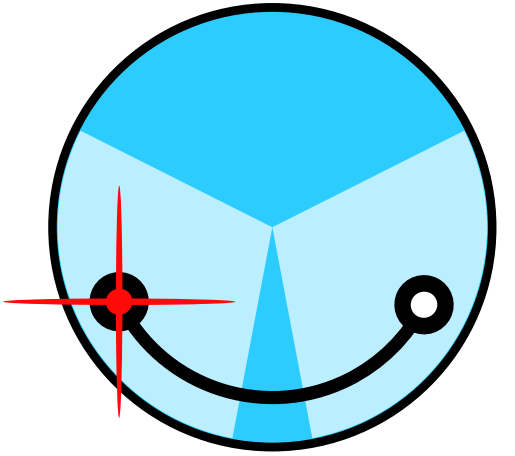
Taint tracking

useful, but slow as hell

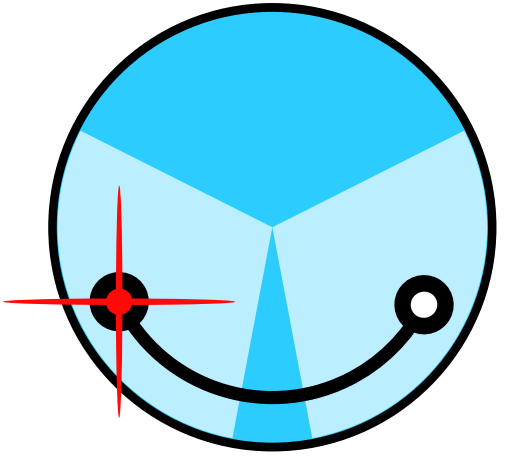
Is this slowness fundamental?



minemu

fast emulator
memory layout
use SSE registers to hold taint

Is this slowness fundamental?



minemu

- fast emulator
  memory layout
  use SSE registers to hold taint

Emulator

- process-level emulator

Emulator

process-level emulator

- fast x86 -> x86 jit compiler

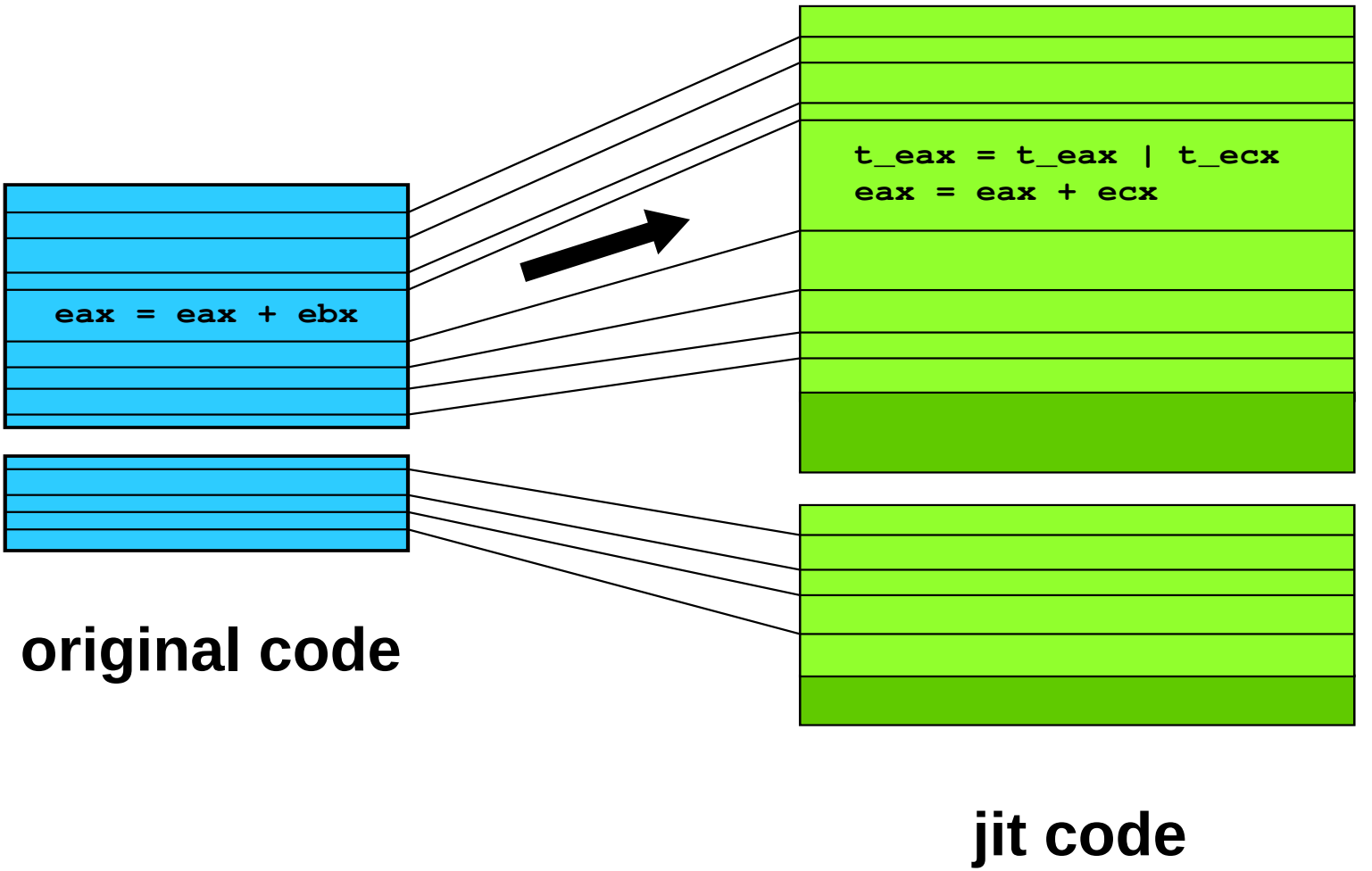
Emulator

process-level emulator

fast x86 -> x86 jit compiler

- keeps register state the same

Emulator



Emulator

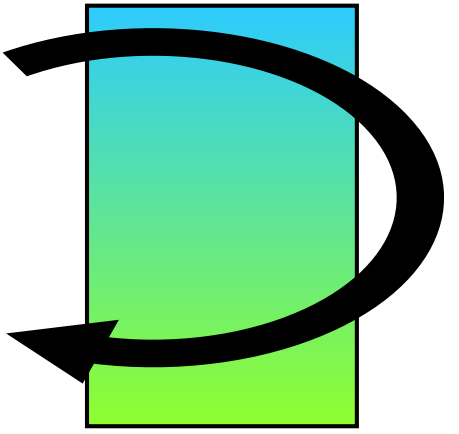
process-level emulator

fast x86 -> x86 jit compiler

keeps register state the same

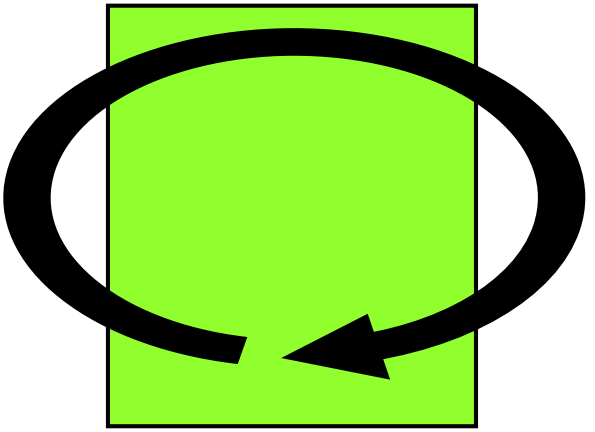
- translates big chunks of code all at once

Emulator

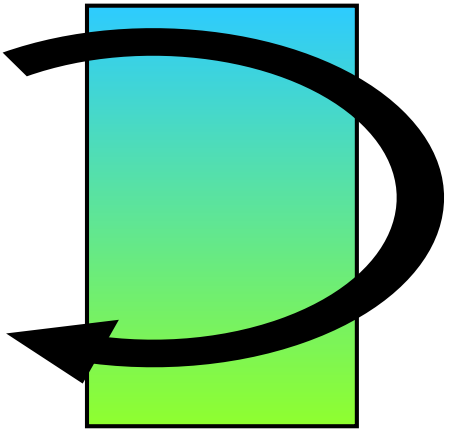


compile jit code

Emulator

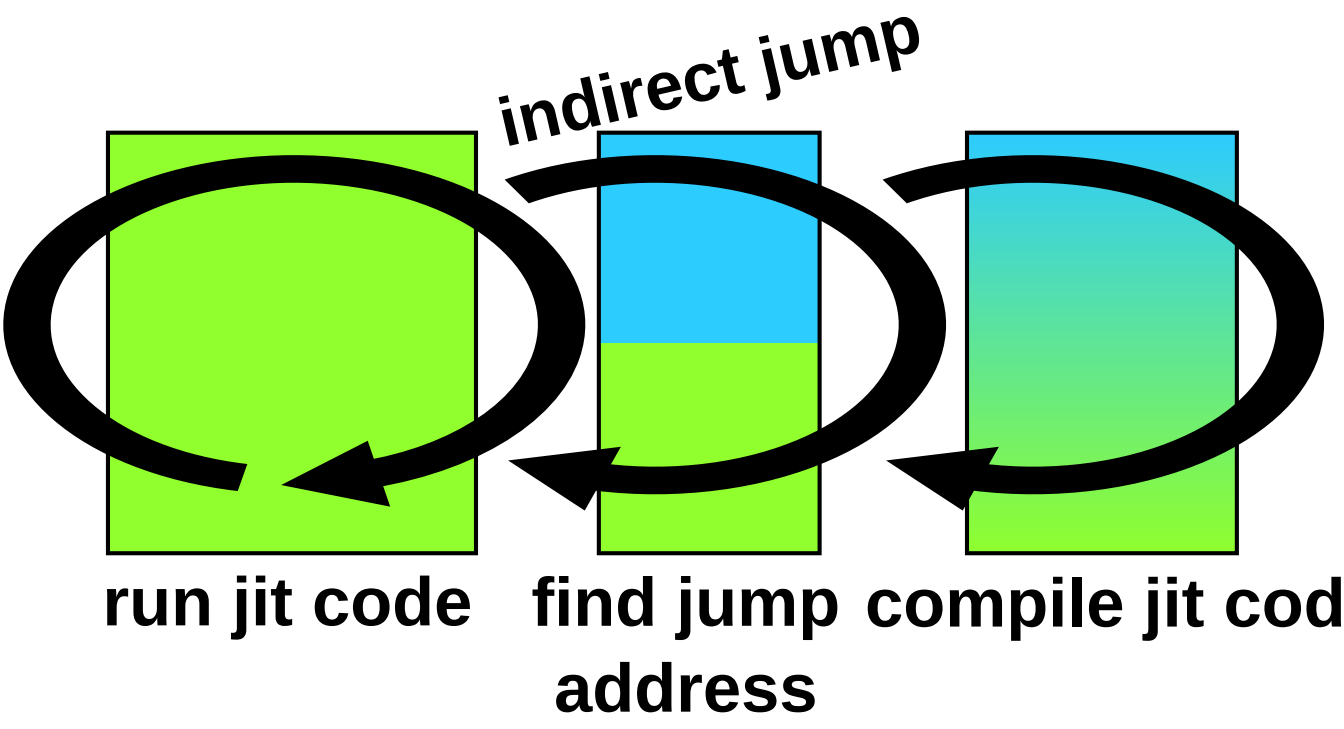


run jit code

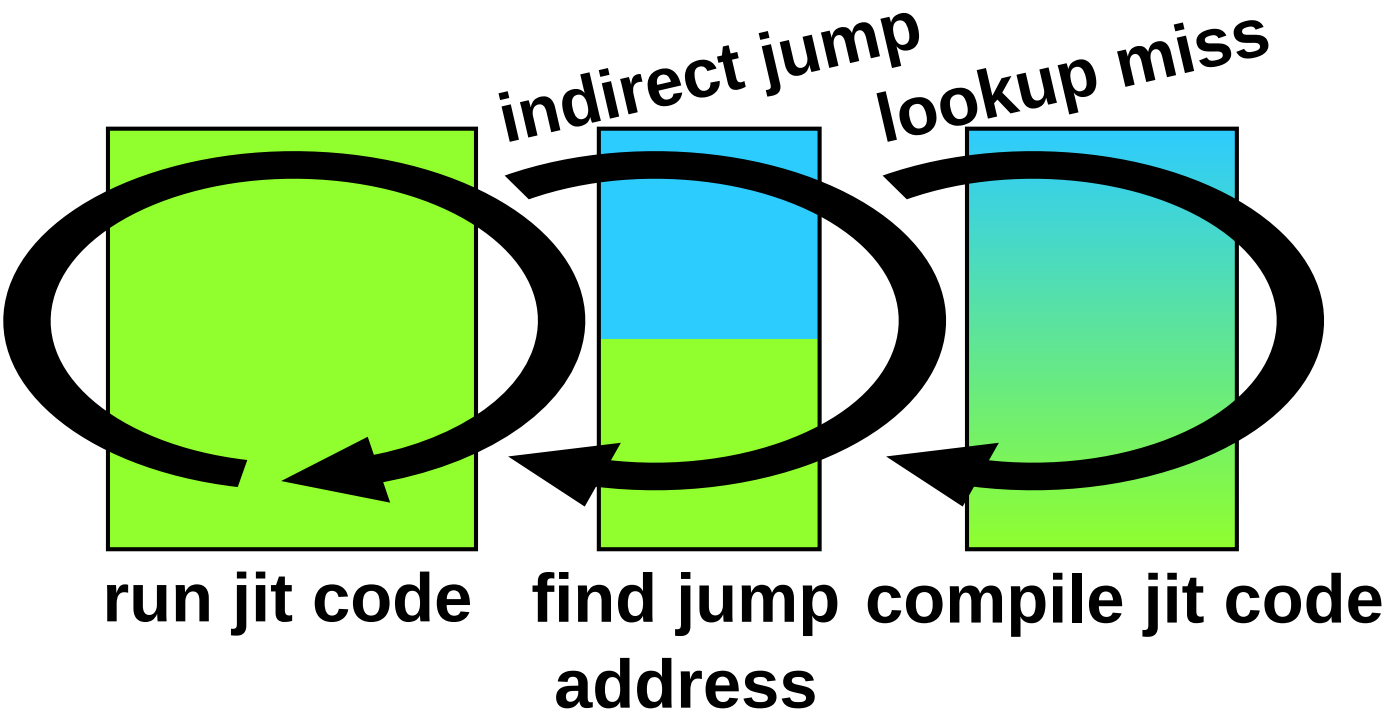


compile jit code

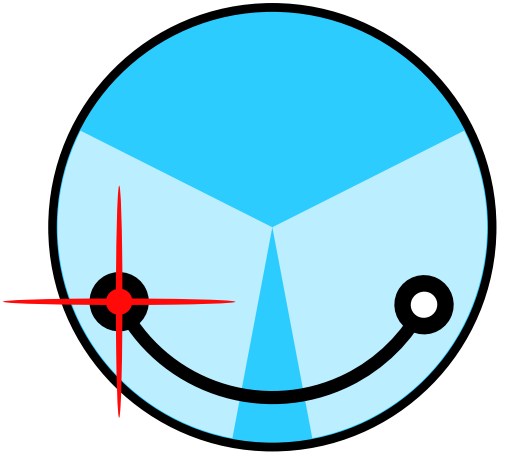
Emulator


indirect jump
run jit code
find jump address
compile jit code

Emulator

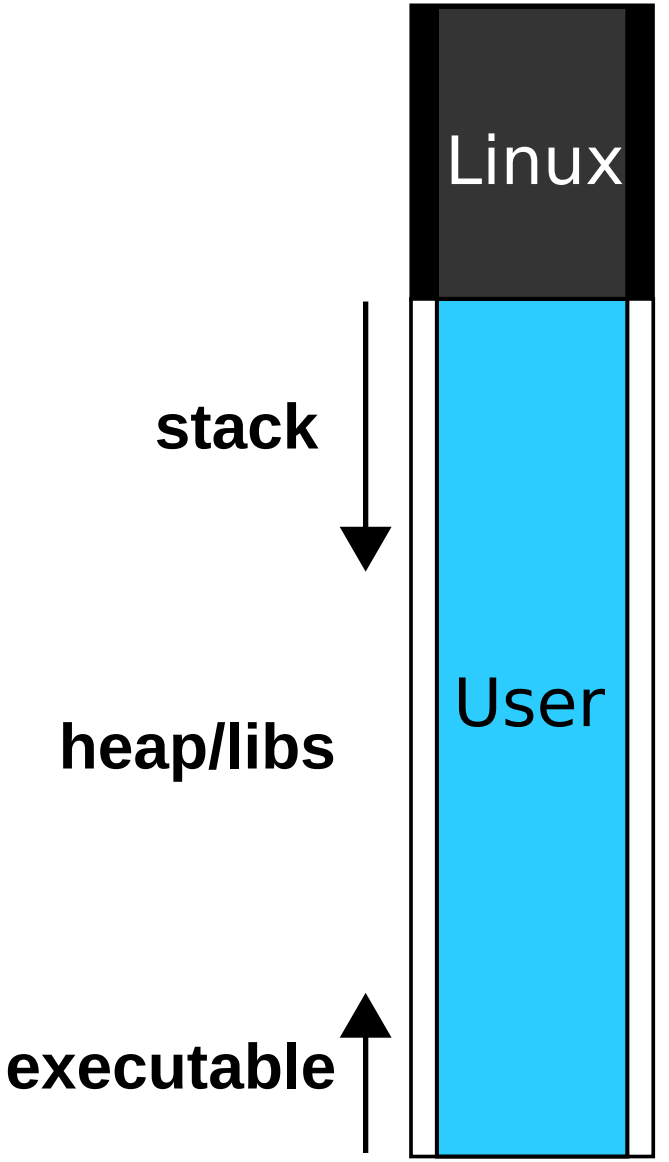

indirect jump
lookup miss
run jit code
find jump address
compile jit code

Is this slowness fundamental?

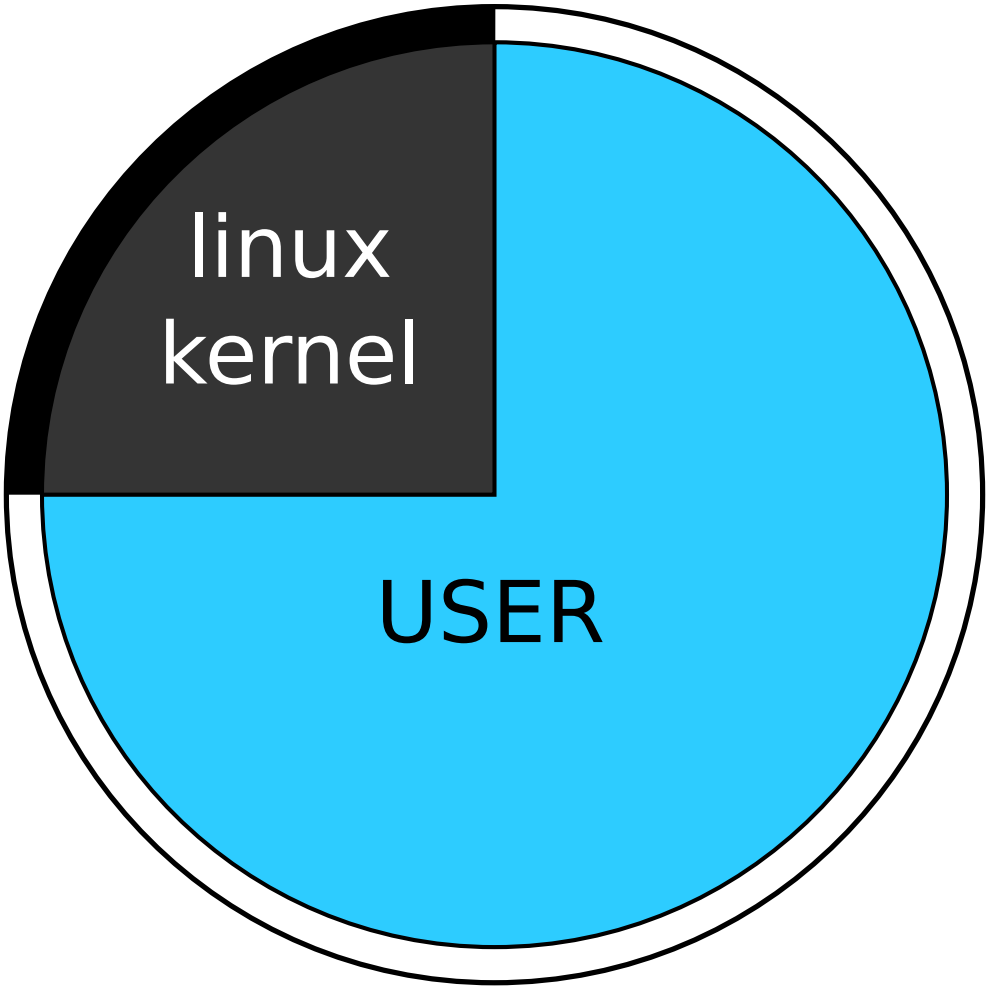


minemu

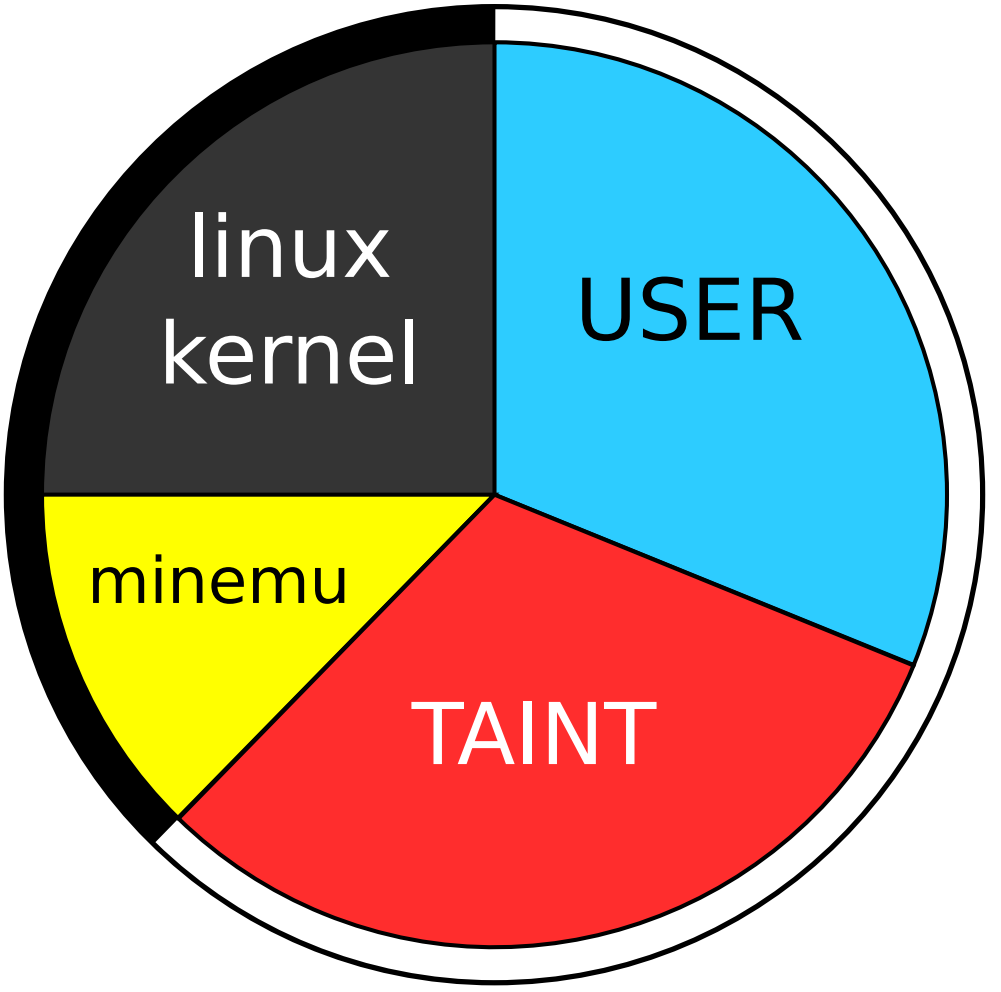
- fast emulator
- memory layout
- use SSE registers to hold taint


Linux
stack
User
heap/libs
executable

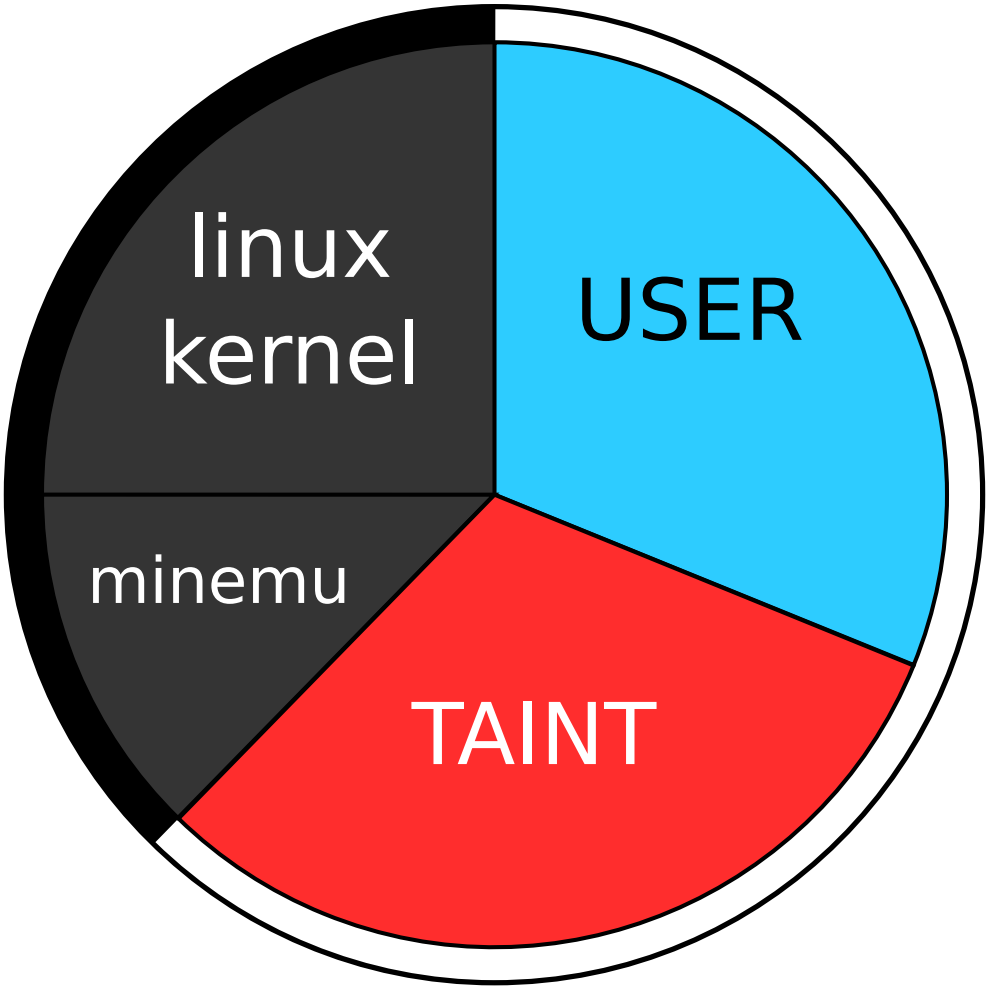
Memory layout (linux)



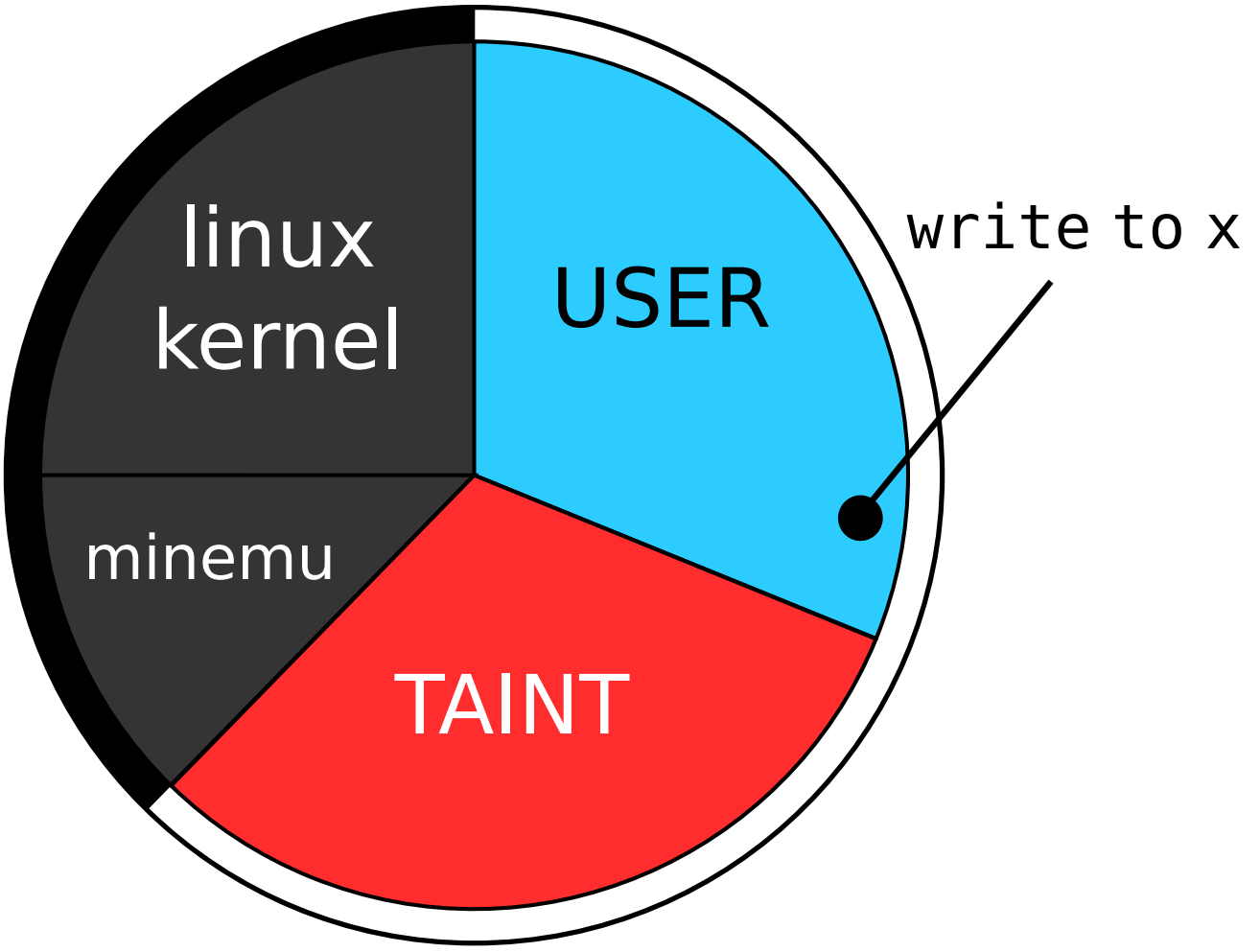
Memory layout (minemu)



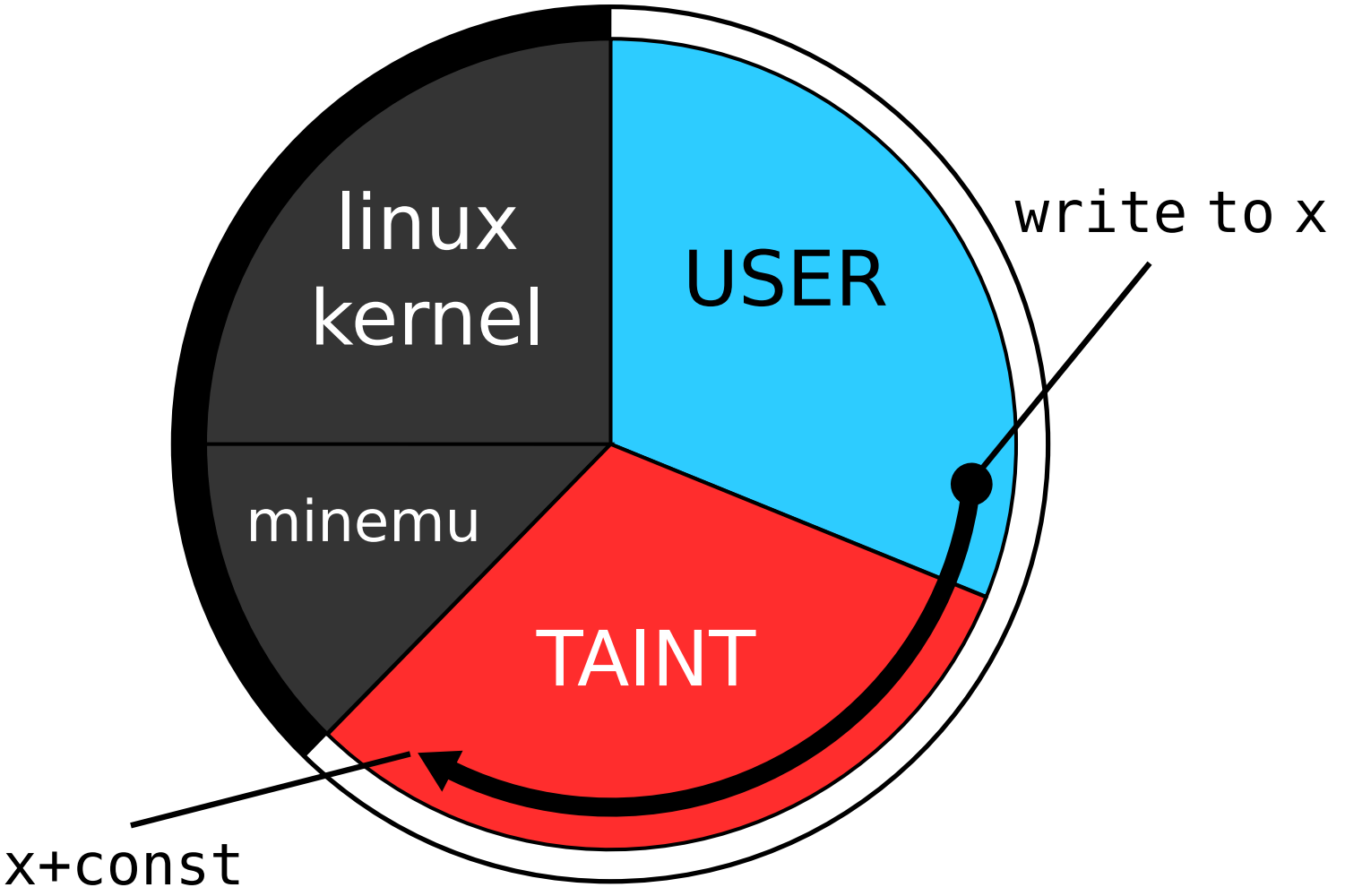
Memory layout (minemu)



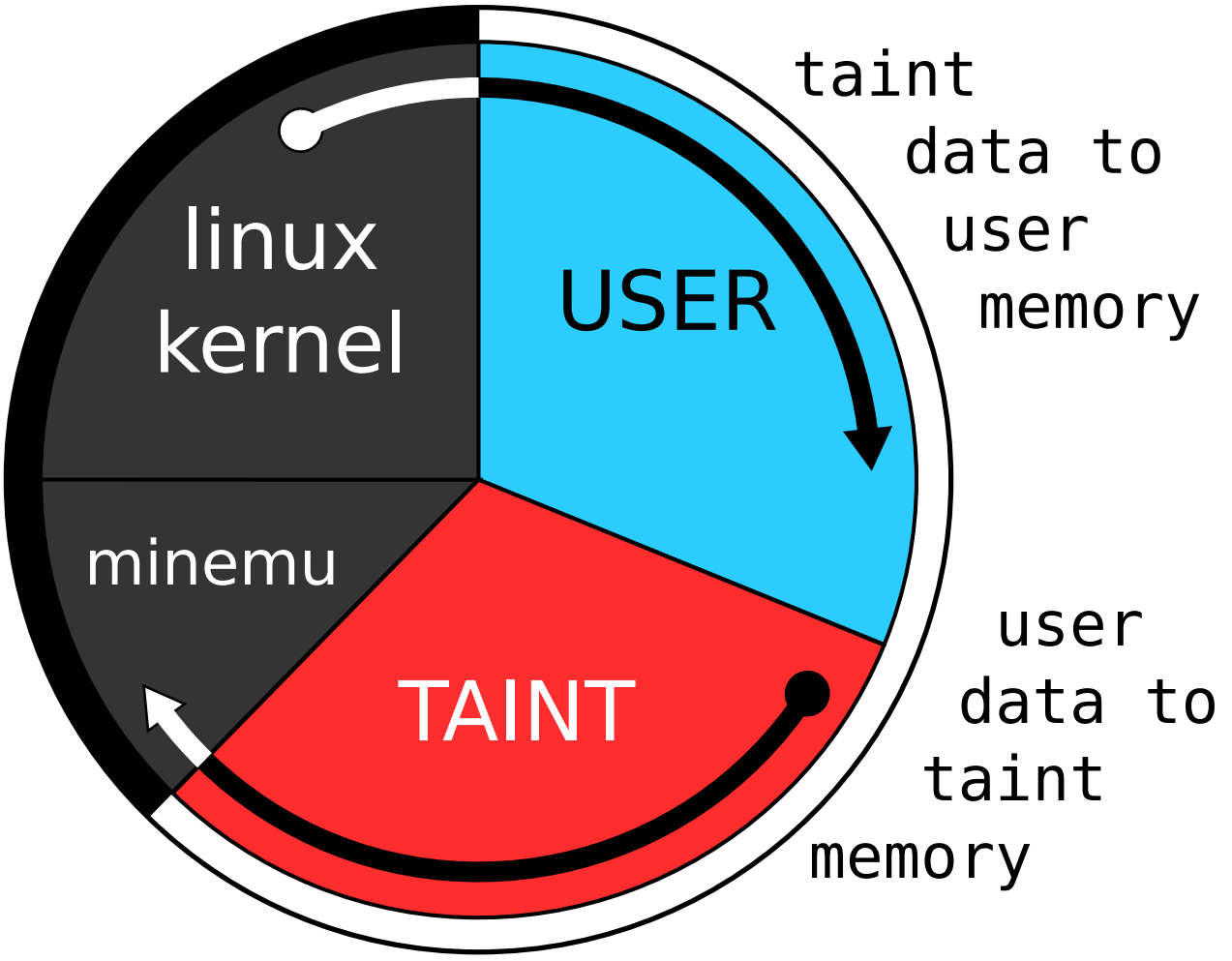
Memory layout (minemu)



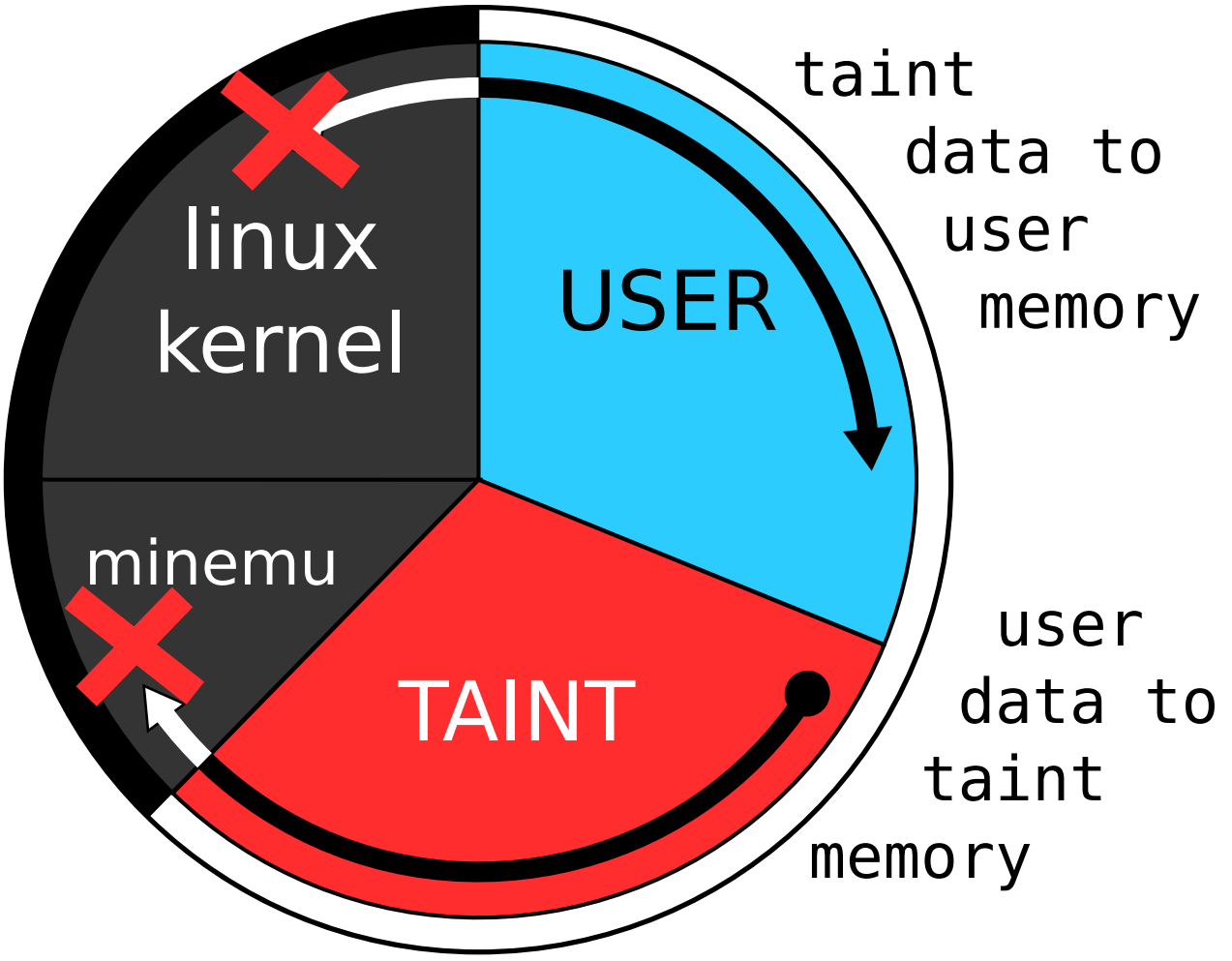
Memory layout (minemu)



Memory layout (minemu)



Memory layout (minemu)



Addressing shadow memory

```
mov EAX, (EDX)
```

Addressing shadow memory

```
mov EAX, (EDX)

address:

    EDX
```

Addressing shadow memory

```
mov EAX, (EDX)

address:

    EDX

taint:

    EDX+const
```

Addressing shadow memory

```
mov EAX, (EDX+EBX*4)
```

Addressing shadow memory

```
mov EAX, (EDX+EBX*4)

address:

    EDX+EBX*4
```

Addressing shadow memory

```
mov EAX, (EDX+EBX*4)

address:

    EDX+EBX*4

taint:

    EDX+EBX*4+const
```

Addressing shadow memory

```
push ESI
```

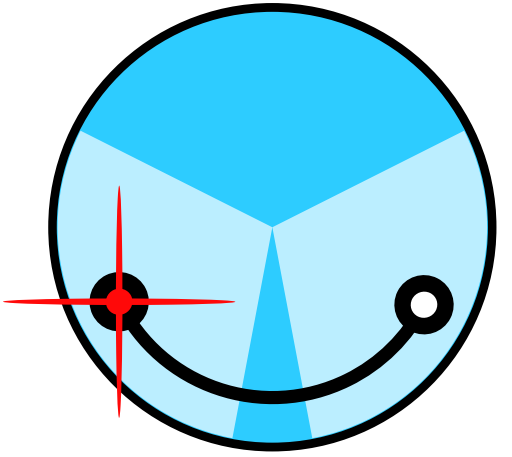
Addressing shadow memory

```
push ESI

address:

    ESP
```

Addressing shadow memory

```
push ESI

address:

    ESP

taint:

    ESP+const
```

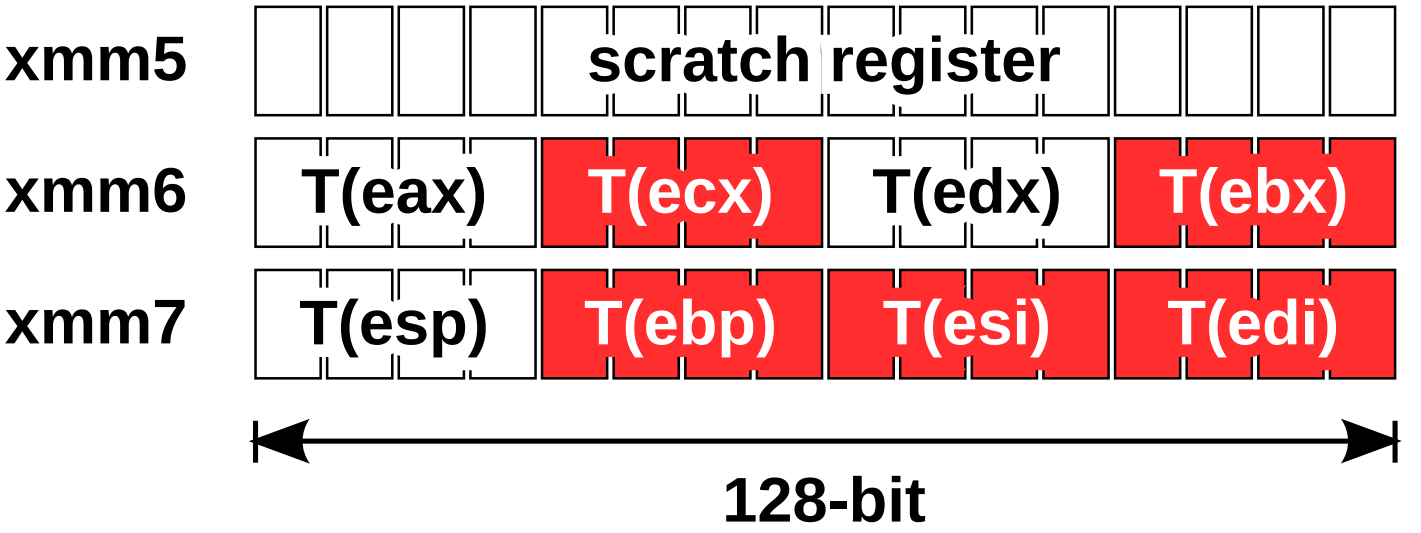
Is this slowness fundamental?



minemu

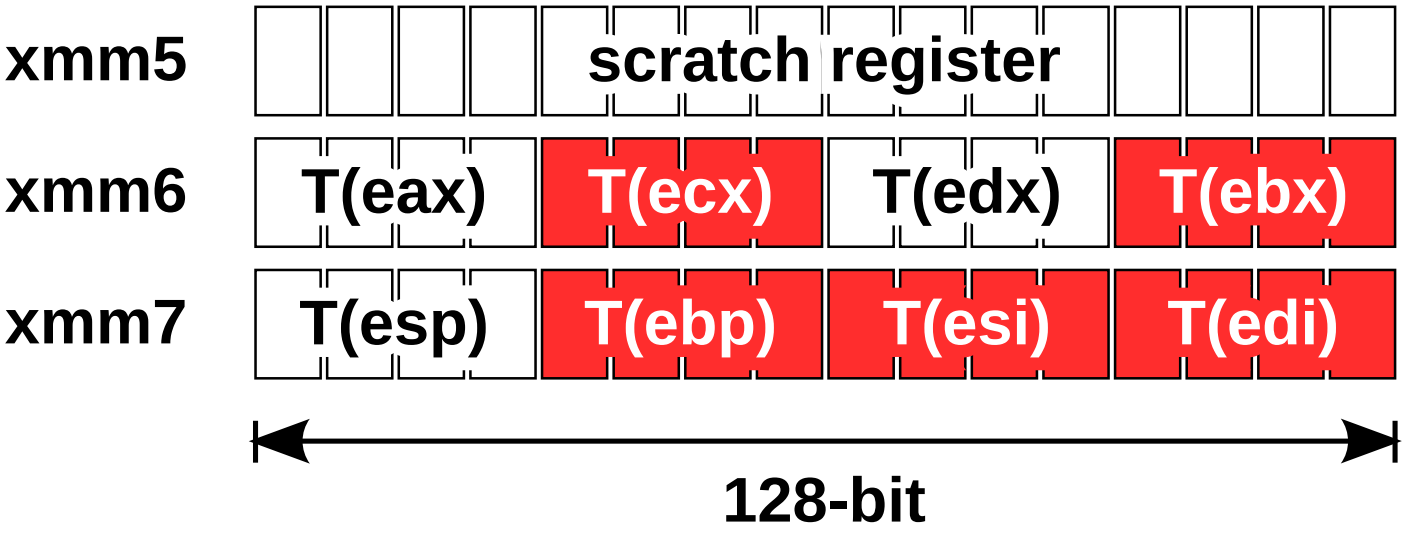
- fast emulator
- memory layout
- use SSE registers to hold taint

Taint propagation in SSE registers



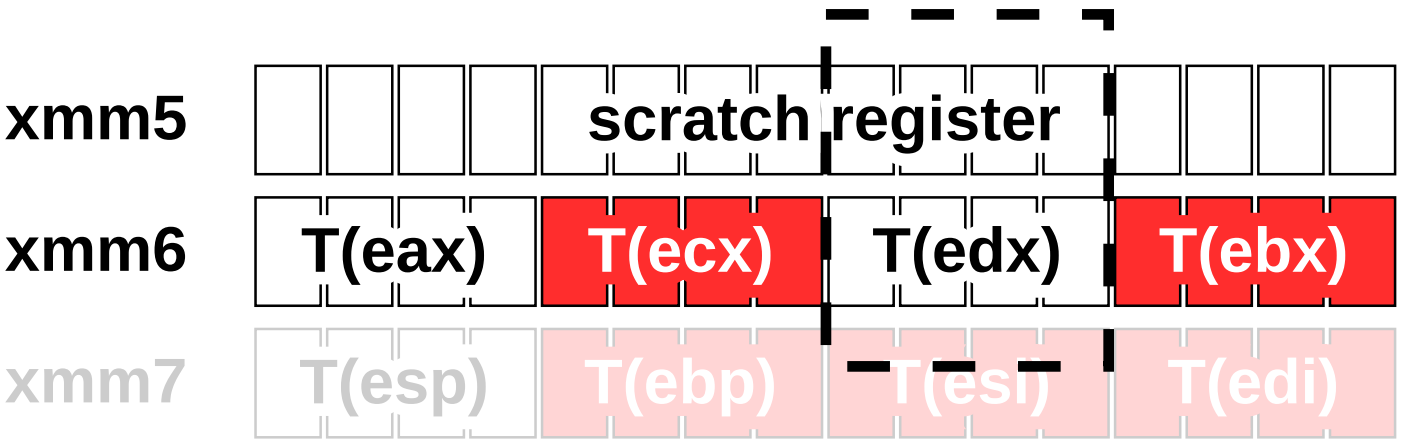
Taint propagation in SSE registers

```
add EDX, x
```



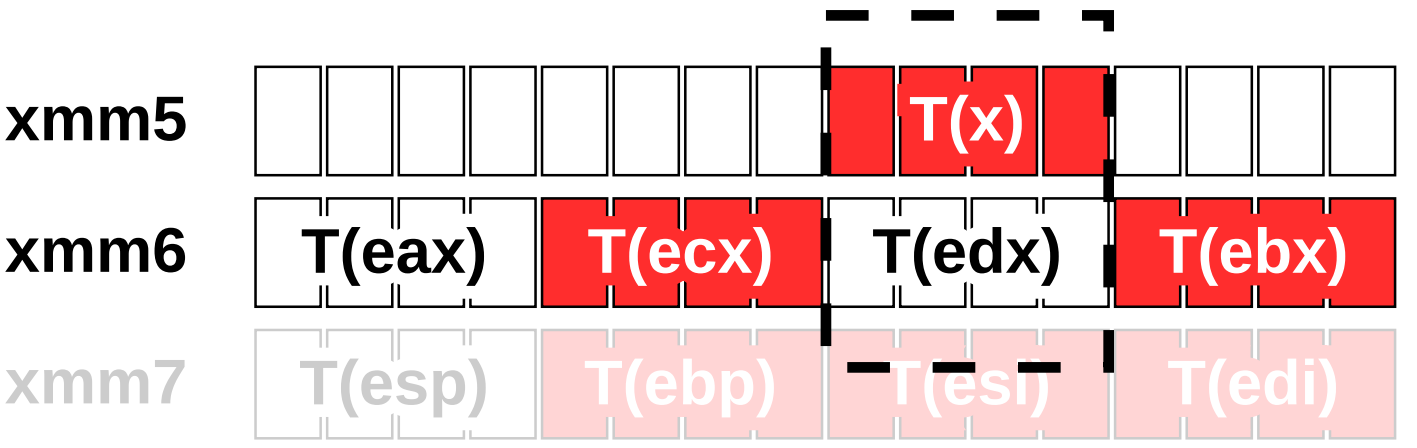
Taint propagation in SSE registers

```
add EDX, x
```



Taint propagation in SSE registers

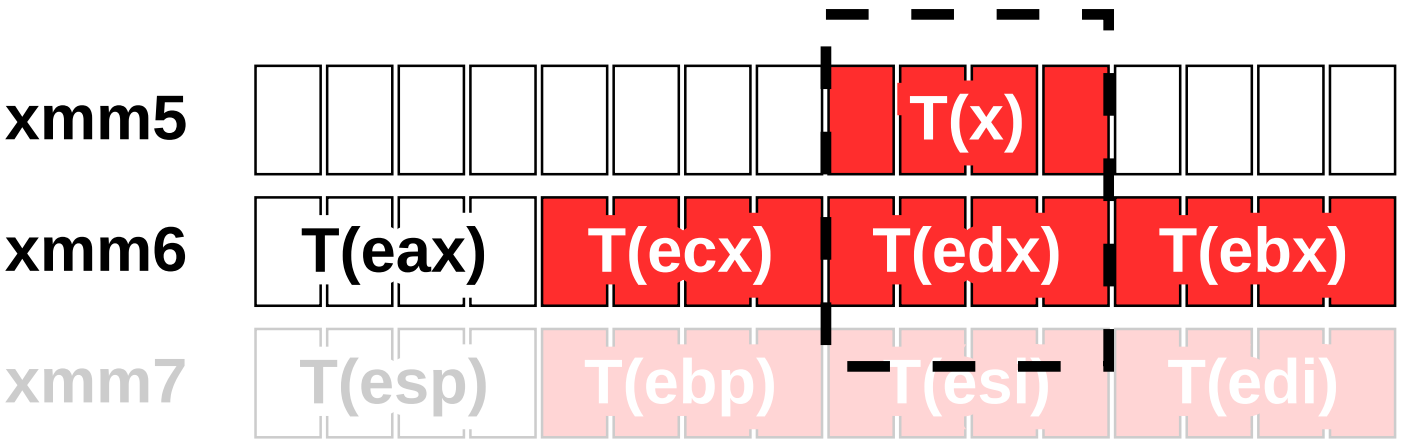
`add EDX, x`



`vector insert`

Taint propagation in SSE registers

`add EDX, x`

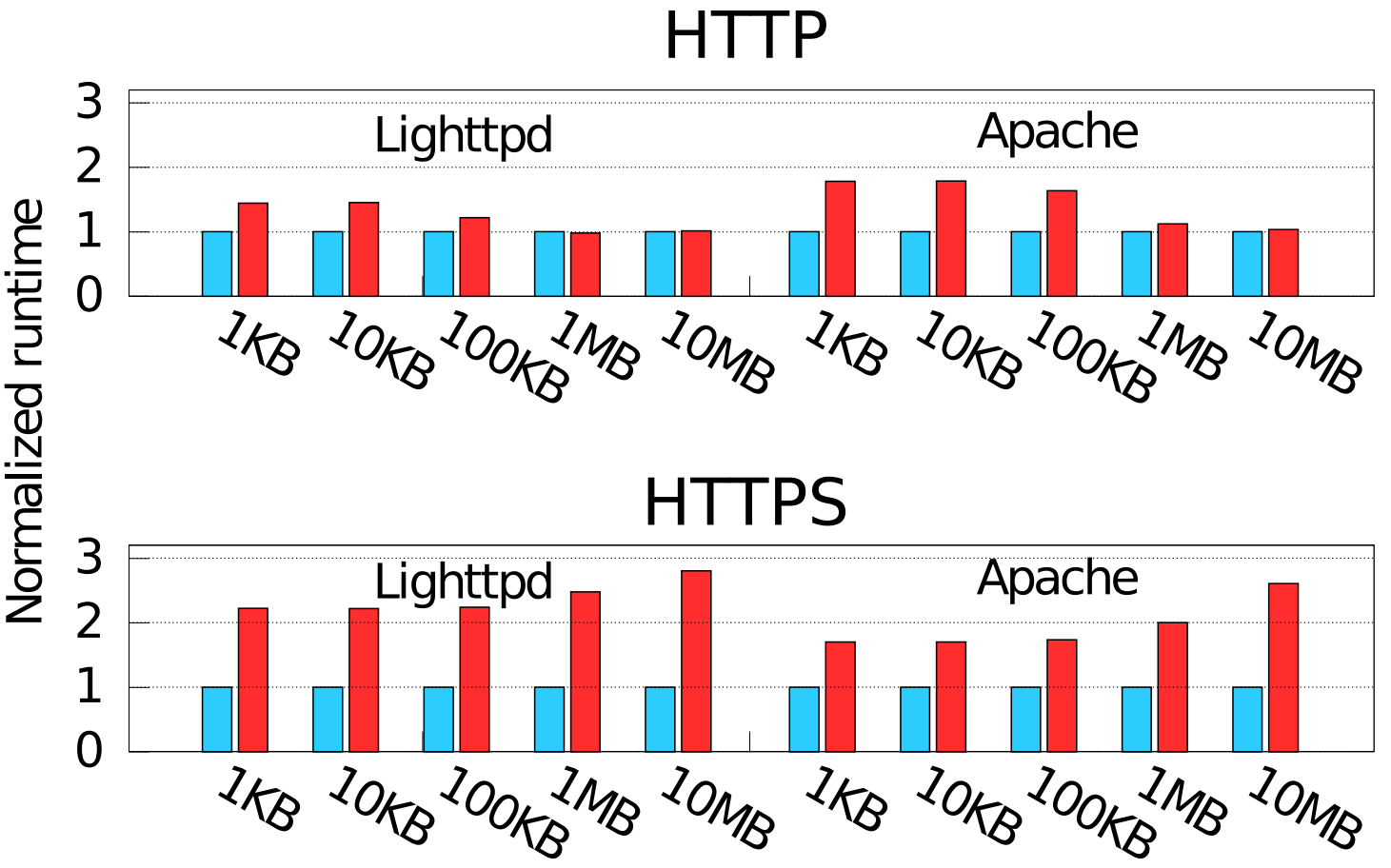


or

Effectiveness

Application	Type of vulnerability	Security advisory
Snort 2.4.0	Stack overflow	CVE-2005-3252
Cyrus imapd 2.3.2	Stack overflow	CVE-2006-2502
Samba 3.0.22	Heap overflow	CVE-2007-2446
Memcached 1.1.12	Heap overflow	CVE-2009-2415
Nginx 0.6.32	Buffer underrun	CVE-2009-2629
Proftpd 1.3.3a	Stack overflow	CVE-2010-4221
Samba 3.2.5	Heap overflow	CVE-2010-2063
Telnetd 1.6	Heap overflow	CVE-2011-4862
Ncompress 4.2.4	Stack overflow	CVE-2001-1413
Iwconfig V.26	Stack overflow	CVE-2003-0947
Aspell 0.50.5	Stack overflow	CVE-2004-0548
Htget 0.93	Stack overflow	CVE-2004-0852
Socat 1.4	Format string	CVE-2004-1484
Aeon 0.2a	Stack overflow	CVE-2005-1019
Exim 4.41	Stack overflow	EDB-ID#796
Htget 0.93	Stack overflow	
Tipxd 1.1.1	Format string	OSVDB-ID#12346

Performance



Performance