

Чудеса кросс-компиляции и свобода от ограничений в новом языке программирования.

Разработчики представили Virgil — новый системный язык программирования, нацеленный на создание компактных и производительных приложений. Для работы с ним потребуется:

Виртуальная машина Java 1.3 или более поздней версии

Компьютер под управлением Linux x86

Компьютер под управлением MacOS X x86

По словам одного из создателей языка, Бена Тицера, который также принимал участие в разработке WebAssembly, Virgil не ставит своей целью конкурировать с другим молодым языком Zig, предназначенным для подобных задач.

Virgil лишен устаревших недостатков и обладает мощными возможностями для написания надежного системного кода. Среди потенциальных областей применения — виртуальные машины, компиляторы, ядра операционных систем, сетевые стеки и другие низкоуровневые компоненты.

Сейчас, наряду с Rust — это язык программирования, разработанный компанией Mozilla. Он сочетает в себе производительность и безопасность, позволяя создавать быстрые и надёжные программы. Основные особенности Rust включают в себя строгую систему типов, которая позволяет избежать большинства ошибок времени выполнения, а также возможность эффективной работы с памятью и безопасное управление потоками данных. Rust также обладает многопоточностью, что позволяет создавать параллельные приложения. Данный язык может быть использован для различных целей, включая создание операционных систем, драйверов, веб-серверов, игр и многих других приложений. Rust постоянно развивается и улучшается благодаря активной разработке сообщества, что делает его всё более популярным среди разработчиков. Rust, появляется много новых языков, ориентированных на создание легковесных высокопроизводительных систем с кроссплатформенными компиляторами. При этом, по утверждению Тицера, Rust не может выполнять некоторые задачи, которые под силу Virgil.

Запустив Warp — терминал на базе Rust, пользователю нужно клонировать

репозиторий. Если вы давно не использовали GitHub, команду для клонирования можно найти на главной странице репозитория под вкладкой код:

Теперь мы просто клонируем репозиторий в командной оболочке:

За основу взята идея исключения устаревших элементов при проектировании высокопроизводительных систем. Вместо этого учитываются современные требования к оптимизации компиляторов, структурам кода, обработке ошибок, безопасности и другие факторы, важность которых менялась с течением времени. Например, сегодня память – довольно дешевый ресурс, но безопасность имеет первостепенное значение, в то время как 30 лет назад ситуация была обратной.

После установки Virgil можно сразу же приступать к написанию и запуску кода. Вот так будет выглядеть программа «Hello World»:

Данный код можно запустить со встроенным интерпретатором Virgil:

Однако для компиляции программы требуется указать целевую аппаратную архитектуру. К счастью, при запуске на локальной машине Virgil способен автоматически определить ее:

После компиляции в текущем каталоге появляется исполняемый файл. (Следует учесть, что показанное в `Warp` время выполнения может не отражать реальную производительность).

Запуск скомпилированного кода выводит приветствие:

Таким образом, Virgil позволяет как интерпретировать, так и компилировать написанный код, адаптируясь к аппаратной платформе.

В языке также используется концепция изменяемых и неизменяемых переменных. Изменяемые переменные вводятся ключевым словом `var`, а неизменяемые — `def`:

Функции объявляются с указанием типов параметров и возвращаемого значения,

например:

Классы в Virgil имеют схожий с другими ЯП синтаксис:

Уникальной особенностью являются алгебраические типы данных — комбинация полиморфизма, конструкторов и сопоставления с образцом:

В Virgil есть поддержка многопоточности и асинхронного программирования с использованием ключевого слова `asunc`:

Язык позволяет напрямую работать с низкоуровневой памятью через опасные необработанные указатели:

Таким образом, Virgil предоставляет возможность эффективной кросс-компиляции, генерируя машинный код или байт-код в зависимости от целевой платформы. Сам язык использует подобный кросс-компилятор, так что весь тулчейн написан на нем же.

Это лишь вводное описание, и многие возможности языка остались за кадром. Очевидно, что проект активно развивается. Программировать можно как в объектно-ориентированном стиле, так и в функциональном, что иногда называют мультипарадигмальным подходом.