

*Lua Workshop 2016, San Francisco*

---

# LuaJIT: Something interesting inside

[corsix.org/wshop16.pdf](http://corsix.org/wshop16.pdf)

---

Peter Cawley

corsix on GitHub / www / ...

Not Mike Pall

(or Vyacheslav Egorov  
or Thomas Fransham or ...)

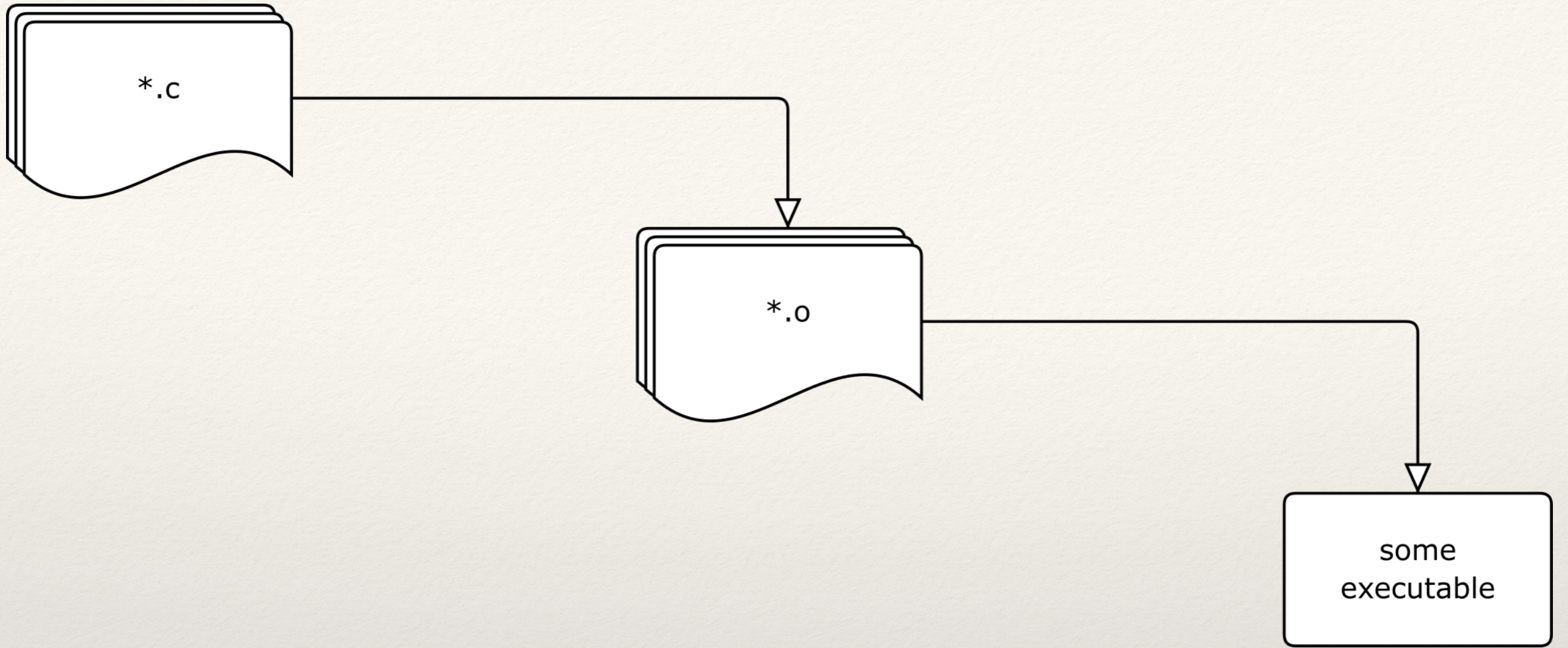
Not my employer

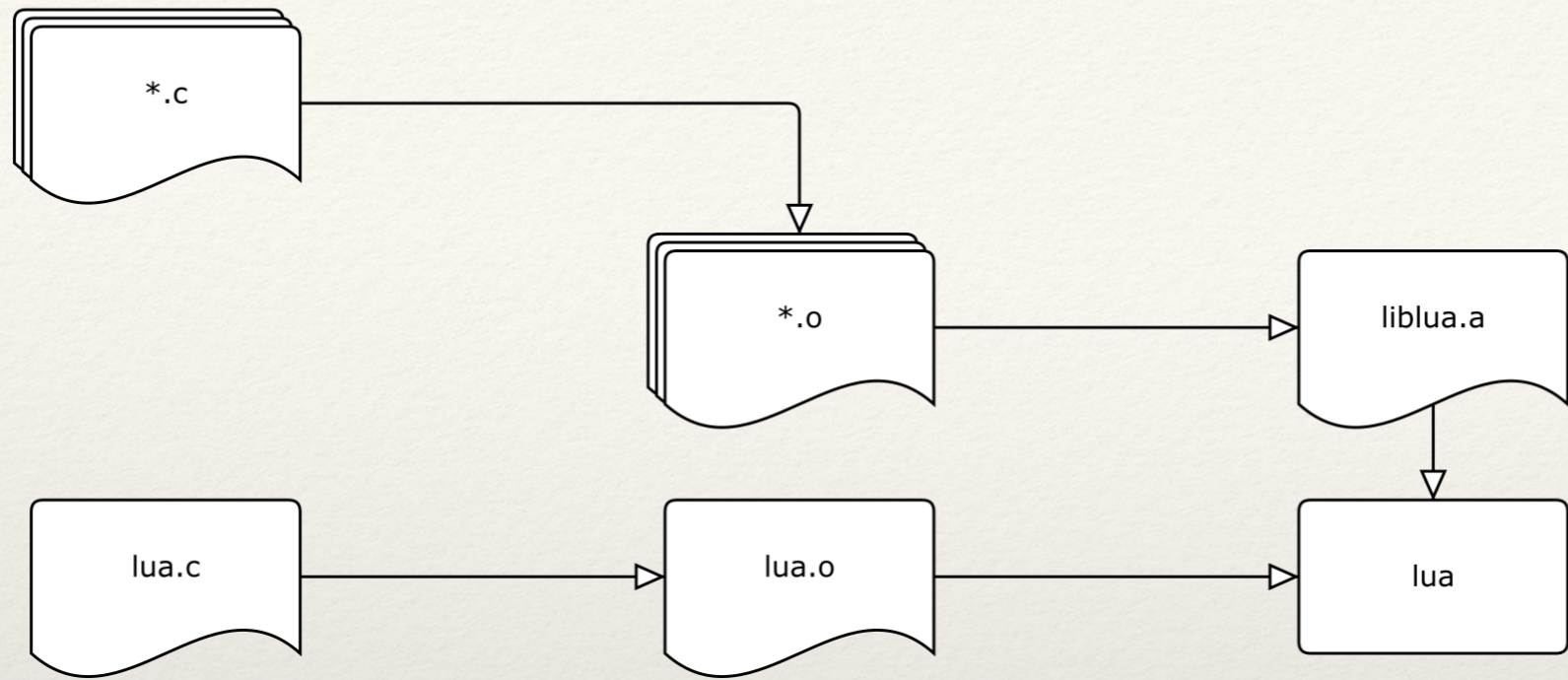
```
git clone http://github.com/LuaJIT/LuaJIT
cd LuaJIT
git checkout v2.1
make -j
sudo make install
```

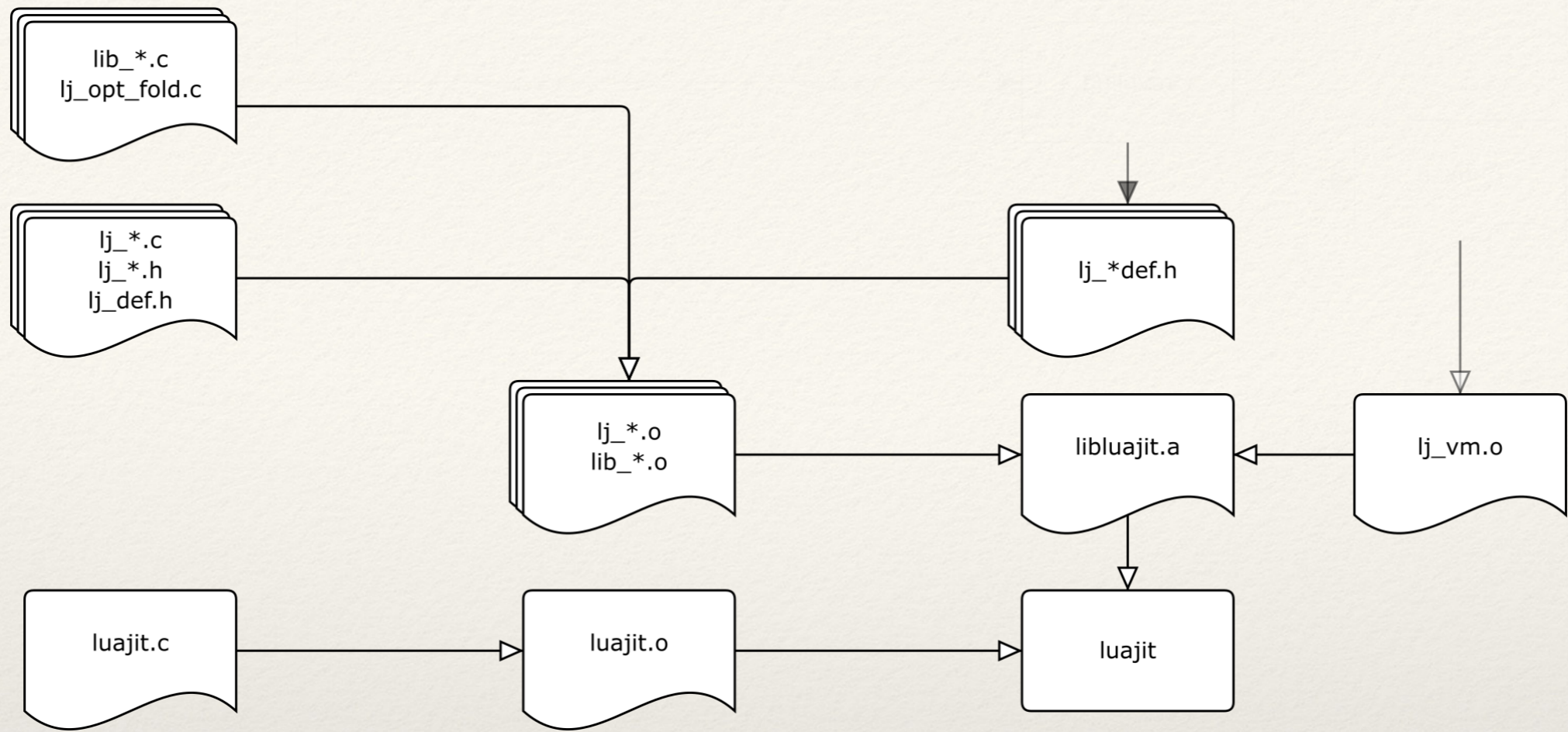
```
git clone http://github.com/LuaJIT/LuaJIT  
cd LuaJIT  
git checkout v2.1
```

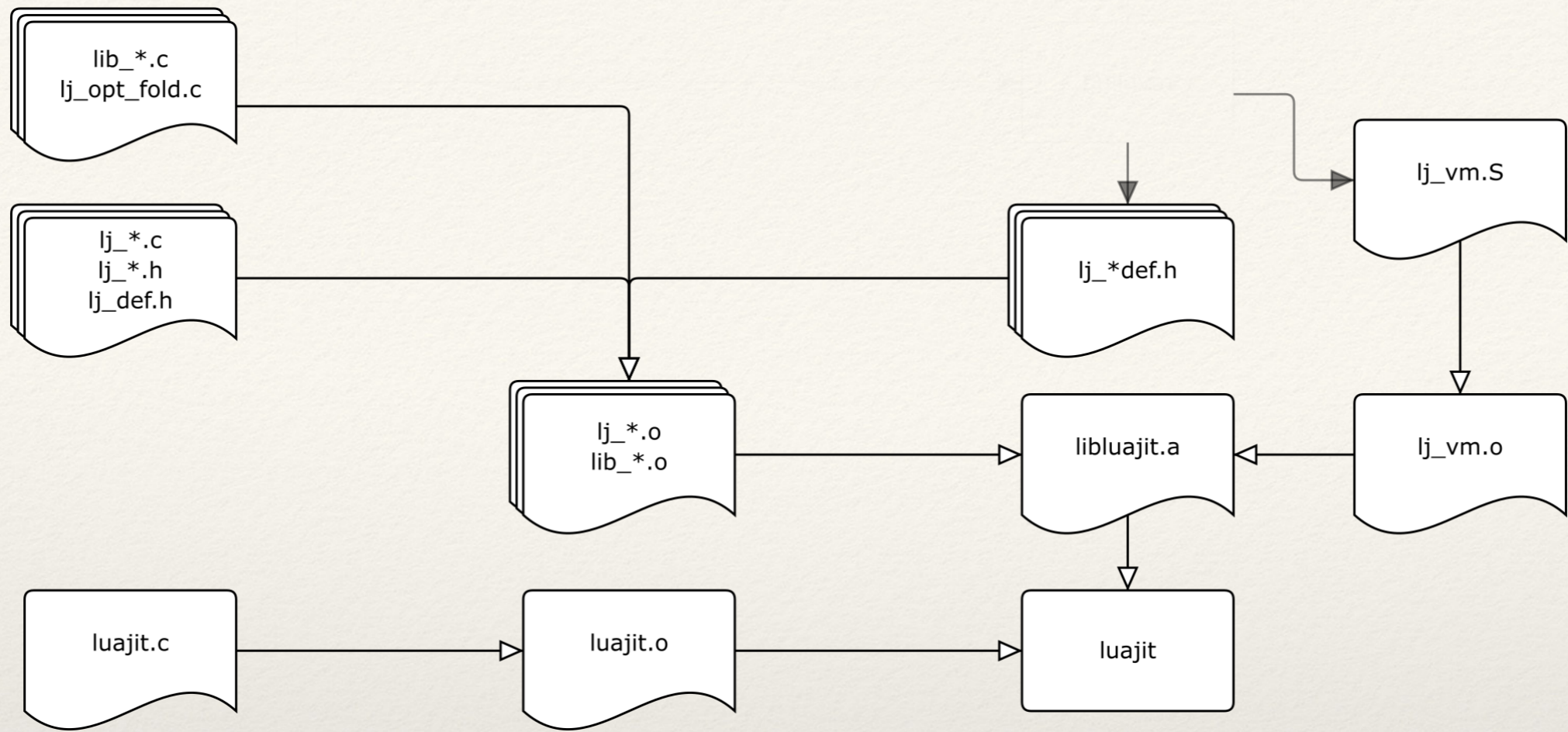
```
make -j
```

```
sudo make install
```







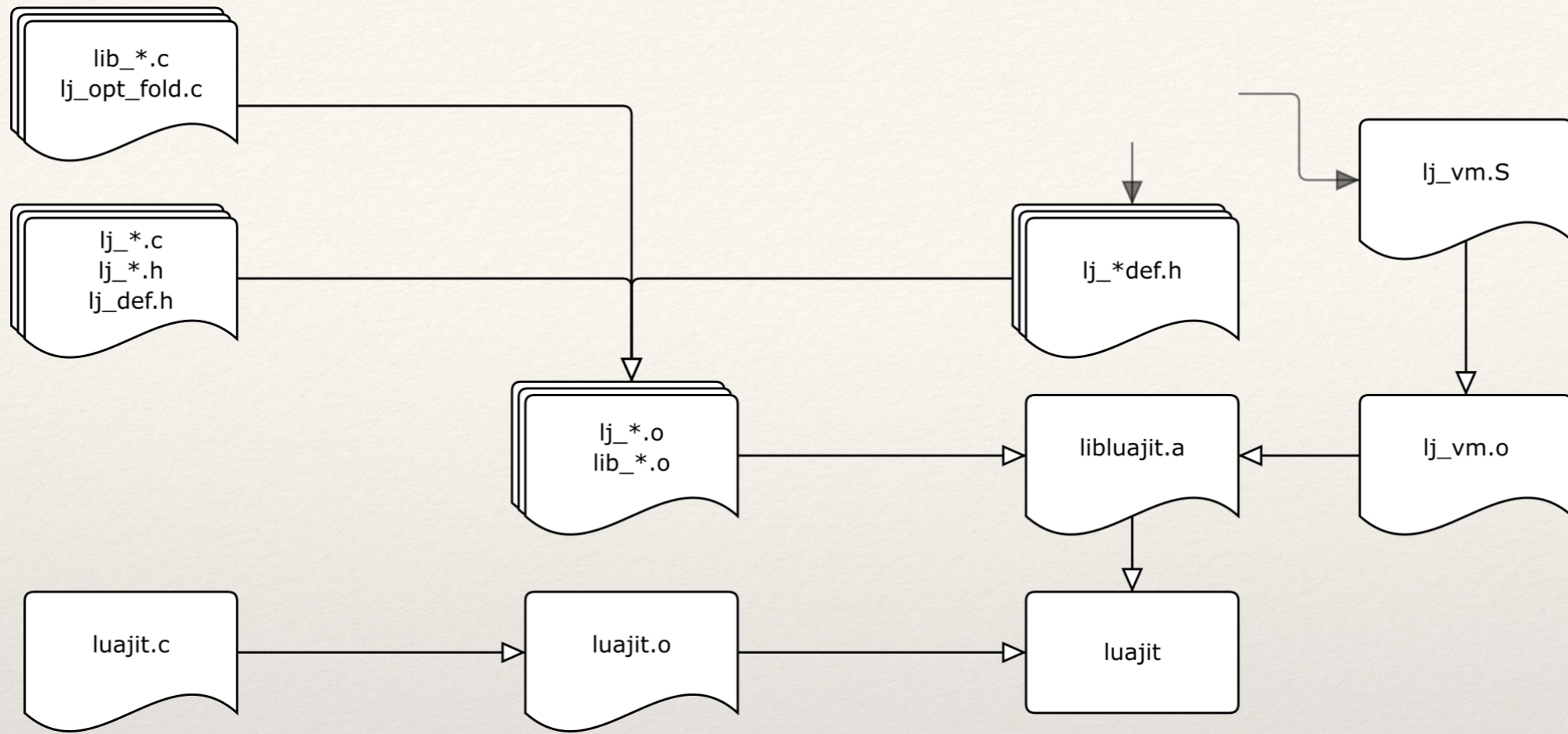


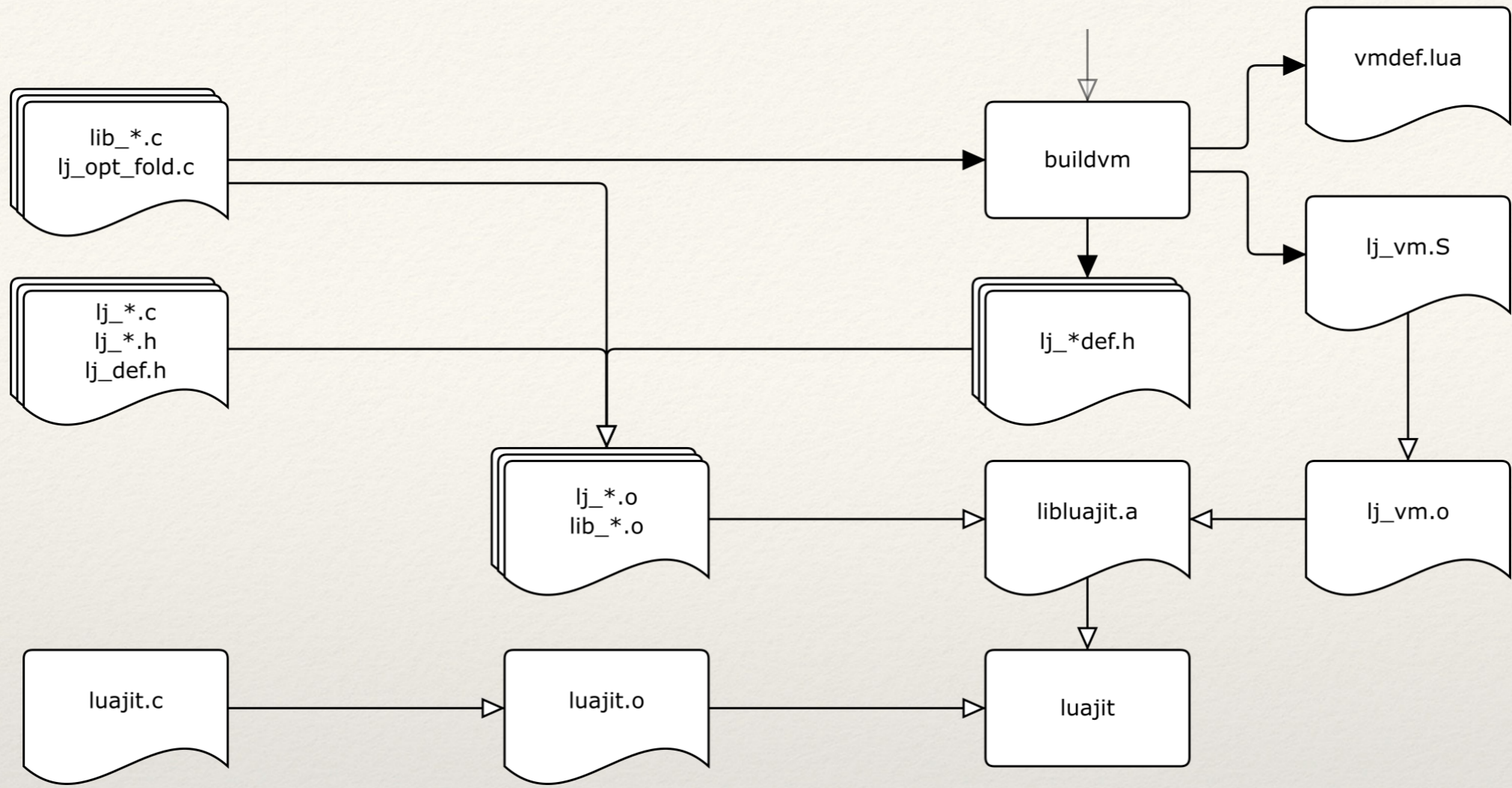


```
    .private_extern _lj_BC_TDUP
_lj_BC_TDUP:
    .byte 72,247,208,139,108,36,24,65,139,142,248,243,255,255,137,92
    .byte 36,28,65,59,142,252,243,255,255,137,85,16,115,47,65,139
    .byte 52,135,137,239
    call _lj_tab_dup
    .byte 139,85,16,15,182,75,253,137,4,202,199,68,202,4,244,255
    .byte 255,255,139,3,15,182,204,15,182,232,131,195,4,193,232,16
    .byte 65,255,36,238,137,239
    call _lj_gc_step_fixtop
    .byte 15,183,67,254,72,247,208,235,193
```

```
    .private_extern _lj_BC_GGET
_lj_BC_GGET:
    .byte 72,247,208,139,106,248,139,109,8,65,139,4,135,233,193,0
    .byte 0,0
```

```
    .private_extern _lj_BC_GSET
_lj_BC_GSET:
    .byte 72,247,208,139,106,248,139,109,8,65,139,4,135,233,113,2
    .byte 0,0
```





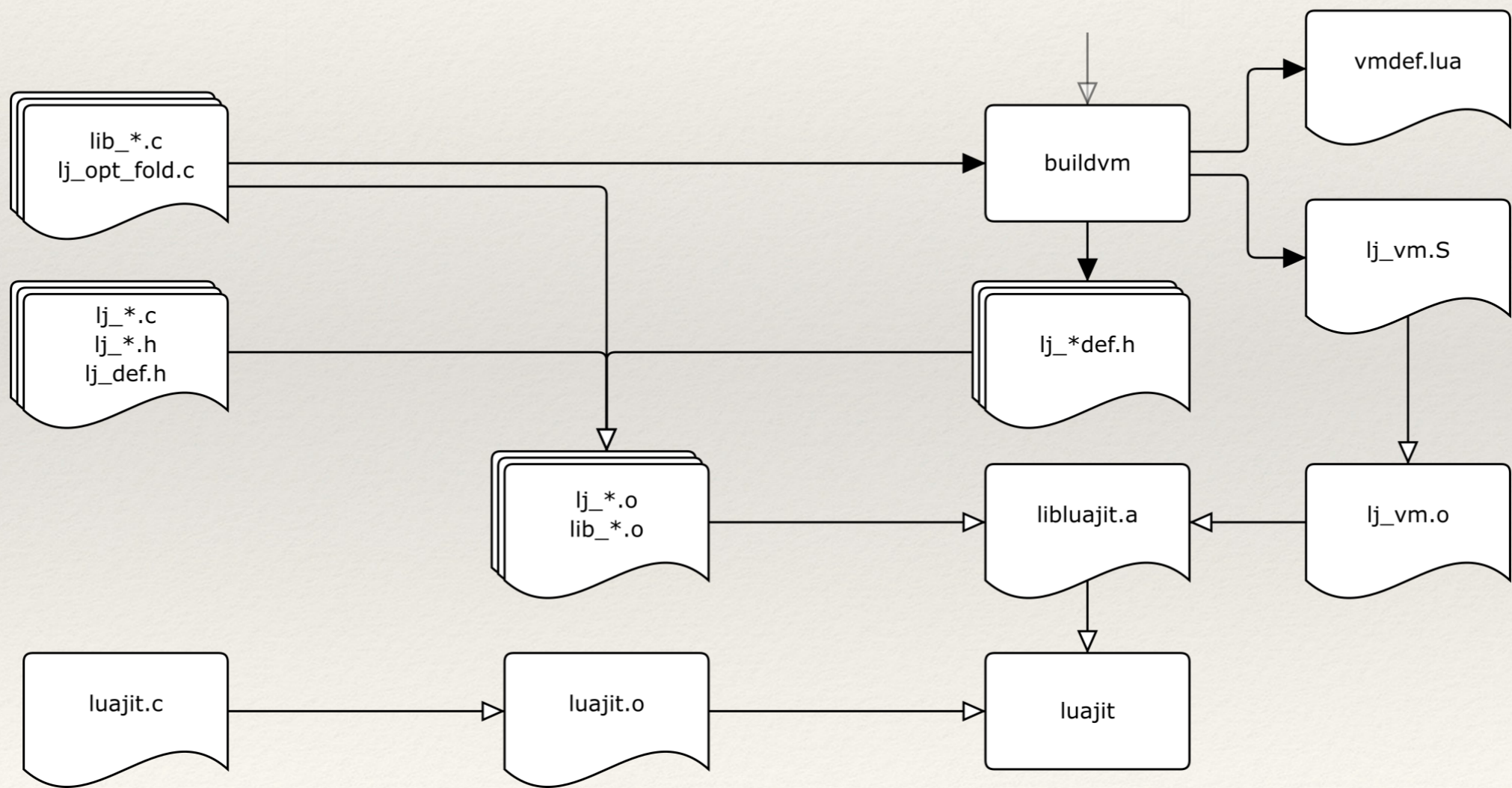
```
LJLIB_CF(raisequal)      LJLIB_REC(.)
{
    cTValue *o1 = lj_lib_checkany(L, 1);
    cTValue *o2 = lj_lib_checkany(L, 2);
    setboolV(L->top-1, lj_obj_equal(o1, o2));
    return 1;
}
```

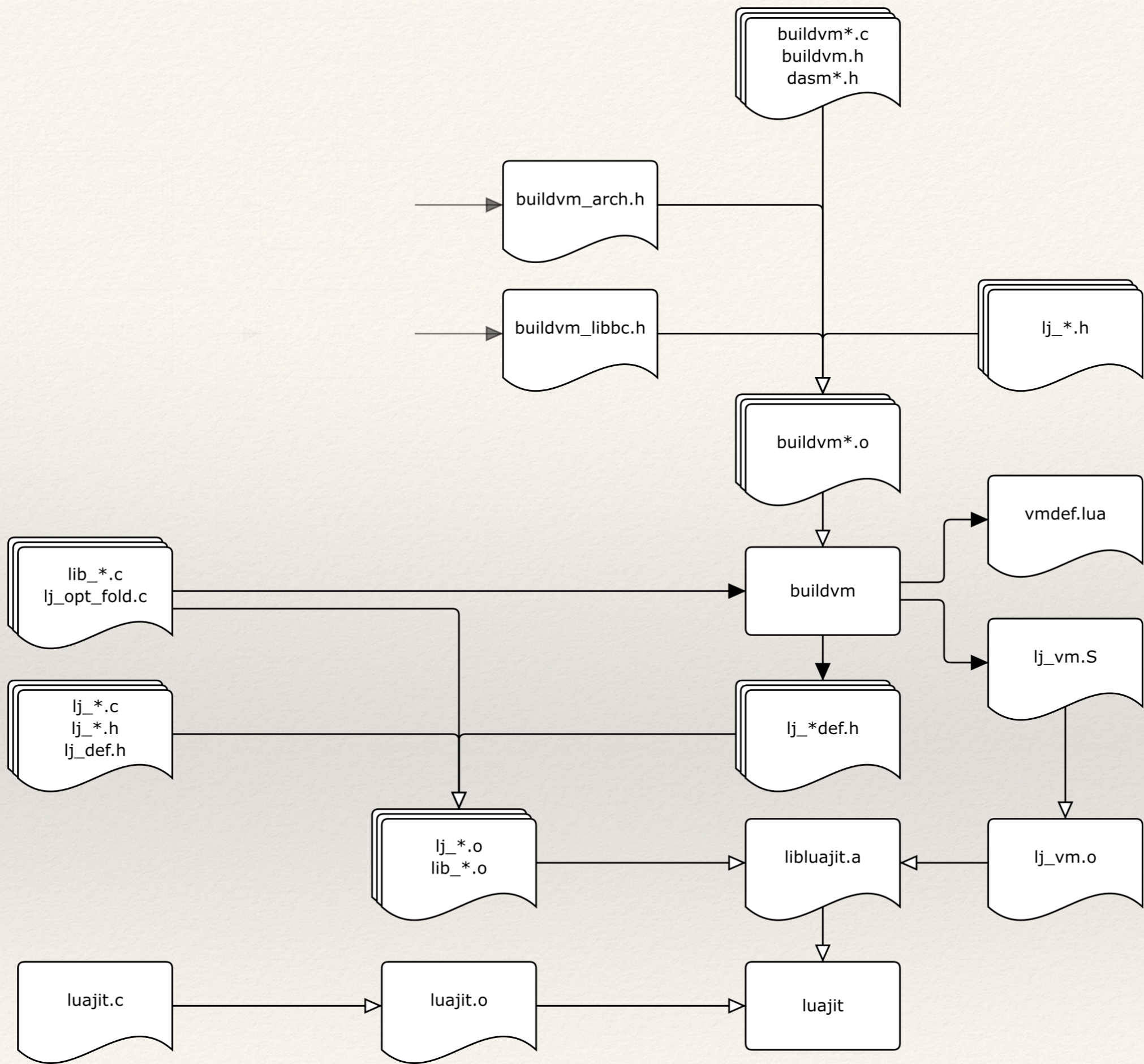
```
LJLIB_CF(rawequal)      LJLIB_REC(.)
{
    cTValue *o1 = lj_lib_checkany(L, 1);
    cTValue *o2 = lj_lib_checkany(L, 2);
    setboolV(L->top-1, lj_obj_equal(o1, o2));
    return 1;
}
```

```
#define LJLIB_CF(name)      static int lj_cf_##name(lua_State *L)
#define LJLIB_REC(handler)
```

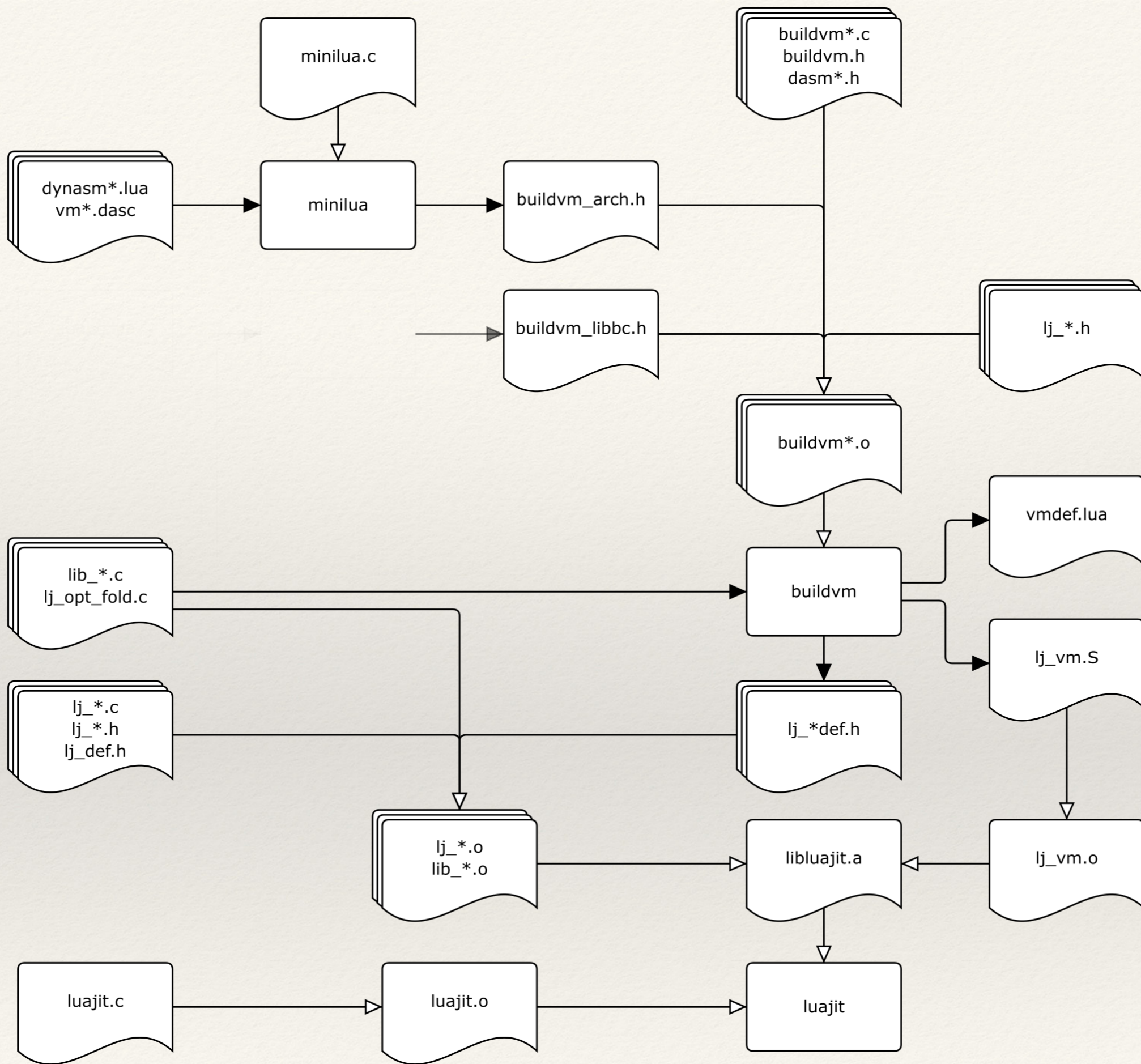
```
LJFOLD(ADD any KINT)
LJFOLD(ADDOV any KINT)
LJFOLD(SUBOV any KINT)
LJFOLDF(simplify_intadd_k)
{
    if (fright->i == 0) /* i o 0 ==> i */
        return LEFTFOLD;
    return NEXTFOLD;
}
```

```
#define LJFOLDF(name) static TRef LJ_FASTCALL fold_##name(jit_State *J)
#define LJFOLD(x)
```









```

case BC_TDUP:
| ins_AND // RA = dst, RD = table const (~) (holding template table)
| mov L:RB, SAVE_L
| mov RA, [DISPATCH+DISPATCH_GL(gc.total)]
| mov SAVE_PC, PC
| cmp RA, [DISPATCH+DISPATCH_GL(gc.threshold)]
| mov L:RB->base, BASE
| jae >3
|2:
| mov TAB:FCARG2, [KBASE+RD*4] // Caveat: FCARG2 == BASE
| mov L:FCARG1, L:RB // Caveat: FCARG1 == RA
| call extern lj_tab_dup@8 // (lua_State *L, Table *kt)
| // Table * returned in eax (RC).
| mov BASE, L:RB->base
| movzx RA, PC_RA
| mov [BASE+RA*8], TAB:RC
| mov dword [BASE+RA*8+4], LJ_TTAB
| ins_next
|3:
| mov L:FCARG1, L:RB
| call extern lj_gc_step_fixtop@4 // (lua_State *L)
| movzx RD, PC_RD // Need to reload RD.
| not RDa
| jmp <2
break;

```

```

case BC_GGET:
| ins_AND // RA = dst, RD = str const (~)
| mov LFUNC:RB, [BASE-8]
| mov TAB:RB, LFUNC:RB->env
| mov STR:RC, [KBASE+RD*4]
| jmp ->BC_TGETS_Z
break;

```

```

case BC_GSET:
| ins_AND // RA = src, RD = str const (~)
| mov LFUNC:RB, [BASE-8]
| mov TAB:RB, LFUNC:RB->env
| mov STR:RC, [KBASE+RD*4]
| jmp ->BC_TSETS_Z
break;

```

```

case BC_TDUP:
| ins_AND
| mov L:RB, SAVE_L
| mov RA, [DISPATCH+DISPATCH_GL(gc.total)]
| mov SAVE_PC, PC
| cmp RA, [DISPATCH+DISPATCH_GL(gc.threshold)]
| mov L:RB->base, BASE
| jae >3
| 2:
| mov TAB:FCARG2, [KBASE+RD*4]
| mov L:FCARG1, L:RB
| call extern lj_tab_dup@8
| // Table * returned in eax (RC).
| mov BASE, L:RB->base
| movzx RA, PC_RA
| mov [BASE+RA*8], TAB:RC
| mov dword [BASE+RA*8+4], LJ_TTAB
| ins_next
| 3:
| mov L:FCARG1, L:RB
| call extern lj_gc_step_fixtop@4
| movzx RD, PC_RD
| not RDa
| jmp <2
break;

```

```

case BC_GGET:
| ins_AND // RA = dst, RD = str const (~)
| mov LFUNC:RB, [BASE-8]
| mov TAB:RB, LFUNC:RB->env
| mov STR:RC, [KBASE+RD*4]
| jmp ->BC_TGETS_Z
break;

```

```

case BC_GSET:
| ins_AND // RA = src, RD = str const (~)
| mov LFUNC:RB, [BASE-8]
| mov TAB:RB, LFUNC:RB->env
| mov STR:RC, [KBASE+RD*4]
| jmp ->BC_TSETS_Z
break;

```

```

lj_BC_TDUP:
not rax
mov ebp, dword ptr [rsp+60h]
mov ecx, dword ptr [rbx-0C08h]
mov dword ptr [rsp+64h], esi
cmp ecx, dword ptr [rbx-0C04h]
mov dword ptr [rbp+10h], edx
jae lj_BC_TDUP+49h
mov edx, dword ptr [rdi+rax*4]
mov ecx, ebp
call lj_tab_dup
mov edx, dword ptr [rbp+10h]
movzx ecx, byte ptr [rsi-3]
mov dword ptr [rdx+rcx*8], eax
mov dword ptr [rdx+rcx*8+4], 0FFFFFFF4h
mov eax, dword ptr [rsi]
movzx ecx, ah
movzx ebp, al
add esi, 4
shr eax, 10h
jmp qword ptr [rbx+rbp*8]
mov ecx, ebp
call lj_gc_step_fixtop
movzx eax, word ptr [rsi-2]
not rax
jmp lj_BC_TDUP+1Ch

```

```

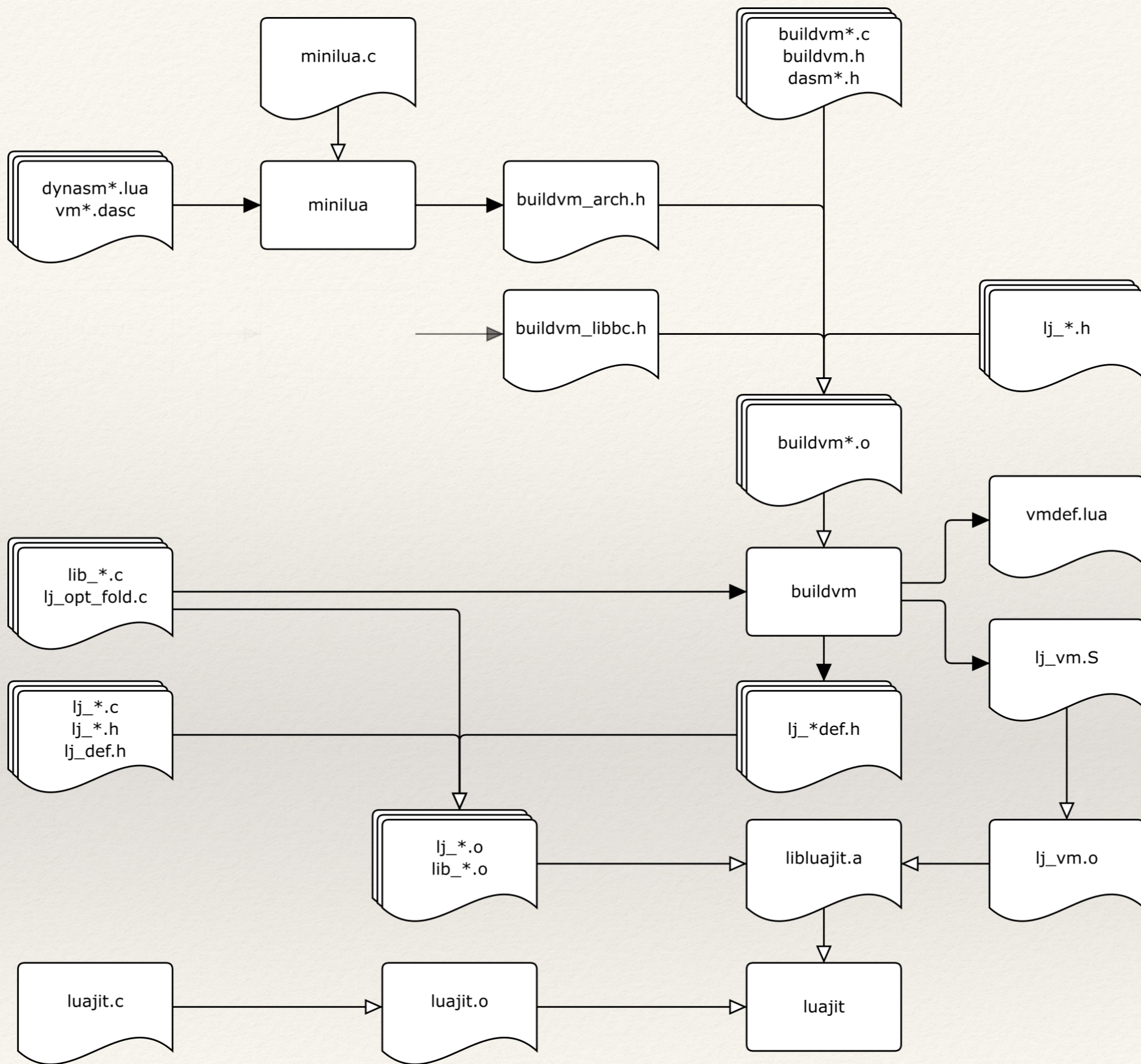
lj_BC_GGET:
not rax
mov ebp, dword ptr [rdx-8]
mov ebp, dword ptr [rbp+8]
mov eax, dword ptr [rdi+rax*4]
jmp lj_BC_TGETS+1Ah

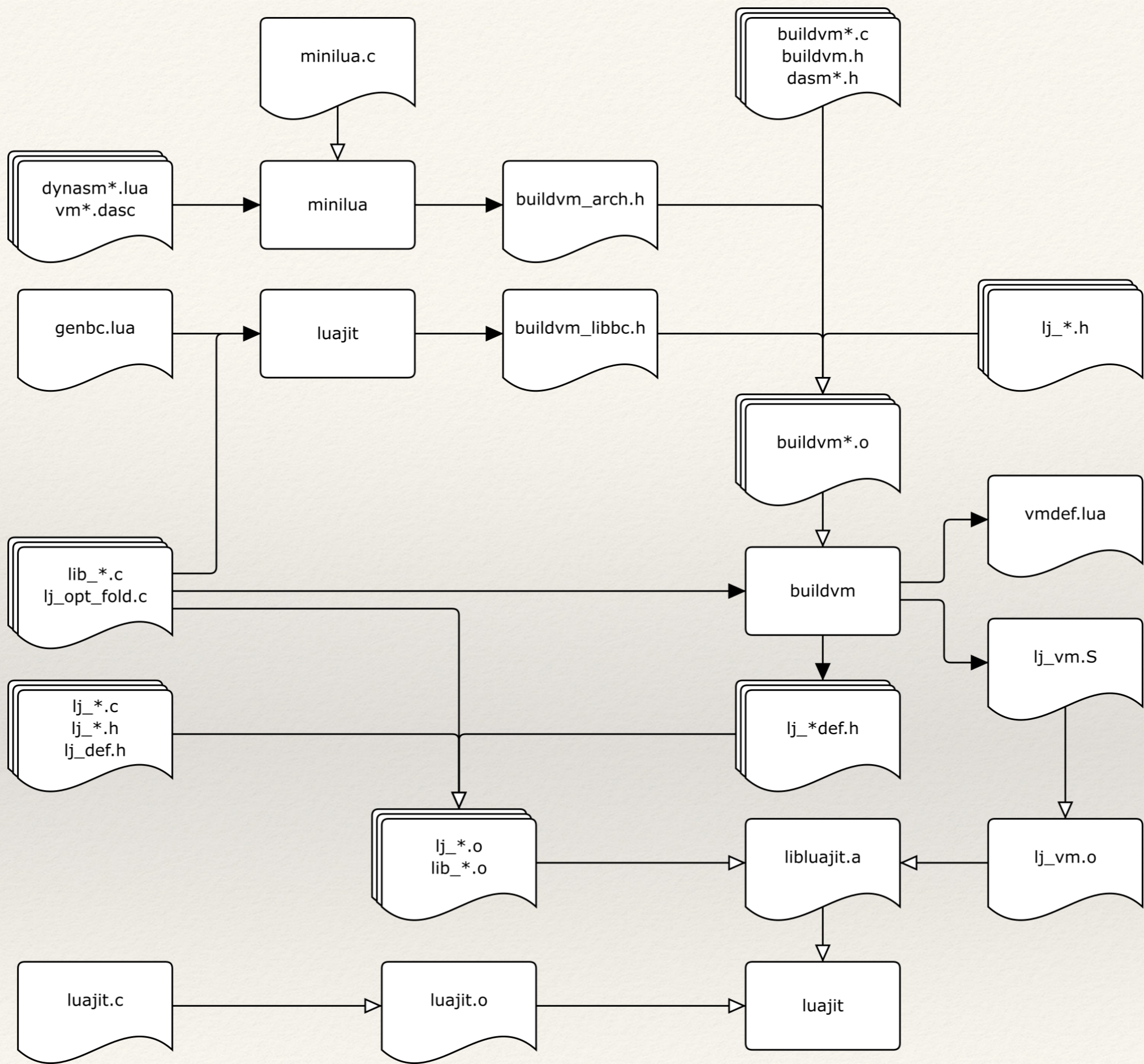
```

```

lj_BC_GSET:
not rax
mov ebp, dword ptr [rdx-8]
mov ebp, dword ptr [rbp+8]
mov eax, dword ptr [rdi+rax*4]
jmp lj_BC_TSETS+1Ah

```





```
LJLIB_LUA(table_foreachi) /*  
  function(t, f)  
    CHECK_tab(t)  
    CHECK_func(f)  
    for i=1,#t,1 do  
      local r = f(i, t[i])  
      if r ~= nil then return r end  
    end  
  end  
*/
```

```
#define LJLIB_LUA(name)
```

```

LJLIB_LUA(table_foreachi) /*
  function(t, f)
    CHECK_tab(t)
    CHECK_func(f)
    for i=1,#t,1 do
      local r = f(i, t[i])
      if r ~= nil then return r end
    end
  end
*/

```

```

#define LJLIB_LUA(name)

```

```

-- BYTECODE -- builtin:foreach--1
0001      ISTYPE      0  12
0002      ISTYPE      1   9
0003      KSHORT      2   1
0004      LEN          3   0
0005      KSHORT      4   1
0006      FORI         2 => 0015
0007 => MOV           6   1
0008      MOV          8   5
0009      TGETR        9   0   5
0010      CALL         6   2   3
0011      ISEQP        6   0
0012      JMP          7 => 0014
0013      RET1         6   2
0014 => FORL          2 => 0007
0015 => RET0          0   1

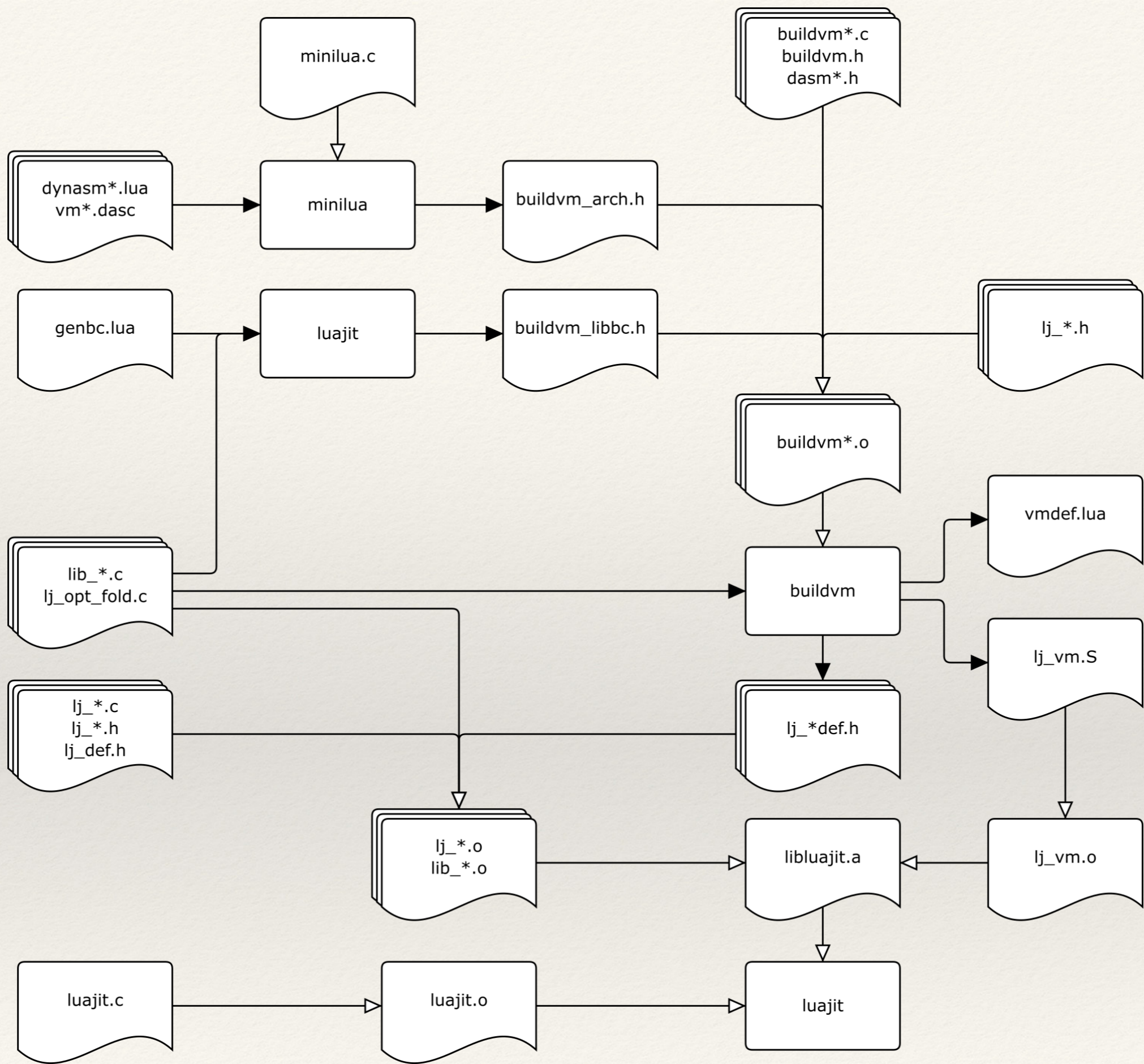
```

```
static const uint8_t libbc_code[] = {
0,1,2,0,0,1,2,24,1,0,0,76,1,2,0,241,135,158,166,3,220,203,178,130,4,0,1,2,0,
0,1,2,24,1,0,0,76,1,2,0,243,244,148,165,20,198,190,199,252,3,0,1,2,0,0,0,3,
16,0,5,0,21,1,0,0,76,1,2,0,0,2,9,0,0,0,15,16,0,12,0,16,1,9,0,41,2,1,0,21,3,
0,0,41,4,1,0,77,2,8,128,18,6,1,0,18,7,5,0,59,8,5,0,66,6,3,2,10,6,0,0,88,7,1,
128,76,6,2,0,79,2,248,127,75,0,1,0,0,2,10,0,0,0,16,16,0,12,0,16,1,9,0,43,2,
0,0,18,3,0,0,41,4,0,0,88,5,7,128,18,7,1,0,18,8,5,0,18,9,6,0,66,7,3,2,10,7,0,
0,88,8,1,128,76,7,2,0,70,5,3,3,82,5,247,127,75,0,1,0,0,1,2,0,0,0,3,16,0,12,
0,21,1,0,0,76,1,2,0,0,2,10,0,0,2,30,16,0,12,0,21,2,0,0,11,1,0,0,88,3,7,128,
8,2,0,0,88,3,23,128,59,3,2,0,43,4,0,0,64,4,2,0,76,3,2,0,88,3,18,128,16,1,14,
0,41,3,1,0,3,3,1,0,88,3,14,128,3,1,2,0,88,3,12,128,59,3,1,0,22,4,1,1,18,5,2,
0,41,6,1,0,77,4,4,128,23,8,1,7,59,9,7,0,64,9,8,0,79,4,252,127,43,4,0,0,64,4,
2,0,76,3,2,0,75,0,1,0,0,2,0
};
```

```
static const struct { const char *name; int ofs; } libbc_map[] = {
{"math_deg",0},
{"math_rad",25},
{"string_len",50},
{"table_foreachi",69},
{"table_foreach",136},
{"table_getn",207},
{"table_remove",226},
{NULL,355}
};
```



```
static const lua_CFunction lj_lib_cf_table[] = {
    lj_cf_table_maxn,
    lj_cf_table_insert,
    lj_cf_table_concat,
    lj_cf_table_sort
};
static const uint8_t lj_lib_init_table[] = {
89,57,8,249,8,'f','o','r','e','a','c','h','i',0,2,9,0,0,0,15,16,0,12,0,16,1,9,
0,41,2,1,0,21,3,0,0,41,4,1,0,77,2,8,128,18,6,1,0,18,7,5,0,59,8,5,0,66,6,3,2,
10,6,0,0,88,7,1,128,76,6,2,0,79,2,248,127,75,0,1,0,249,7,'f','o','r','e','a',
'c','h',0,2,10,0,0,0,16,16,0,12,0,16,1,9,0,43,2,0,0,18,3,0,0,41,4,0,0,88,5,7,
128,18,7,1,0,18,8,5,0,18,9,6,0,66,7,3,2,10,7,0,0,88,8,1,128,76,7,2,0,70,5,3,
3,82,5,247,127,75,0,1,0,249,4,'g','e','t','n',0,1,2,0,0,0,3,16,0,12,0,21,1,
0,0,76,1,2,0,4,'m','a','x','n',6,'i','n','s','e','r','t',249,6,'r','e','m','o',
've',0,2,10,0,0,2,30,16,0,12,0,21,2,0,0,11,1,0,0,88,3,7,128,8,2,0,0,88,3,
23,128,59,3,2,0,43,4,0,0,64,4,2,0,76,3,2,0,88,3,18,128,17,1,15,0,41,3,1,0,3,
3,1,0,88,3,14,128,3,1,2,0,88,3,12,128,59,3,1,0,22,4,1,1,18,5,2,0,41,6,1,0,77,
4,4,128,23,8,1,7,59,9,7,0,64,9,8,0,79,4,252,127,43,4,0,0,64,4,2,0,76,3,2,0,
75,0,1,0,0,2,6,'c','o','n','c','a','t',4,'s','o','r','t',254,254,255
};
```



```
LJLIB_CF(rawequal)      LJLIB_REC(.)
{
    cTValue *o1 = lj_lib_checkany(L, 1);
    cTValue *o2 = lj_lib_checkany(L, 2);
    setboolV(L->top-1, lj_obj_equal(o1, o2));
    return 1;
}
```

```
#define LJLIB_CF(name)      static int lj_cf_##name(lua_State *L)
#define LJLIB_REC(handler)
```

```

LJLIB_LUA(table_foreachi) /*
  function(t, f)
    CHECK_tab(t)
    CHECK_func(f)
    for i=1,#t,1 do
      local r = f(i, t[i])
      if r ~= nil then return r end
    end
  end
*/

```

```

#define LJLIB_LUA(name)

```

```

-- BYTECODE -- builtin:foreachi--1
0001      ISTYPE      0  12
0002      ISTYPE      1   9
0003      KSHORT      2   1
0004      LEN         3   0
0005      KSHORT      4   1
0006      FORI        2 => 0015
0007 => MOV          6   1
0008      MOV         8   5
0009      TGETR       9   0   5
0010      CALL        6   2   3
0011      ISEQP       6   0
0012      JMP         7 => 0014
0013      RET1        6   2
0014 => FORL         2 => 0007
0015 => RET0         0   1

```

```
| .ffunc_1 type  
|   mov RC, [BASE]  
|   sar RC, 47  
|   mov RBd, LJ_TISNUM  
|   cmp RCd, RBd  
|   cmovb RCd, RBd  
|   not RCd  
|   mov CFUNC:RB, [BASE-16]  
|   cleartp CFUNC:RB  
|   mov STR:RC, [CFUNC:RB+RC*8+((char *)(&((GCfuncC *)0)->upvalue))]  
|   mov PC, [BASE-8]  
|   settp STR:RC, LJ_TSTR  
|   mov [BASE-16], STR:RC  
|   jmp ->fff_res1
```

```
LJLIB_ASM(rawget)    LJLIB_REC(.)
{
    lj_lib_checktab(L, 1);
    lj_lib_checkany(L, 2);
    return FFH_UNREACHABLE;
}
```

```
typedef struct GCfunc {
    GCobj* nextgc;
    uint8_t marked;
    uint8_t gct;
    uint8_t ffid;
    uint8_t nupvalues;
    GCtab* env;
    GCobj* gclist;
    BCIns* pc;
    union {
        struct {
            lua_CFunction f;
            TValue upvalue[1];
        } c;
        struct {
            GCupval* uvptr[1];
        } l;
    };
} GCfunc;
```

```
typedef struct GCfunc {  
  
    uint8_t ffid;  
    uint8_t nupvalues;  
    GCtab* env;  
  
    BCIns* pc;  
    union {  
        struct {  
            lua_CFunction f;  
            TValue upvalue[1];  
        } c;  
        struct {  
            GCupval* uvptr[1];  
        } l;  
    };  
} GCfunc;
```



Function Kind	ffid	pc	opcode at *pc	#pc	f
Lua	0	GCproto* + 1	BC_[IJ]?FUNC[FV]	2+	N/A
C (std lib)	>= 1	&global_State:: bc_cfunc_int	BC_FUNCC	1	Function
C (3 <sup>rd</sup> party)	1	&global_State:: bc_cfunc_ext	BC_FUNCC or BC_FUNCCW	1	Function
Assembly	>= 2	&GG_State::bcff[i]	BC__MAX + i	1	Fallback

```

LJLIB_LUA(table_foreachi) /*
  function(t, f)
    CHECK_tab(t)
    CHECK_func(f)
    for i=1,#t,1 do
      local r = f(i, t[i])
      if r ~= nil then return r end
    end
  end
*/

```

```

#define LJLIB_LUA(name)

```

```

-- BYTECODE -- builtin:foreachi--1
0001      ISTYPE      0  12
0002      ISTYPE      1   9
0003      KSHORT      2   1
0004      LEN         3   0
0005      KSHORT      4   1
0006      FORI        2 => 0015
0007 => MOV          6   1
0008      MOV         8   5
0009      TGETR       9   0   5
0010      CALL        6   2   3
0011      ISEQP       6   0
0012      JMP         7 => 0014
0013      RET1        6   2
0014 => FORL         2 => 0007
0015 => RET0         0   1

```

```

LJLIB_LUA(table_foreachi) /*
  function(t, f)
    CHECK_tab(t)
    CHECK_func(f)
    for i=1,#t,1 do
      local r = f(i, t[i])
      if r ~= nil then return r end
    end
  end
*/

```

```

#define LJLIB_LUA(name)

```

```

-- BYTECODE -- builtin:foreach-1
0000      FUNCF      10
0001      ISTYPE    0   12
0002      ISTYPE    1   9
0003      KSHORT    2   1
0004      LEN       3   0
0005      KSHORT    4   1
0006      FORI      2 => 0015
0007 => MOV        6   1
0008      MOV       8   5
0009      TGETR     9   0   5
0010      CALL      6   2   3
0011      ISEQP     6   0
0012      JMP       7 => 0014
0013      RET1      6   2
0014 => FORL      2 => 0007
0015 => RET0      0   1

```

Function Kind	ffid	pc	opcode at *pc	#pc	f
Lua	0	GCproto* + 1	BC_[IJ]?FUNC[FV]	2+	N/A
C (std lib)	>= 1	&global_State:: bc_cfunc_int	BC_FUNCC	1	Function
C (3 <sup>rd</sup> party)	1	&global_State:: bc_cfunc_ext	BC_FUNCC or BC_FUNCCW	1	Function
Assembly	>= 2	&GG_State::bcff[i]	BC__MAX + i	1	Fallback

```
LJLIB_CF(coroutine_wrap)
{
    GCfunc *fn;
    lj_cf_coroutine_create(L);
    fn = lj_lib_pushcc(L, lj_ffh_coroutine_wrap_aux, FF_coroutine_wrap_aux, 1);
    setpc_wrap_aux(L, fn);
    return 1;
}
```

```
/* Fix the PC of wrap_aux. Really ugly workaround. */
static void setpc_wrap_aux(lua_State *L, GCfunc *fn)
{
    setmref(fn->c.pc, &L2GG(L)->bcfff[lj_lib_init_coroutine[1]+2]);
}
```

```
typedef struct GCfunc {  
  
    uint8_t ffid;  
    uint8_t nupvalues;  
    GCtab* env;  
  
    BCIns* pc;  
    union {  
        struct {  
            lua_CFunction f;  
            TValue upvalue[1];  
        } c;  
        struct {  
            GCupval* uvptr[1];  
        } l;  
    };  
} GCfunc;
```

```
print('Hello', 'World')
```

```
print('Hello', 'World')
```

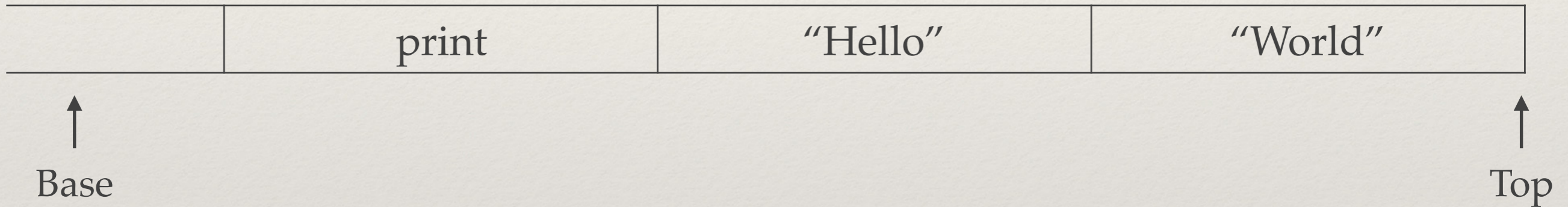
```
lua_getglobal(L, "print");  
lua_pushliteral(L, "Hello");  
lua_pushliteral(L, "World");  
lua_call(L, 2, 0);
```





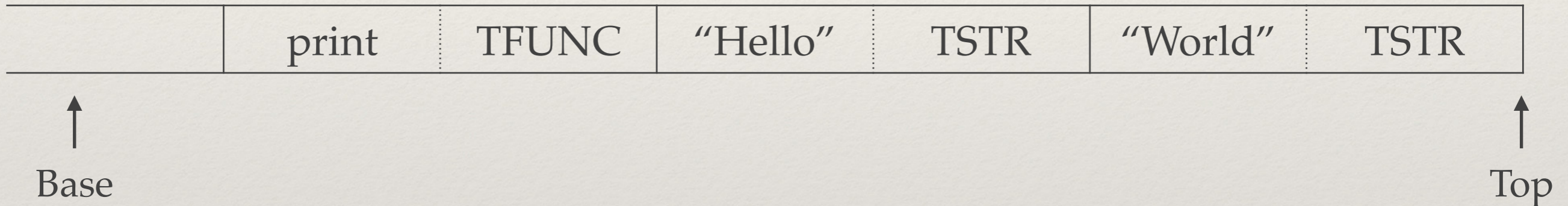
```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001    GGET      0    0      ; "print"  
lua_pushliteral(L, "World");    0002    KSTR     1    1      ; "Hello"  
lua_call(L, 2, 0);              0003    KSTR     2    2      ; "World"  
                                0004    CALL     0    1    3
```



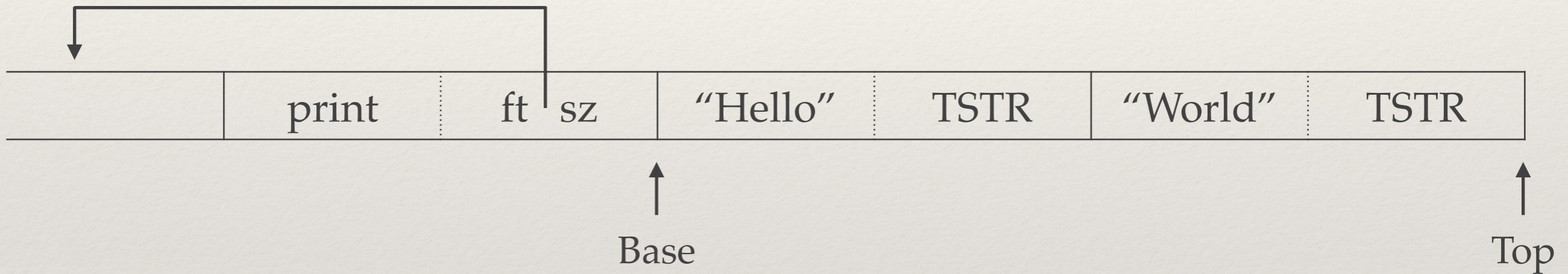
```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001      GGET      0      0      ; "print"  
lua_pushliteral(L, "World");    0002      KSTR     1      1      ; "Hello"  
lua_call(L, 2, 0);              0003      KSTR     2      2      ; "World"  
                                0004      CALL     0      1      3
```



```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001    GGET      0    0      ; "print"  
lua_pushliteral(L, "World");    0002    KSTR     1    1      ; "Hello"  
lua_call(L, 2, 0);              0003    KSTR     2    2      ; "World"  
                                0004    CALL     0    1    3
```



```
case BC_CALL:
|  movzx RC, RCL
|  mov LFUNC:RB, [BASE+RA*8]
|  cmp dword [BASE+RA*8+4], LJ_TFUNC
|  jne ->vmeta_call_ra
|  lea BASE, [BASE+RA*8+8]
|  mov [BASE-4], PC
|  mov PC, LFUNC:RB->pc
|  mov RA, [PC]
|  movzx OP, RAL
|  movzx RA, RAH
|  add PC, 4
|  jmp aword [DISPATCH+OP*4]
```

case BC_CALL:	
movzx RC, RCL	Prepare RB and RC for the
mov LFUNC:RB, [BASE+RA*8]	upcoming FUNC opcode
cmp dword [BASE+RA*8+4], LJ_TFUNC	Check that we're calling
jne ->vmeta_call_ra	a function
lea BASE, [BASE+RA*8+8]	Shift up base and replace
mov [BASE-4], PC	LJ_TFUNC by ft/sz
mov PC, LFUNC:RB->pc	Fetch *pc and decode it
mov RA, [PC]	into OP and RA (parts B
movzx OP, RAL	and C of *pc are ignored)
movzx RA, RAH	
add PC, 4	
jmp aword [DISPATCH+OP*4]	Jump to a FUNC opcode

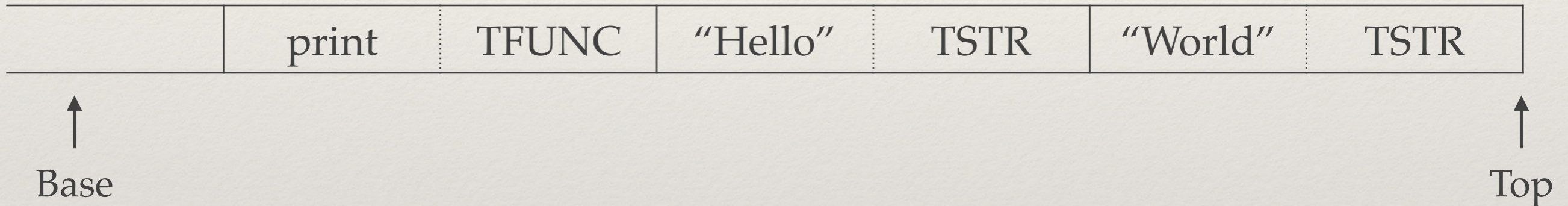
```
case BC_FUNCF:  
| hotcall RB  
case BC_IFUNCF:  
| mov L:RB, SAVE_L  
| lea RA, [BASE+RA*8]  
| cmp RA, L:RB->maxstack  
| ja ->vm_growstack_f  
| mov KBASE, [PC-4+PC2PROTO(k)]  
| movzx RA, byte [PC-4+PC2PROTO(numparams)]  
| cmp RC, RA  
| jbe >3  
| ins_next
```

case BC_FUNCF:	Maybe trigger JIT-compilation of callee
hotcall RB	(RB was set by BC_CALL)
case BC_IFUNCF:	
mov L:RB, SAVE_L	
lea RA, [BASE+RA*8]	Check for stack overflow
cmp RA, L:RB->maxstack	
ja ->vm_growstack_f	
mov KBASE, [PC-4+PC2PROTO(k)]	Load constants array
movzx RA, byte [PC-4+PC2PROTO(numparams)]	Check for missing args
cmp RC, RA	(RC was set by BC_CALL)
jbe >3	(code at label 3 omitted)
ins_next	Execute next instruction



```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001    GGET      0    0      ; "print"  
lua_pushliteral(L, "World");    0002    KSTR     1    1      ; "Hello"  
lua_call(L, 2, 0);             0003    KSTR     2    2      ; "World"  
                                0004    CALL     0    1    3
```



```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001    GGET      0    0      ; "print"  
lua_pushliteral(L, "World");    0002    KSTR     1    1      ; "Hello"  
lua_call(L, 2, 0);              0003    KSTR     2    2      ; "World"  
                                0004    CALL     0    1    3
```



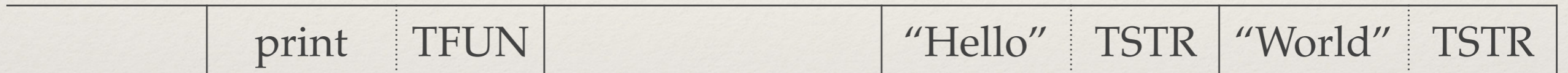
↑  
Base

↑  
Top

```
LUA_API void lua_call(lua_State *L, int nargs, int nresults)
{
    TValue *o = L->top, *base = L->top - nargs;
#ifdef LJ_FR2
    L->top++;
    for (; o > base; o--) copyTV(L, o, o-1);
    setnilV(o);
    base++;
#endif
    lj_vm_call(L, base, nresults+1);
}
```

```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001      GGET      0      0      ; "print"  
lua_pushliteral(L, "World");    0002      KSTR     1      1      ; "Hello"  
lua_call(L, 2, 0);              0003      KSTR     2      2      ; "World"  
                                0004      CALL     0      1      3
```

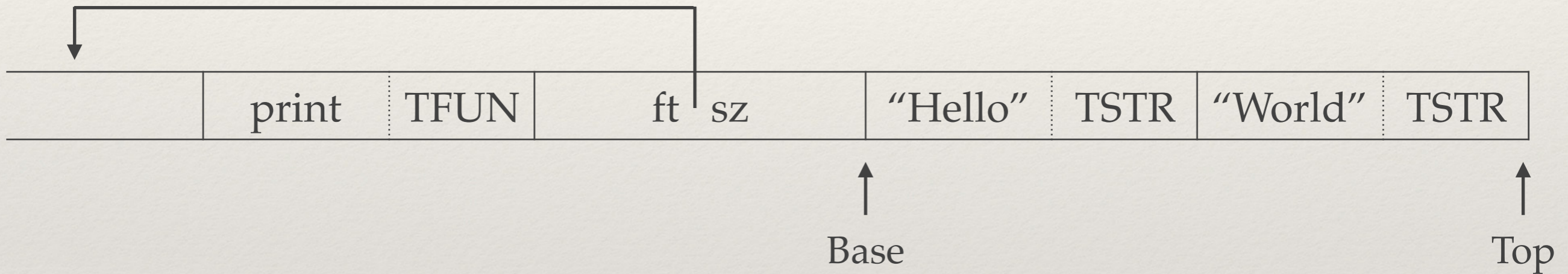


↑  
Base

↑  
Top

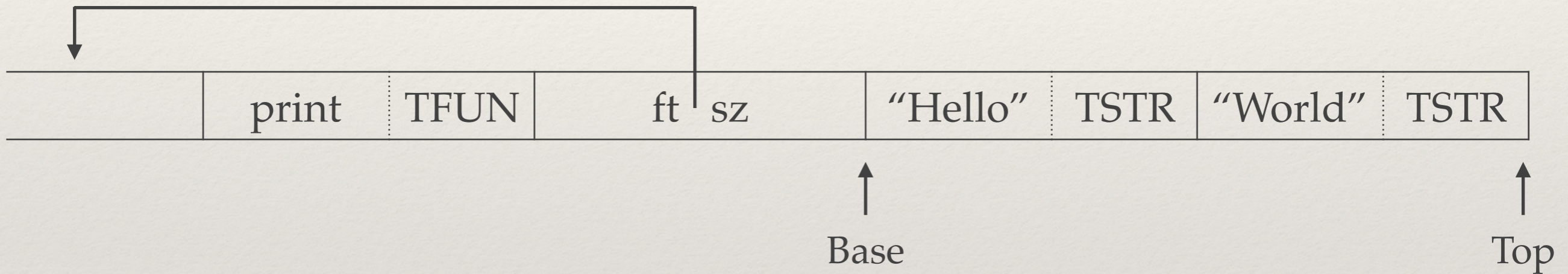
```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001      GGET      0      0      ; "print"  
lua_pushliteral(L, "World");    0002      KSTR     1      1      ; "Hello"  
lua_call(L, 2, 0);              0003      KSTR     2      2      ; "World"  
                                0004      CALL     0      1      3
```



```
print('Hello', 'World')
```

```
lua_getglobal(L, "print");      -- BYTECODE -- "print('Hello', 'World')"  
lua_pushliteral(L, "Hello");    0001      GGET      0  0      ; "print"  
lua_pushliteral(L, "World");    0002      KSTR     12  1      ; "Hello"  
lua_call(L, 2, 0);              0003      KSTR     23  2      ; "World"  
                                0004      CALL     0  1  3
```



```
static const uint8_t libbc_code[] = {
0,1,2,0,0,1,2,24,1,0,0,76,1,2,0,241,135,158,166,3,220,203,178,130,4,0,1,2,0,
0,1,2,24,1,0,0,76,1,2,0,243,244,148,165,20,198,190,199,252,3,0,1,2,0,0,0,3,
16,0,5,0,21,1,0,0,76,1,2,0,0,2,9,0,0,0,15,16,0,12,0,16,1,9,0,41,2,1,0,21,3,
0,0,41,4,1,0,77,2,8,128,18,6,1,0,18,7,5,0,59,8,5,0,66,6,3,2,10,6,0,0,88,7,1,
128,76,6,2,0,79,2,248,127,75,0,1,0,0,2,10,0,0,0,16,16,0,12,0,16,1,9,0,43,2,
0,0,18,3,0,0,41,4,0,0,88,5,7,128,18,7,1,0,18,8,5,0,18,9,6,0,66,7,3,2,10,7,0,
0,88,8,1,128,76,7,2,0,70,5,3,3,82,5,247,127,75,0,1,0,0,1,2,0,0,0,3,16,0,12,
0,21,1,0,0,76,1,2,0,0,2,10,0,0,2,30,16,0,12,0,21,2,0,0,11,1,0,0,88,3,7,128,
8,2,0,0,88,3,23,128,59,3,2,0,43,4,0,0,64,4,2,0,76,3,2,0,88,3,18,128,16,1,14,
0,41,3,1,0,3,3,1,0,88,3,14,128,3,1,2,0,88,3,12,128,59,3,1,0,22,4,1,1,18,5,2,
0,41,6,1,0,77,4,4,128,23,8,1,7,59,9,7,0,64,9,8,0,79,4,252,127,43,4,0,0,64,4,
2,0,76,3,2,0,75,0,1,0,0,2,0
};
```

```
static const uint8_t libbc_code[] = {
#if LJ_FR2
0,1,2,0,0,1,2,24,1,0,0,76,1,2,0,241,135,158,166,3,220,203,178,130,4,0,1,2,0,
0,1,2,24,1,0,0,76,1,2,0,243,244,148,165,20,198,190,199,252,3,0,1,2,0,0,0,3,
16,0,5,0,21,1,0,0,76,1,2,0,0,2,10,0,0,0,15,16,0,12,0,16,1,9,0,41,2,1,0,21,3,
0,0,41,4,1,0,77,2,8,128,18,6,1,0,18,8,5,0,59,9,5,0,66,6,3,2,10,6,0,0,88,7,1,
128,76,6,2,0,79,2,248,127,75,0,1,0,0,2,11,0,0,0,16,16,0,12,0,16,1,9,0,43,2,
0,0,18,3,0,0,41,4,0,0,88,5,7,128,18,7,1,0,18,9,5,0,18,10,6,0,66,7,3,2,10,7,
0,0,88,8,1,128,76,7,2,0,70,5,3,3,82,5,247,127,75,0,1,0,0,1,2,0,0,0,3,16,0,12,
0,21,1,0,0,76,1,2,0,0,2,10,0,0,2,30,16,0,12,0,21,2,0,0,11,1,0,0,88,3,7,128,
8,2,0,0,88,3,23,128,59,3,2,0,43,4,0,0,64,4,2,0,76,3,2,0,88,3,18,128,16,1,14,
0,41,3,1,0,3,3,1,0,88,3,14,128,3,1,2,0,88,3,12,128,59,3,1,0,22,4,1,1,18,5,2,
0,41,6,1,0,77,4,4,128,23,8,1,7,59,9,7,0,64,9,8,0,79,4,252,127,43,4,0,0,64,4,
2,0,76,3,2,0,75,0,1,0,0,2,0
#else
0,1,2,0,0,1,2,24,1,0,0,76,1,2,0,241,135,158,166,3,220,203,178,130,4,0,1,2,0,
0,1,2,24,1,0,0,76,1,2,0,243,244,148,165,20,198,190,199,252,3,0,1,2,0,0,0,3,
16,0,5,0,21,1,0,0,76,1,2,0,0,2,9,0,0,0,15,16,0,12,0,16,1,9,0,41,2,1,0,21,3,
0,0,41,4,1,0,77,2,8,128,18,6,1,0,18,7,5,0,59,8,5,0,66,6,3,2,10,6,0,0,88,7,1,
128,76,6,2,0,79,2,248,127,75,0,1,0,0,2,10,0,0,0,16,16,0,12,0,16,1,9,0,43,2,
0,0,18,3,0,0,41,4,0,0,88,5,7,128,18,7,1,0,18,8,5,0,18,9,6,0,66,7,3,2,10,7,0,
0,88,8,1,128,76,7,2,0,70,5,3,3,82,5,247,127,75,0,1,0,0,1,2,0,0,0,3,16,0,12,
0,21,1,0,0,76,1,2,0,0,2,10,0,0,2,30,16,0,12,0,21,2,0,0,11,1,0,0,88,3,7,128,
8,2,0,0,88,3,23,128,59,3,2,0,43,4,0,0,64,4,2,0,76,3,2,0,88,3,18,128,16,1,14,
0,41,3,1,0,3,3,1,0,88,3,14,128,3,1,2,0,88,3,12,128,59,3,1,0,22,4,1,1,18,5,2,
0,41,6,1,0,77,4,4,128,23,8,1,7,59,9,7,0,64,9,8,0,79,4,252,127,43,4,0,0,64,4,
2,0,76,3,2,0,75,0,1,0,0,2,0
#endif
};
```



```
git clone http://github.com/LuaJIT/LuaJIT
cd LuaJIT
git checkout v2.1
make -j
sudo make install
```

```
git clone http://github.com/LuaJIT/LuaJIT
cd LuaJIT
git checkout v2.1
make -j CFLAGS='-DLUAJIT_ENABLE_GC64'
sudo make install
```

```
git clone http://github.com/corsix/LuaJIT
cd LuaJIT
git checkout newgc
make -j CFLAGS='-DLUAJIT_ENABLE_GC64'
sudo make install
```