Closures - Syntax, Semantics and Implementation

Zhou Zhihao* xzhouz00@stud.fit.vutbr.cz
Zhang Menghan* xzhang04@stud.fit.vutbr.cz

Abstract: Closure, also known as lexical closure or function closure, is a function that holds external environment variables. The external environment refers to the lexical scope where the closure is defined. External environment variables are those that are not defined within the closure [1]. Closures are widely used in various programming languages. Therefore, it is a pretty worthwhile concept to grasp.

In this presentation, we will begin with an introduction to the definition and purpose of closures. We will follow an implementation of closures with Python as an example in conjunction with closures in a programming language. Furthermore, finally, introduce some applications of closures.

Then, we will introduce compilers and interpreters with a clear illustration and discuss how they work. We will also explore the differences between compilers and interpreters.

After understanding the principles of closures and how a compiler or interpreter works, we will discuss how closures are formed during the work of the compiler or interpreter, as well as the role of each stage in the implementation of the closure.

Bibliography

[1] Han D. 2019. *The Rust Way of Programming.* Beijing Publishing House of Electronics Industry. 580 PP. ISBN 9787121354854.