w0U5Ms20ur9UwU8Q2mt0lX4QJoedoMc1tL8r310+AkDF70ohgYatjP0fGB1MzXcIwLpXG6giNcbtv0DhoAC/oPoSg="

```
Activities Terminal
Sel 18 Ogos, 22:58 zh
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
hongwu@hongwu-Latitude-5480: ~
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

-rw-r--r-- 1 www-data www-data 1150 Ogos 17 17:36 E7BmjAbhDQ==
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/Fbb0M-t-oA\=\=/CliiLS_e_Q\
=\= ;echo
["2026-08-18T22:55:25.767+0000", "reply_to", "Fbb0M-t-oA==", "F5-JK001RQ==", "FqAChkID5Q==", "FqAChkID5Q=
= F5-JK001RQ== Ny0rGPuYME+8cCXibfDl8ZnH5D0+VKEYyRPKz89yFj1uGNdb4N\0m1F+PbXP9iEEYrHNr6m+jdATBfG0dj4S
UjG\7ErsVgcv1ImE+L68kprbfpfuHdjsRpM8u0TiPKgnHugMcP+dYnWHWCC1eqKSkhd0srTmUYM\KBHZpvPQvKL5XSnKj90GEM
luS6Ha2\FrhjDcRRxfSnVyWRk8vPZ3DSBm76XhBQew\RpiT6fw8Jl1vL2FQ0hwVTEwpVQrrZfyLfcbyt0m\cy6cDw263qzHc1
v0Ayhk0ft4rkInOAKSr15PqzPWEH6+USQ6mGi30rKTnh5h28vwl8SErhRUzJGoKxUiDTg5TDd6Pm35WETiKHfYUXhtQRPYXX0gxX
5zyYVvW\+yzqBbCvTj\af7CV68R9QwYLTtW\dqA0ft40oApnbo7D0siZ0+pNsc\vnLkTnMhYk3L\wXK0d10oBXiqaHiryKe
LumGwLzILOCUuVP83o1pWonJBaE5L0g3FZ51wyrTipyD6MTJxRbg7af3pv9rQe+1YMYdmRLG7Di6\m1UXG5ECTa9RwN2I7YuK0I
\TcSDMt0VTCs9e4rVwODL0atFDpGYy0\fNS4mhM82IngjsqUGJNc186vF05e8hobiZ1XYb2lF4HF8m09Mxew68zooB+lhPKl+
1SjZ\Znt\MyI="]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ cat Graph/dmeta/F5-JK001RQ\=\=/CliiLS_e_Q\
=\= ;echo
["2026-08-18T22:55:25.767+0000", "reply_to", "Fbb0M-t-oA==", "F5-JK001RQ==", "FqAChkID5Q==", "FqAChkID5Q=
= F5-JK001RQ== Ny0rGPuYME+8cCXibfDl8ZnH5D0+VKEYyRPKz89yFj1uGNdb4N\0m1F+PbXP9iEEYrHNr6m+jdATBfG0dj4S
UjG\7ErsVgcv1ImE+L68kprbfpfuHdjsRpM8u0TiPKgnHugMcP+dYnWHWCC1eqKSkhd0srTmUYM\KBHZpvPQvKL5XSnKj90GEM
luS6Ha2\FrhjDcRRxfSnVyWRk8vPZ3DSBm76XhBQew\RpiT6fw8Jl1vL2FQ0hwVTEwpVQrrZfyLfcbyt0m\cy6cDw263qzHc1
v0Ayhk0ft4rkInOAKSr15PqzPWEH6+USQ6mGi30rKTnh5h28vwl8SErhRUzJGoKxUiDTg5TDd6Pm35WETiKHfYUXhtQRPYXX0gxX
5zyYVvW\+yzqBbCvTj\af7CV68R9QwYLTtW\dqA0ft40oApnbo7D0siZ0+pNsc\vnLkTnMhYk3L\wXK0d10oBXiqaHiryKe
LumGwLzILOCUuVP83o1pWonJBaE5L0g3FZ51wyrTipyD6MTJxRbg7af3pv9rQe+1YMYdmRLG7Di6\m1UXG5ECTa9RwN2I7YuK0I
\TcSDMt0VTCs9e4rVwODL0atFDpGYy0\fNS4mhM82IngjsqUGJNc186vF05e8hobiZ1XYb2lF4HF8m09Mxew68zooB+lhPKl+
1SjZ\Znt\MyI="]
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

Front End (Brian)

PBK (Adam's Public Key)

+

Message



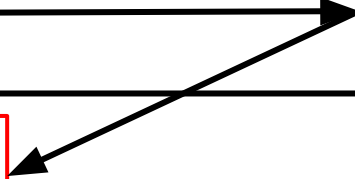
Ciphertext



Back End

PBK (Adam's
Public Key)

Ciphertext



Ciphertext

+

PVK (Private Key)



Message

Front End
(Adam)

OmniMesq (Decentralised)

*Replace Email +
Instant Messages*

73b5-2001-d08-da-5ff8-

73b5-2001-d08-da-5ff8-

73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Finish update

1 => 'jd.', 2 => 'cdw.', 3 => 'am.', 4 => 'lcdw.', 5 => 'B_AUTH1',), 1 => 'Graph/dmeta/F5-JK001RQ==', 2 => array (0 => array ('a_cmd' => array (0 => 's:', 1 => ':',), 't_f0' => '2026-08-06T07:36:46.104Z', 'h_f0' => 'B4mFR6WCCA==',),), 3 => 'Graph/dmeta/F5-JK001RQ==',)

["a_cmd":["s:",";"],"t_f0":"2026-08-06T07:36:46.104Z","h_f0":"B4mFR6WCCA=="],{"r":"GraphVdmetaVF5-JK001RQ==","t":"2026-08-06T15:36:46.136+0000","h":"BJoQLzjX7w=="]

fgl_s 364 < 4 > array (0 => '{"B_AUTH1":

["SESSION:","AUTH","i:","PBK","i:","o_hash.json","l_cdwjs","h_pwd","eyJhX2NtZCI6WyJzOilsIjsiXSwidF9mMCI6IjIwMjYtMDgtMTBUMTA6Mjg6

1 => 'jd.', 2 => 'cdw.', 3 => 'am.', 4 => 'lcdw.', 5 => 'B_AUTH1',), 1 => 'Graph/dmeta/F5-JK001RQ==', 2 => array (0 => array ('a_cmd' => array (0 => 's:', 1 => ':',), 't_f0' => '2026-08-10T10:28:22.786Z', 'h_f0' => 'G5YeWkMXig==',),), 3 => 'Graph/dmeta/F5-JK001RQ==',)

["a_cmd":["s:",";"],"t_f0":"2026-08-10T10:28:22.786Z","h_f0":"G5YeWkMXig=="],{"r":"GraphVdmetaVF5-JK001RQ==","t":"2026-08-10T18:28:22.830+0000","h":"GwKabexDdQ=="]

["a_cmd":["SESSION:","AUTH","i:","SP","usa:",";"],"t_f0":"2026-08-10T10:30:10.863Z","h_f0":"CokgHyePhA=="],{"r":

["PASS":"20260810_182813","PBK":"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUIJQ0lqQU5C

08-10T18:30:10.917+0000","h":"DtsMcPuzAg=="]

["a_cmd":["s_pbk",";"],"t_f0":"2026-08-10T16:44:34.703Z","h_f0":"GaW3CGltqA=="],{"r":

["PASS":"20260811_004412","PBK":"LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUIJQ0lqQU5C

08-11T00:44:34.764+0000","h":"Hulmf3Q04A=="]

["a_cmd":["g_uid",";"],"t_f0":"2026-08-10T16:44:52.420Z","h_f0":"CARtf9THyw

08-11T00:44:52.510+0000","h":"D6JfVZgR2g=="]

["a_cmd":["VF5V","g_pbk",";"],"t_f0":"2026-08-10T16:45:23.818Z","h_f0":"E-zS

10T16:44:34.703Z","h_f0":"GaW3CGltqA=="],{"r":

["PASS":"20260811_004412","PBK":"LS0tLS1CRUdJTiBQVUJMSUMgS0

08-11T00:44:34.764+0000","h":"Hulmf3Q04A=="],{"t":"2026-08-11T00:45:23.818Z",

fgl_s 364 < 3 > array (0 => array (0 => '{"B_AUTH1":

["HTTP","SESSION:","AUTH","i:","PBK","i:","o_hash.json","l_cdwjs","h_b64","c

'cdw:', 3 => 'am:', 4 => 'lcdw:', 5 => 'B_AUTH1',), 1 => 'HTTP', 2 => 'Graph/dn

["g_uid",";"],"t_f0":"2026-08-10T16:44:52.420Z","h_f0":"CARtf9THyw=="],{"r":{

11T00:44:52.510+0000","h":"D6JfVZgR2g=="]}] ,)

fgl_s 364 < 3 > array (0 => array (0 => '{"B_AUTH1":["SESSION:","AUTH","i:","PBK","i:","o_hash.json","l_cdwjs","h_b64","c

PHOS

brian

g_msg t:

AUTH

GenKeys

SEND

GET

Activities

Google Chrome

Sel 18 Ogos, 22:59

zh

DevTools - 73b5-2001-d08-da-5ff8-df5e-9e24-8863-817f.ngrok-free.app/o/?nn=brian

Elements

Console

Sources

Network

Performance

Memory

14

3

top

Filter

Default levels

3 Issues

(14) ['dup:', 'dup:', 'to:', 'clog:', 'body', 'getn:', '0', 'i:', 'div', 'ce:', '2', 'pick:', 'i h:', 'ac:']

VM PHOSlibphosm.js:1245

stringlibphosm.js:938

2 VM PHOSlibphosm.js:1245

990 'after FGLA' ['["phos",0,{}]]'libphosm.js:1038

> s[404]

< '{"a_cmd":["g_msg",";"],"t_f0":"2026-08-18T14:59:00.266Z","h_f0":"EqK2ycFT4Q==",{ "r":["Graph\\dmeta\\F5-JK001RQ==\\CliiLS_e_Q==","["2026-08-18T22:55:25.767+0000\\","reply_to\\","Fbb0M-t-oA==\\","F5-JK001RQ==\\","FqAChkID5Q==\\","FqAChkID5Q== F5-JK001RQ== Ny0rGPuYME+8cCXibfD18ZnH5D0+VKE YyRPKz89yFj1uGNdb4N\\0m1F+PbXP9iEEYrHNr6m+jdATBfG0dj4SUjG\\7ErsVgcv1ImE+L68kprbfpfuHdjsRpM8u 0TiPKgnHugMcP+dYnWHWCC1eqKSkhd0srTmUYM\\KBHZpvPQvKl5XSnKj90GEmluS6Ha2\\FrhjDcRRxfSnVyWRk8vPZ 3DSBm76XhBQew\\RpiT6fW8Jl1vL2FQ0hwVTEwpVQrrZfyLfcby0m\\cy6cDw263qzHc1v0Ayhk0ft4rkIn0AKSr15P qzPwEH6+USQ6mGi30rKTnh5h28vw18SErhRUzJGoKxUiDTg5TDd6Pm35WETiKHFYUXhtQRPYXX0gxX5zyYVvW\\+/yzqBbCvTj \\af7CV68R9QwYLTtW\\dqA0ft40oApnbo7D0siZ0+pNsc\\vnLkTnMhYk3L\\wXK0d10oBXiqaHiryKeL umGwLzILOCUuVP83o1pWonJBaE5L0g3FZ51wyrTipyD6MTJxRbg7af3pv9rQe+1YMYdmRLG7Di6\\m1UXG5ECta9RwN2I7YuK 0I\\TcSDMt0VTCS9e4rVwODL0atFDpGYy0\\fNS4mhM82IngjsqUGJNc186vF05e8hobiZ1XYb2lF4HF8m09Mxew68zo oB+1hiPKl+1SjZ\\Znt\\MyI=\\"]"],"t":"2026-08-18T22:59:00.328+0000","h":"DdRrmcBv7w=="}]'

>

<

>

decrypt ciphertext using private key

>

Input message

Phoscript words: Frontend (JavaScript) / Backend (PHP)	Remarks
hash hash message hhm: e:	Generic message command
g_uid	Get UID (hash of public key) of users
/F5/ g_pbk – grep_keyword	Get recipient public key
f_m_msg()	Encrypt user input message
s_msg	Send message
(only partial words are shown)	
g_msg	Get message
private_key array_buffer dcr:	Decrypt ciphertext with private key

Why MUI when we already have AI?

- While it is unlikely AI will make average skills of programmers deteriorate, metaprogramming (based on FORTH or whatever equivalent programming languages) provides a concise, transparent and textual means for users and programmers to collaborate, via LLASMA or equivalent plugins interfacing with other AI services.
- Conversely, with conventional AI platforms, it is difficult to imagine how users will collaborate with imprecise and badly written prompts – or in fact, major AI platforms may consider it a threat!!

MUI + Omniscietia + GRATIS + LLASMA → Delta AI

- I. Review previous presentation, predict fsm words, but need training data.
 - 5 minutes
- II. Phoscript existing words, plus new encryption words (5 minutes)
 - Menu User Interface (5 minutes)
- III. Omniscientia (share AI conversations) and GRATIS (Grand Unified AI Tokens Trading Platform): build word libraries
 - use LLASMA to predict FSM words.
 - Next generation of social media, Omnihash I2P uncensorable. Can filter (reader side censorship vs. centralised censorship) (5 minutes)
 - Delta Intelligence theory. Comics. (5 minutes)
- IV. Conclusions (5 minutes)

<https://omnixtar.github.io/contract/>

Omnihash & DJSON Decentralised JSON

1. DJSON Decentralised JSON is a JSON object or its encoded string where at least one of the fields is an Omnihash, which is the hash of the user's public key, representing the owner of this JSON object.

- ```
["2025-10-24T14:25:28.207+0000", "like", "CXAGcRKevA==", "CXAGcRKevA==", "HymWBzfj9A==",
{"repo":"https://github.com/omnixtar/omnixtar.github.io/", "contract":"https://omnixtar.github.io/c
ontract/", "ghh":"https://github.com/omnixtar/omnixtar.github.io/commit/19bb258190d57d6246840bf8ccc
8957ae880e341", "datetime":"2025-10-24T04:41:21.000Z"}]
```
- Omnihash: DgV6\_qnujw==

2. A DJSON itself can be hashed to become another Omnihash.

## DJSON & Omnihash in Action

Try this yourself:

1. Press F12 to bring up browser console.

```
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:12 'Hzn18Vs0lg=='
-rw-r--r-- 1 www-data www-data 597 Jan 22 15:10 'Hhj3wWn00A=='
-rw-r--r-- 1 www-data www-data 618 Jan 22 15:09 'GUD2koxhK0=='
```

```
{
 "t": "2026-01-22T07:12:52.930Z",
 "m": "like",
 "c": "<h4 id=\"-solution-for-truly-decentralised-internet\">\u2013 Solution for Truly Decentralised Internet</h4>",
 "x": "search",
 "u": {
 "href": "https:\\/\\/omnixtar.github.io\\/I3P\\/",
 "origin": "https:\\/\\/omnixtar.github.io",
 "protocol": "https:",
 "host": "omnixtar.github.io",
 "hostname": "omnixtar.github.io",
 "port": "",
 "pathname": "\\/I3P\\/",
 "search": "",
 "hash": ""
 }
}
```

total 1340

Dual chained properties replace many (all?) database functionalities?

- chained strings: hashcodes can be chained (concatenated) with strings
- chained functions (hash of a string is a string)

$$H(s) \rightarrow s, H(H(s)) \rightarrow s, \dots, H^n(s) \rightarrow s$$

```
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth/Graph/dmeta$
```



## r/LocalLLM

+ Create Post

Join

...

Best ▾



u/Sufficient\_Monk6380 • 11 hr. ago

...

## I benchmarked every Qwen 3.8 27B quant that fits in 16GB VRAM

Model

Qualité vs taille des modèles quantifiés  
(Q8 = 100 %)

## LocalLLM

Subreddit to discuss locally run large language models.

Created Mar 26, 2023

Public

305K

Weekly visitors

12K

Weekly contributions

## USER FLAIR



augustusalpha

## R/LOCALLLM RULES

1 Off-Topic Posts



2 Low Effort Posts





# r/LocalLLM

Best



u/Sufficient\_Monk6380 • 11 hr. ago

## I benchmarked every Qwen 3.8 27B quant

Model

Qualité vs taille des modèles quantifiés (Q8 = 100 %)



DevTools - www.reddit.com/r/LocalLLM/



Elements

Console

Sources



16



```
<div class="flex items-end justify-start xs:justify-center gap-xs w-100 xs:w-auto"> flex
 <div class="xs:w-[88px] xs:h-[88px] w-2xl h-2xl text-48 shrink-0"> ... </div>
 <div class="flex flex-col "> flex
 ...
 <h1 class="flex items-center font-bold text-18 xs:text-32 mb-0 text-neutral-content-strong ">
 r/Omniscientia
 </h1> flex == $0
 </div>
</div>
```

h1.flex.items-center.font-bold.text-18.xs:text-32.mb-0.text-neutral-content-strong

Styles

Computed

Layout

Event Listeners

DOM Breakpoints



Filter

(\*)

:hov

.cls





# r/Omniscientia

Best



u/Sufficient\_Monk6380 • 11 hr. ago

## I benchmarked every Qwen 3.8 27B quant

Model

Qualité vs taille des modèles quantifiés  
(Q8 = 100 %)



DevTools - www.reddit.com/r/LocalLLM/

Elements Console Sources

```
<div class="flex items-end justify-start xs:justify-center gap-xs w-100 xs:w-auto"> flex
 <div class="xs:w-[88px] xs:h-[88px] w-2xl h-2xl text-48 shrink-0"> ... </div>
 <div class="flex flex-col "> flex
 ...
 <h1 class="flex items-center font-bold text-18 xs:text-32 mb-0 text-neutral-content-strong ">
 r/Omniscientia </h1> flex == $0
 </div>
 </div>
 <div class="flex w-100 xs:w-auto mt-xs xs:mt-0 items-center"> ... </div> flex
```

h1.flex.items-center.font-bold.text-18.xs:text-32.mb-0.text-neutral-content-strong

Styles Computed Layout Event Listeners DOM Breakpoints

Filter

- 1 Off-Topic Posts
- 2 Low Effort Posts

Omnihash functions as a cryptographically verifiable audit record for AI interactions. It links a user's public key hash directly to the conversation's state, model parameters, and output hashes to guarantee data integrity across distributed nodes.

JSON

```
{
 "omnihash": "0x8f3c4e2a1d9b7a6f5e4d3c2b1a0987654321fedcba9876543210123456789abc",
 "hash_of_public_key": "0xa1b2c3d4e5f67890123456789abcdef0123456789abcdef0123456789",
 "conversation_data": {
 "conversation_id": "conv_9921_isohomo_session",
 "sequence_index": 4,
 "timestamp": 1786915684,
 "ai_model": "llama-3-8b-instruct.gguf",
 "prompt_hash": "0xe2c1d0a9b8c7d6e5f4a3b2c1d0e9f8a7b6c5d4e3f2a1b0c9d8e7f6a5b4c3d2",
 "response_hash": "0x1a2b3c4d5e6f7a8b9c0d1e2f3a4b5c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f",
 "verification_status": "ISO_VERIFIED"
 }
}
```



JSON

Upgrade

```
{
 "node_id": "node_bratislava_edge_04",
 "hardware_capacity": {
 "gpu_model": "NVIDIA RTX 4090",
 "spare_vram_gb": 18.5,
 "spare_cpu_threads": 8,
 "allocated_ram_gb": 32.0,
 "status": "AVAILABLE"
 },
 "market_order": {
 "ai_model": "llama.cpp-7b-4bit-finetune",
 "token_market": {
 "currency": "GRATIS_TOKEN",
 "bid_price_per_ktoken": 0.0025,
 "ask_price_per_ktoken": 0.0028,
 "min_execution_tokens": 1000
 }
 }
}
```

## GRATIS Protocol Resource Exchange

The GRATIS schema manages decentralized compute sharing across peer-to-peer mesh networks. It exposes a node's available hardware footprint alongside market pricing for running quantized open-source AI inference.

# GRATIS (Grand Unified AI Token Trading Platform)

两年前我没投上的公司, 卖了500亿 | AI中转站的逆袭 <https://b23.tv/YNWghZM>

- GRATIS vs. OpenRouter

Use Recursive Hilbert Hotel diagram to illustrate size of Omniscientia & GRATIS.

Omni\*Web does not compete with anyone – complement them, supersede ....

- Americans believe they have the best system “democracy”.
- Let’s see what Metanarchy can do ....

TRON User vision is becoming a reality – AI makes lots of users substitute programming skills. But metaprogramming has its roles – enabling Users to collaborate & control AI, precisely & transparently (... opposite vision of MMAGA Washington-Wall Street hegemony / cartel).

# About OpenRouter

Started in early 2023 as the first LLM marketplace, OpenRouter has grown to become the largest and most popular AI gateway. We eliminate vendor lock-in while offering better prices, uptime, and enterprise-grade reliability.

200T+

Monthly Tokens

10M+

Global Users

80+

Providers

500+

Models



## A One-Stop Shop for A.I. Models Raises \$113 Million

An investment arm of Alphabet is backing OpenRouter, which helps companies choose among hundreds of models for different software tasks.

[Read the full article →](#)

# 今天就来讲讲这个看起来没什么技术含量的“模型贩子”

The startup aims to become the go-to platform for developers to access and compare different AI models from various providers.

[Read the full article →](#)





## Summary Comparison

Source	Tokens/day	Share	
PCs (CPU only)	~280 T	22%	
PCs (gaming GPU)	~220 T	17%	
All PCs	~500 T	~40%	
Data Centers	~700 T	~60%	
Global Total	~1.2 P (quadrillion)	100%	

## Key Takeaways



询问 Qwen

自动





That gives an approximate comparison:

升级

分享



AI infrastructure	Tokens/day
CPU-only personal computers	~1-4T
Gaming PCs	~50-400T
All participating PCs	~50-400T
Commercial AI data centers	~100-500T
Ratio: PCs / data centers	~0.1x-4x

## The striking result

Under your assumption that **50% of PCs run AI inference for 12 hours/day**, distributed personal computing could plausibly become **comparable to today's commercial inference infrastructure**.



The key isn't CPU machines. It is the **gaming-GPU population**.

+ 有问题，随便问

思考



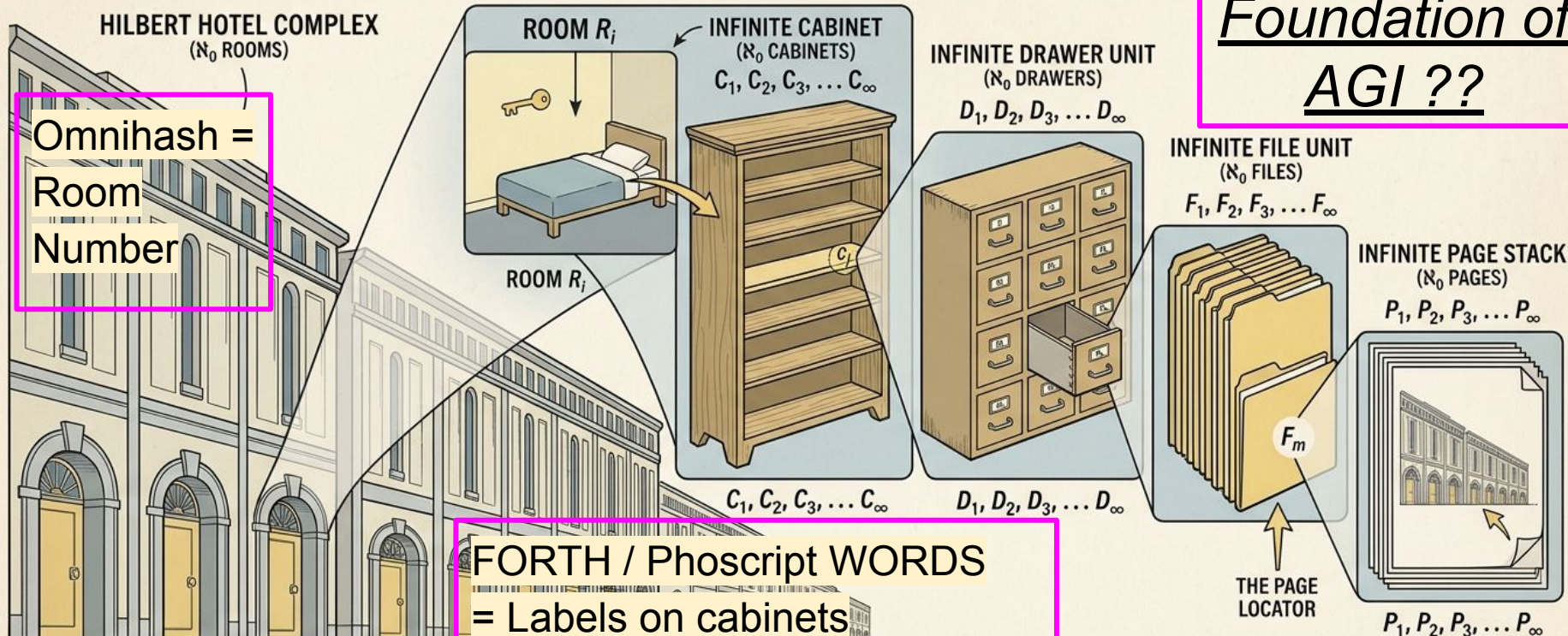
LN

potentially hundreds of millions of such machines, whereas commercial AI infrastructure is concentrated into a much smaller number of extremely powerful accelerators

# THE RECURSIVE HILBERT HOTEL

Foundation of  
AGI ??

Omnihash =  
Room  
Number



FORTH / Phoscript WORDS  
= Labels on cabinets  
& files & .....

VS.

HILBERT HOTEL

RECURSIVE HILBERT HOTEL

THE CARTESIAN PRODUCT  $\aleph^5$   
IS STILL COUNTABLE.

```
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135 {"dup", &PhosVM::dup},
136 {"depth", &PhosVM::depth},
137 {"gettype", &PhosVM::gettype},
138 {"tostr", &PhosVM::tostr},
139 {"type", &PhosVM::type},
140 {"cr", &PhosVM::cr},
141 {"s", &PhosVM::showstack},
142 {"peek", &PhosVM::peek},
143 {"peekr", &PhosVM::peekr},
144 {"pick", &PhosVM::pick},
145 {"p_M", &PhosVM::p_M},
146 {"g_M", &PhosVM::g_M},
147 {"s_M", &PhosVM::s_M},
148 {"k_M", &PhosVM::k_M},
149 {"pvs", &PhosVM::pvs},
150 {"inV", &PhosVM::inV},
151 {"vcb", &PhosVM::pick}, // varia
152 {"veq", &PhosVM::pick}, // varia
153 {"vdc", &PhosVM::pick}, // varia
154 {"find", &PhosVM::find}, // get
155 {"ADD", &PhosVM::add},
156 {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

llama.cpp PhosVM: 50 words  
JavaScript + PHP Phoscript: 150 words  
Omniscientia+GRATIS: 500 words (2026 Q4)  
LLASMA FORTH Stack Machine Words:  
5000 words (2027 Q2)  
OmniBeta (domain specific applications /  
programmer teams): 50K words (2027)  
[Cesium.js](#) + Physics: 50K words (2027)  
Delta AGI: 500K words (2028)

C/ObjC Header ▾ Tab Width: 2 ▾ Ln 11, Col 19 ▾ INS

```
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/Phos
hongwu@hongwu-Latitude-5480: ~
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/Phos
hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

h; l_s=S[2][1].value.length; s_i=S[2][1].value; s_d=s_i.substr(l_n+1,l_s-l_n-4); console.log(v_n, l_n, l_s, "--"+s_d+"--", S[2][1].value); eval('window.'+ v_n +'='+ s_d); } f_dlb() { f_dlb_bottom() { f_dlb_right() { f_dlbur() { f_dlb_url() { // alias to dlbur: f_dlbx() { f_dom() { f_download() { f_drop() { f_duet() { f_d() { v_n=(S[2][1].value.split(' ')[0]); l_n=v_n.length; l_s=S[2][1].value.length; s_i=S[2][1].value; s_d=s_i.substr(l_n+1,l_s-l_n-4); console.log(v_n, l_n, l_s, "--"+s_d+"--", S[2][1].value); window[v_n]=s_d; } f_e(){ M.F(eval(s.pop())) } f_e(){ s.push(S[2][1].value.split(' ')[S[2][1].value.split(' ').length-2]); M.F(eval(s.pop())) } f_ev(){ s.push(eval(s.pop())) } f_false() { f_file() { f_file0() { // deprecated 2021-04-02 f_forin() { var S=M.S; var A=S.pop(); var i; f_forin() { var S=M.S; var A=S.pop(); var i; var f0 = S0.$CDW[S0.cda[0]][S0.cda[1]+ 1]; console.log(JSON.stringify(S0.cda),f0); for (i in A) { console.log(A[i]) } } f_get_childre n() { f_getn() { f_get_pos() { f_gv() { f_h53() f_hhm() { f_ih() { f_imsbk_d() // import singing public key ArrayBuffer f_include() { f_init_ws() { f_input() { f_iz() { f_iza() { f_jd() { f_js() { f_jscol() { // json to column f_load() { f_load0() { // deprecated 2021-04-02 f_log() { f_lpbk() { f_lpbks() // load pbk string f_lpvk() { f_l_sign_kp() { // load sign kp from S0.KS to stack f_ms() { f_ms() { // merge string on stack f_nn_form() { f_not() { f_oe() { f_p() { f_path() // path safe b64 f_pbot() { f_pick() { f_pos() { f_pw_get() { f_pw_mk() { f_rec() { f_right() { f_rot() { f_row() { f_s2ab() { f_savekp() { f_savek s() { f_scene() { f_slice() { f_splice() { f_ss() { f_ss() { console.log(' ss: ',s.length, JSON.stringify(s[s.length-1])) } f_sse() f_sse_log() f_ssl() { // stack slice f_s v() { f_tn() { f_to() { f_true() { f_typeof() { f_uuid() { f_val() { f_value () { // in f_rsa_oaep()
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/Phos$ grep -h 'n f_' *js|sort|uniq|wc
139 621 4825
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/Phos$
```

» `Object.keys(S0.$CDW)`

← `Array(35) [ "phos", "B_F", "F_O", "AJAX", "A_SS", "AJAXP", "B_AUTH", "A_AUTH", "B_AUTH_U", "A_AUTH_U", ... ]`

» `JSON.stringify(Object.keys(S0.$CDW))`

← `'["phos","B_F","F_O","AJAX","A_SS","AJAXP","B_AUTH","A_AUTH","B_AUTH_U","A_AUTH_U","do_sign","im_pbk","mk_sign","ecr_sig","rsa_sign","do_scj","ab_scj","u_chat","a_scj","x_dsc","u_do_sign","u_mk_sign","u_rsa_sign","u_do_scj","u_ecr_sig","u_dsc","l_do_sign","l_mk_sign","l_do_scj","l_ecr_sig","l_rsa_sign","l_dsc","AJE","F_E","dlb_n"]'`

» |



ActivitiesTerminal

Sab 22 Ogos, 07:31

zh

hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

hongwu@hongwu-Latitude-5480: ~

hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

hongwu@hongwu-Latitude-5480: /var/www/dmeta/o/auth

```
split_match() // split first string by matches fgl_spln() // $A $n splice: array_splice($A, $n) fgl_spls() // splice $S fgl_sq() fgl_sqr() // square fgl_ss() fgl/sscdw() fgl/sscdwm() // merge with old SESSION.CDW 2024-12 fgl/ssid() // set session id fgl/sslh() // stripslashes fgl_ssl() // substring-length: array start length ssl: fgl/sspl() // splice $$; full $n $l sspl: array_splice($A, $n, $l) fgl/ssrg() fgl/ss_start() fgl/sstr() // substr($, n, l) fgl/st() fgl/star() // * character fgl_strval() fgl/stv() // fgl_substr fgl_substr() fgl_sv() fgl/svt() // save Top of Stack, used for 1 in N out, N in N out, Chain Alias etc. fgl_swap() fgl_sys() fgl_time() fgl_tn() { fgl/tostr() fgl/touuid() // 16 bytes to UUID fgl_trim() fgl_trims() fgl_trims() // remove spaces fgl_t() // save Top of Stack, used for 1 in N out, N in N out, Chain Alias etc. fgl_tus() // microsecond time fgl_tz() fgl_ucd w() fgl_ud() fgl_ue() fgl_ufa() fgl/unindex() fgl_uniq() // array_unique fgl/unpack() // unpack array, push elements to stack fgl_url() // preg match url fgl_usa() // unset key in array fgl_usc() // underscore fgl_user() fgl_us() // _/ special string for log file name search fgl/usss() // unset session variable fgl_uss() // update session variable fgl_uuid() fgl_va() fgl_var() fgl_vs() // call var_src() fgl_vt() fgl_vv() fgl_w() fgl_w0() // deprecated see 2025-1-6 fgl_w480() fgl_wa() fgl/wdbg() // debug write 2024-11 fgl_while() // while loop derived from array map(): InputArray OutputArray while: WORD Condition fgl_wn() fgl_wnl() // write append with newline fgl_word() // process words fgl_wpn() // write path node fgl_wx() // write if file not exist, for hash file, return date if exists fgl_x() fgl_xblk() fgl_xc() { echo " xc: exception ..."; } fgl_xcmd() // invert grep exclude "a_cmd" fgl_xforeach() fgl_xif() fgl_xr() // invert grep exclude "a_cmd" fgl_xs() fgl_x_sp() fgl_xword($v) fgl_xw() // suppress warning fgl_叠(){ fgl_s(); } fgl_复() fgl_时() fgl_添() fgl_组() fgl_编() fgl_解(){ fgl_jd(); } fgl_读(){ fgl_fgc(); }
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ grep -h 'n fgl_' *php|sort|uniq|wc
420 1876 13846
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$
```

```
fgl_s 364 < 3 > array (0 => array (0 => 'phos.php', 1 => 'cdw:', 2 => 'ak:', 3 => 's:',), 1 => 'p
hos.php', 2 => array (0 => 'chtag', 1 => 'chn', 2 => 'atag', 3 => 'aot', 4 => 'add', 5 => 'chk_sp',
6 => 'auth_pass', 7 => 'auth_csv', 8 => 'w_cdw', 9 => 'r_cdw', 10 => 'u_cdw', 11 => 'req_auth', 12
=> 'i_shm', 13 => 'shm_inc', 14 => 'chat', 15 => 'nick', 16 => 'req_auth_sc', 17 => 'b_cdw', 18 => '
ss_cdw', 19 => 's_ss', 20 => 'uu_ls', 21 => 'uu_mk', 22 => 'a_cache', 23 => 'a_comment', 24 => 'a_ke
yword', 25 => 'a_html', 26 => 'pbk_file', 27 => 'to_uuid', 28 => 'uuid_chain', 29 => 'uu_chain', 30
=> 'uuid_auth', 31 => 'a_A', 32 => 'a_AB', 33 => 'last_pbk', 34 => 'has_uu', 35 => 'uu_node', 36 =>
'uu_l1', 37 => 'key_u', 38 => 'f_ls', 39 => 'fea_ls', 40 => 'echo_on', 41 => 'a_vkn', 42 => 'id_url'
, 43 => 'p_o', 44 => 'p_i', 45 => 'comment_add', 46 => 'r_v', 47 => 'price_dcnt', 48 => 'ls_url', 49
=> 'parent_0', 50 => 'uu_l2', 51 => 'nn_sv', 52 => 'nn_r', 53 => 'uu_nn', 54 => 'nn_u2', 55 => 'g_r
g', 56 => 'g_rg_0', 57 => 'uu_l3', 58 => 'nn_uu', 59 => 'grhg', 60 => 'ctmp', 61 => 'hgnc', 62 => 'h
f', 63 => 'hgje', 64 => 'xhash', 65 => 'cmdh', 66 => 'xhje', 67 => 'xh', 68 => 'hje', 69 => 'echo',
70 => 'wh', 71 => 'hg_', 72 => 'l_cdwjs', 73 => 'm_para', 74 => 'h_mkdir', 75 => 'a_fw', 76 => 'a_w'
, 77 => 'h_b64', 78 => 'last', 79 => 'lex', 80 => 'newest_0', 81 => 'old_record', 82 => 'mk_hash', 8
3 => 'fwm1', 84 => 'fwm2', 85 => 'wm1', 86 => 'wm2', 87 => 'htmp', 88 => 'mkhf', 89 => 'fkmod', 90 =
> 'rhf', 91 => 'hgp', 92 => 'chkhd', 93 => 'hbp', 94 => 'g_ohc', 95 => 'h_pwd', 96 => 'u_g', 97 => '
u_ff', 98 => 'u_spx', 99 => 'u_2fr', 100 => 'u_xfr', 101 => 'm_hcdw', 102 => 'rm_th', 103 => 'm_cdw'
, 104 => 'l_hcdw', 105 => 'r_hcdw', 106 => 'var_bv', 107 => 'search_user', 108 => 'g_x', 109 => 'AUT
H_SP', 110 => 'AUTH_SP0', 111 => 'AUTH_A', 112 => 'AUTH_A0', 113 => 'AUTH_B', 114 => 'AUTH_W', 115 =
> 'AUTH_W0', 116 => 'AUTH_WF', 117 => 'a_expl', 118 => 'guid_sp', 119 => 'm_ecr_t', 120 => 'm_ecr',
121 => 'g_uid', 122 => 'get_u', 123 => 's_pbk', 124 => 'get0', 125 => 'g_pbk', 126 => 'newest', 127
=> 'fn_newest', 128 => 'g_msg', 129 => 's_msg', 130 => 's_msg_d',),)
(base) hongwu@hongwu-Latitude-5480:/var/www/dmeta/o/auth$ php phos.php cdw: ak: s:
```

```
133 using Handler = bool (PhosVM::*)(int);
134 std::unordered_map<std::string, Handler> handlers = {
135 {"dup", &PhosVM::dup},
136 {"depth", &PhosVM::depth},
137 {"gettype", &PhosVM::gettype},
138 {"tostr", &PhosVM::tostr},
139 {"type", &PhosVM::type},
140 {"cr", &PhosVM::cr},
141 {"s", &PhosVM::showstack},
142 {"peek", &PhosVM::peek},
143 {"peekr", &PhosVM::peekr},
144 {"pick", &PhosVM::pick},
145 {"p_M", &PhosVM::p_M},
146 {"g_M", &PhosVM::g_M},
147 {"s_M", &PhosVM::s_M},
148 {"k_M", &PhosVM::k_M},
149 {"pvs", &PhosVM::pvs},
150 {"inV", &PhosVM::inV},
151 {"vcb", &PhosVM::pick}, // varia
152 {"veq", &PhosVM::pick}, // varia
153 {"vdc", &PhosVM::pick}, // varia
154 {"find", &PhosVM::find}, // get
155 {"ADD", &PhosVM::add},
156 {"push", &PhosVM::push}, {"md5", &PhosVM::md5},
```

llama.cpp PhosVM: 50 words  
JavaScript + PHP Phoscript: 150 words  
Omniscientia+GRATIS: 500 words (2026 Q4)  
LLASMA FORTH Stack Machine Words:  
5000 words (2027 Q2)  
OmniBeta (domain specific applications /  
programmer teams): 50K words (2027)  
[Cesium.js](#) + Physics: 50K words (2027)  
Delta AGI: 500K words (2028)