Assignment_8

Question 1

a. 127.0.0.1 means the IP address of this computer, 3000 means port

b. Future is a kind of zero-cost abstract and it is the value represented by a series of asynchronous computations. For example issuing an HTTP request may return a future for the HTTP response, as it probably hasn't arrived yet. With an object representing a value that will eventually be available, futures allow for powerful composition of tasks through basic combinators that can perform operations like chaining computations, changing the types of futures, or waiting for two futures to complete at the same time.

c. Parse this address

d. The body is a stream of Bytes, it is usually async, stream of Request or Response

e. The BoxFut is BoxFuture, an owned dynamically typed Future for use in cases where you can't statically type your result or need to add some indirection.

f. No, it shouldn’t

g. Yes, it should use lifetime, and it requires a ‘static’ lifetime for the boxed future.

h. The curl is an instruction of data transmission, it used to get the resource for the 3000 port. It doesn’t use Async/IO, if we want use Async/IO, we need to change Read, Write and Seek to AsyncRead, AsyncWrite, AsyncSeek and AsyncBufRead

Question 2

a. Libra written in Move language

b. Because this language is used for cryptocurrency, so the most important thing is safety. There are four features of Move, First-class resources, flexibility, safety and verifiability. Move and libra are all written in Rust, Rust could also achieve these features.

First-class resources, the program on a blockchain will interact directly with digital asset, which is a significant different from traditional program, so design the resource type, it can not be copy, cannot destroy implicitly, can transfer only.

Flexibility, Move supports transaction scripts, each transaction contains a script.

Safety, the execution format of Move is typed bytecode, the bytecode will execute bytecode verifier before execution on chain.

Verifiability, Move supports advanced under-chain static validation tools. There are some designs that make Move easier to validate statically than in a general-purpose language.

c. Lazy_static: Using this crate make it is possible to have statics that require code to be executed at runtime in order to be initialized.
Tokio: It is an event-driven, non-blocking I/O platform for writing asynchronous applications with Rust programming language. It is fast, reliable and scalable.
Failure: provides a system for a system for creating and managing errors in Rust.

Question 3

a. The nightly channel is build using the Nightly version: 1.50.0-nightly

b. Unstable features means you can use new features which is under active development in nightly version, these features maybe unstable, but if you want you can use them.

c. Because it uses the llvm_asm macro, which allow inline assembly language, and I think this can be used to develop operating system.

d. The output from this code is J.

```
    mov $$1, %rax #  move 1 to the rax register which is the return
value of the function

    mov $$1, %rdi # move 1 to the rdi register which used as the
first parameter to the function

    mov $0,%rsi # move 0 to rsi register, which used as the second
parameter to the function

    mov $1,%rdx # move 1 to rdx register, which used as the third
parameter to the function

    syscall       # call the function
```