

СОЗДАНИЕ КОНСОЛЬНЫХ .EXE ПРИЛОЖЕНИЙ НА PHP

Все мы знаем, что PHP предназначен для быстрого создания динамических Web страниц. Во всяком случае именно это написано на официальном сайте разработчиков PHP. Но с каждым релизом PHP становится всё быстрее, а при включении JIT (Just-In-Time) компиляции, достигает почти скорости выполнения программ, написанных на языке C. С быстрым интерпретатором PHP скорость выполнения составляет 0,046 с, а в машинных кодах 0,011 с.

JIT – компилятор, который доступен с PHP 8. Он компилирует операционный код в машинные инструкции во время выполнения. Некоторые части кода не интерпретируются в Zend VM, а выполняются непосредственно на уровне процессора. За счёт этого ускоряется работа.

Интерпретация в php происходит следующим образом:

– код PHP читается и преобразуется в набор ключевых слов, известных как токены (Tokens). Этот процесс позволяет интерпретатору понять, в какой части программы написан каждый фрагмент кода. Этот шаг называется лексирование (Lexing) или токенизация (Tokenizing);

– имея на руках токены, интерпретатор PHP проанализирует эту коллекцию токенов и старается найти в них смысл. В результате генерируется абстрактное синтаксическое дерево (Abstract Syntax Tree – AST). AST представляет собой набор узлов, указывающих, какие операции должны быть выполнены. Например, «echo 1 + 1» должно фактически означать «вывести результат 1 + 1» или, более реалистично, «вывести операцию, операция – 1 + 1»;

– преобразование этого дерева во что-то, что может быть выполнено, требует промежуточного представления, которое в PHP мы называем операционный код (Opcode). Процесс преобразования AST в операционный код называется компиляцией;

– теперь, когда у нас есть опкоды, происходит выполнение кода. PHP имеет движок под названием Zend VM, который способен получать список Opcode и выполнять их.

Для создания приложения понадобятся основной exe-файл и dll-библиотека движка. Для этого необходимо скомпилировать основной exe-файл и dll-библиотека движка, исключив все ненужные модули, оставив при этом только JIT и FFI. Для компиляции используем gcc. Полученный файл сжимаем утилитой `enigma virtual box`.