RUST CHINA CONF 2020

首届中国 Rust 开发者大会

2020.12.26–27 深圳

RUST CHINA CONF 2020

首届中国 Rust 开发者大会

2020.12.26–27 深圳

CRYPTAPE
秘猿科技

Internet 2.0 - A decentralized data and value network

Rust, RISC-V 和智能合约

Who am I?

Jiang Jinyang

Blockchain developer @ 秘猿科技

Website: http://justjjy.com

智能合约

Smart contract

Smart contract

区块链上的程序， 类似于数据库的存储过程

- 确定性
- 对安全性要求高（输入不可信！）
- 很难更新

Solidity

最流行的智能合约编程语言

solidity 1

```
mapping (address => uint256) public balanceOf;

// INSECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance */
    require(balanceOf[msg.sender] >= _value);
    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}

// SECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance and for overflows */
    require(balanceOf[msg.sender] >= _value && balanceOf[_to] + _value >= balanceOf[_to]);

    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}
```

https://consensys.github.io/smart-contract-best-practices/known_attacks/#integer-overflow-and-underflow

solidity 1

```
mapping (address => uint256) public balanceOf;

// INSECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance */
    require(balanceOf[msg.sender] >= _value);
    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}

// SECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance and for overflows */
    require(balanceOf[msg.sender] >= _value && balanceOf[_to] + _value >= balanceOf[_to]);

    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}
```

https://consensys.github.io/smart-contract-best-practices/known_attacks/#integer-overflow-and-underflow

solidity 2

```
function play(uint256 number) payable public {
    require(msg.value >= betPrice && number <= 10);

    Game game;
    game.player = msg.sender;
    game.number = number;
    gamesPlayed.push(game);
```

https://swcregistry.io/docs/SWC-109#crypto-roulettesol

solidity 2

```
function play(uint256 number) payable public {
    require(msg.value >= betPrice && number <= 10);

    Game game;
    game.player = msg.sender;
    game.number = number;
    gamesPlayed.push(game);
```

https://swcregistry.io/docs/SWC-109#crypto-roulettesol fixed in solidity 0.5

为什么不用通用编程语言?

VMs

EVM

- EVM 指令集
- Solidity, Vyper， ...

CKB-VM

- RISC-V 指令集
- C, Rust, ...
- 更贴近真实的机器指令集
- 标准化

EWASM

- WASM 指令集
- C, Rust, ...

CKB VM

- rv64imc (RV64I + Mutiplication/division + Compressed instructions extensions)
- no MMU
- Use syscalls to interact with the blockchain.

Rust on CKB-VM

no STD & lang items

- 使用 core 来替代 std，无法使用 std 中和 OS 有关的 modules
- 需要定义一些 lang items
- 需要自定义 global allocator 并通过 alloc crate 使用 Heap 相关的功能

Core crate

mem	Basic functions for dealing with memory.
num	Numeric traits and functions for the built-in numeric types.
ops	Overloadable operators.
option	Optional values.
panic	Panic support in the standard library.
pin	Types that pin data to its location in memory.
prelude	The libcore prelude
primitive	This module reexports the primitive types to allow usage that is not possibly shadowed by other declared types.
ptr	Manually manage memory through raw pointers.
result	Error handling with the `Result` type.
slice	Slice management and manipulation.

https://doc.rust-lang.org/stable/core/index.html

lang items

```
#![no_std]
#![no_main]
#![feature(start)]
#![feature(lang_items)]

#[no_mangle]
#[start]
pub fn start(_argc: isize, _argv: *const *const u8) -> isize {
    0
}

#[panic_handler]
fn panic_handler(_: &core::panic::PanicInfo) -> ! {
    loop {}
}

#[lang = "eh_personality"]
extern "C" fn eh_personality() {}
```

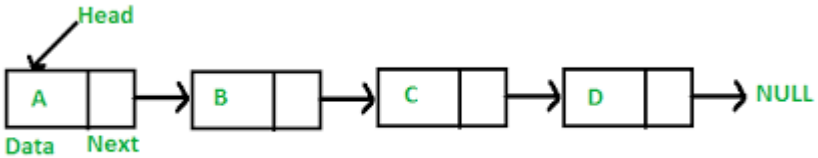
Global allocator

- 有 4M 内存可以用，不可以申请额外内存
- 单线程
- 不需要太过节省内存

Fast allocation

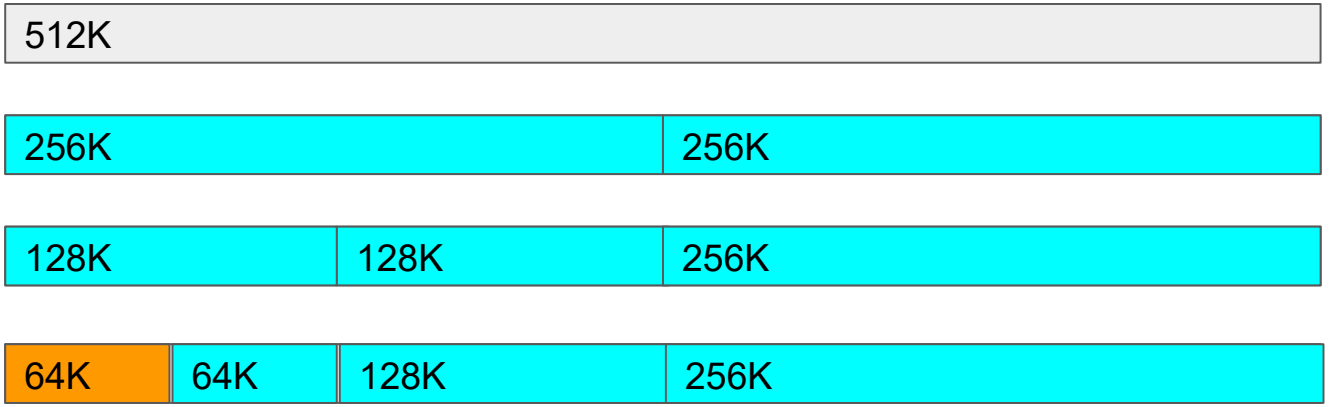
```
// Fix size 64 Bytes
pub const BLOCK_SIZE: usize = 64;

struct Node {
    next: *mut Node,
    prev: *mut Node,
}
```



- malloc -> pop
- free -> push

Buddy allocation



- 时间上稳定， malloc / free 最多迭代 log(n) 次
- 对内存利用率不高

RISC-V Syscall convention

```
.text
.globl syscall

syscall:
    ecall
    ret
```

```
#[link(name = "ckb-syscall")]
extern "C" {
    fn syscall(a0: u64, a1: u64, a2: u64, a3: u64, a4: u64, a5: u64, a6: u64, a7: u64) -> u64;
}

```

遇到的问题

Binary size

- Rust contract - 13KB
- C contract - 1.3KB

```rust
use crate::error::Error;

pub fn main() -> Result<(), Error> {
    // remove below examples and write your code here

    let script = load_script()?;
    let args: Bytes = script.args().unpack();
    debug!("script args is {:?}", args);

    // return an error if args is invalid
    if args.is_empty() {
        return Err(Error::MyError);
    }

    let tx_hash = load_tx_hash()?;
    debug!("tx hash is {:?}", tx_hash);

    let _buf: Vec<_> = vec![0u8; 32];

    Ok(())
}
```

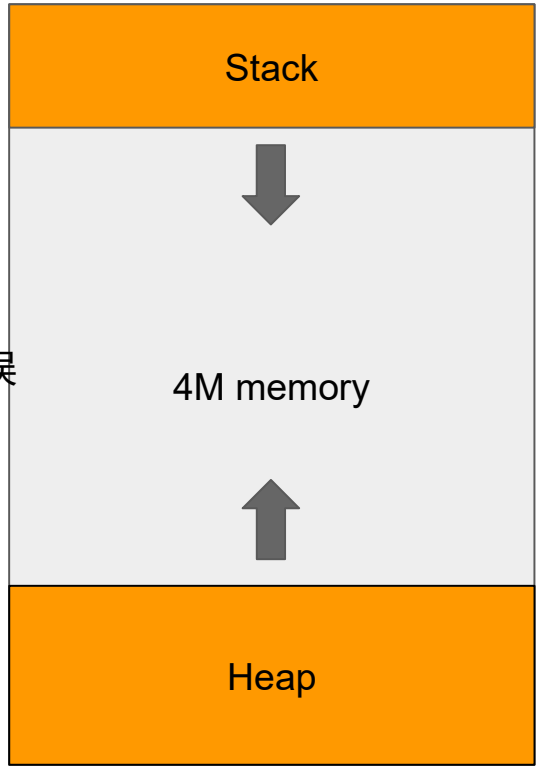
Nightly features
== Never stabled features

- 很多 feature 只能再 Nightly 使用
- Nightly version 的 rustc 有可能包含不稳定因素
- Stable 遥遥无期

```
6
7   #![no_std]
8   #![no_main]
9   #![feature(lang_items)]
10  #![feature(alloc_error_handler)]
11  #![feature(panic_info_message)]
12
```

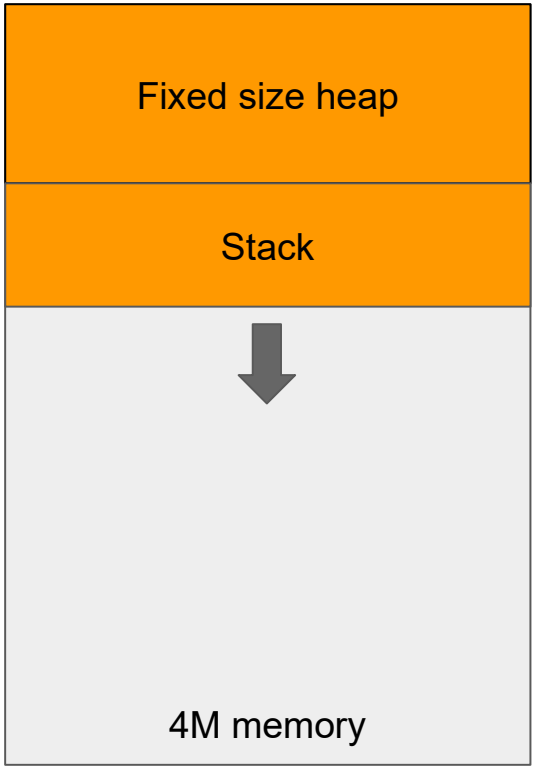
no MMU

- 如果 stack 或 heap 持续增长会造成数据错误



no MMU

- Ferrous Systems 在嵌入式环境的一个方案



THANKS

CRYPTAPE
秘猿科技

Join Us

Please send your resume to join@cryptape.com.

RUST CHINA CONF 2020

首届中国 Rust 开发者大会

2020.12.26–27 深圳

RUST CHINA CONF 2020

首届中国 Rust 开发者大会

2020.12.26–27 深圳