Instituto Politécnico Nacional
Escuela Superior de Cómputo

Avance del proyecto 1

Compilador

Ingeniería de software
Profesora: Martha Rosa Cordero Lopez
Grupo: 3CM3
Adrian González Pardo &
Melani Betsabee Valdez Esquivel
Semestre: 20/02

Índice

1. Nombre del proyecto

Compihinador

1.1. Objetivo

El aprendizaje de lenguajes de programación actualmente tiene muchas vertientes en las cuales se puede partir desde lenguajes con una curva de aprendizaje muy sencilla y fluida, hasta lenguajes en los que su curva de aprendizaje implican el uso de algunas notaciones matemáticas o algunas notaciones que impliquen una alta abstracción del lenguaje.

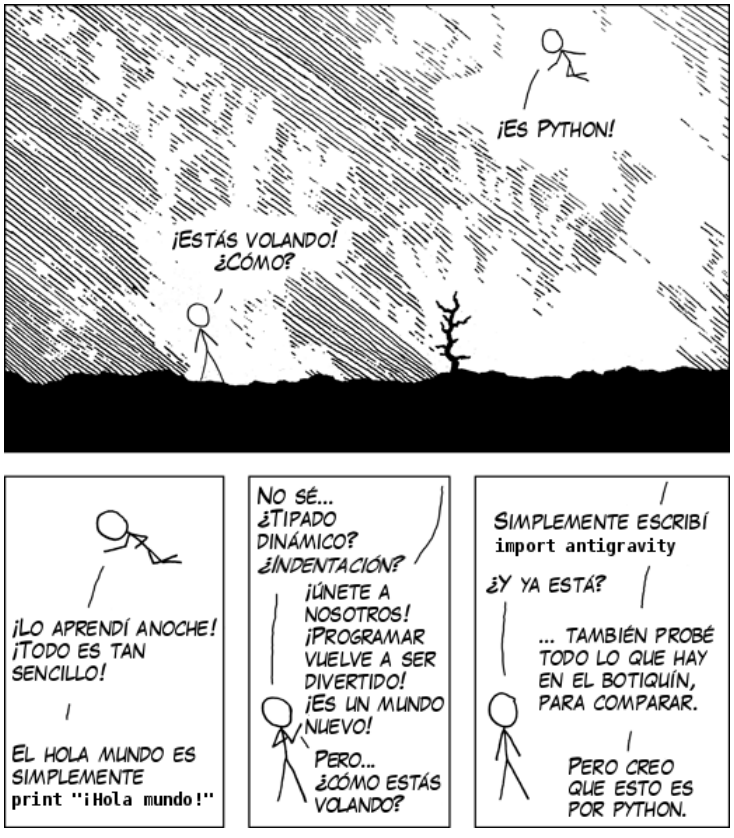

Figura 1: Python siendo mayormente usado como lenguaje de programación, siendo un lenguaje mayormente interpretado

En algunos casos el uso de lenguajes que son interpretados, generan un consumo significativo de energía y esto en volumenes enormes de información genera problemas en la industria e incluso en las pruebas de performance.

Total					
	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Figura 2: Tabla de resultados de uso de energía, tiempo y peso en disco de cada lenguaje de programación a los que se realizaron pruebas

Documento de: How Do Energy, Time, and Memory Relate?

1.2. Descripción del proyecto

El compilador realizara uso de modulos de herramientas que nos puede proporcionar el software especializado para la creación y descripción de compiladores, en los cuales se buscara realizar una aproximación de que el compilador y el uso de memoria no cree un consumo excedente de energía, memoria o tiempo al generar el código objeto.

1.3. Funciones principales

El compilador es realizado con el fin de poder tener una mejor cercanía con usuarios principiantes los cuales tiene noción de algunos lenguajes de programación como son *Python*, *Ruby*, *Perl*, *etc.*
Si bien estos lenguajes son considerados lenguajes de alto nivel, se pretende realizar un compilador que realice algunas operaciones en las que puedan tener un acercamiento a estos tipos de lenguajes y que más tarde puedan mudar de un lenguaje a otro.

1.4. Justificación

El proyecto a desarrollar contiene algunos puntos a resaltar de los cuales son Hardware y Software en el que se realizara el desarrollo del producto y de su documentación.

1.4.1. Hardware

- El hardware en el que sera desarrollado el proyecto trajara bajo una arquitectura x86_64, debido a que es el actual estandar de desarrollo.
- El equipo deseablemente contara con al menos 4GB de RAM.
- Procesador dualcore o más cores superior a 1GHz.

1.4.2. Software

- El producto pretende crear un pseudolenguaje que se compila y pretende tener una mejor cercanía con el programador principiante.
- El Sistema Operativo en el que se desarrollara y sera la plataforma en la que se ejecutara el proyecto sera en sistema Linux, debido a que las herramientas de trabajo como el analizador léxico, sintactico y semantico pueden encontrarse en paqueterias libres que proporciona Linux.
- Se pretende hacer uso de herramientas que proporciona la terminal como el uso de alias y simbolic links para que las llamadas del compilador sean más sencillas en su uso.
- Tambien se en la instalación de la paqueteria del compilador se añadiran los manuales pertinentes para que el usuario pueda trabajar correctamente con el producto.

1.5. Innovación

La innovación del producto de software es el hacer uso de estandares que nos pueden proporcionar el estandar de POSIX, y algunas otras buenas practicas de la programación de modulos de software.
Por otro lado tambien es necesario el hacer uso de algoritmos que permitan optimizar el uso de energía que genera la creación y ejecución de código en su entorno de programación.
Por lo tanto, si bien es un tiempo corto para la realización de todas las actualizaciones y desarrollo, es bueno planificar el hecho de que pueda existir una aproximación a prototipos del compilador.

1.6. Complejidad

De acuerdo a la idea de proyecto a realizar es una tarea larga ya que requiere no solo de uso de herramientas que puede ofrecer el proyecto de GNU/Linux sino que tambien implica el aprendizaje de las mismas y por otro lado tambien implical el uso de herramientas que permitan realizar pruebas de estres y testing como puede realizarlo algunas herramientas de scripting como lo son *Bash*, *sh*, *Ruby*, *Python*.

1.7. Paradigma de desarrollo

El paradigma a utilizar para el proyecto sera el uso de prototipos, así como en la escritura de código sera en el paradigma Estructurado ya que este nos permitira trabajar con las herramientas que nos ofrece el Sistema Operativo, y un poco de paradigma Orientado a Objetos para hacer uso de algunas herramientas que nos ofrece el mismo, como la modularidad y el uso de algunos empaquetamientos.

1.8. Tecnicas de desarrollo

Para la recolección de requisitos se llevara acabo una encuesta en la cual nos permitira recabar información, en la cual podremos interpretar la dirección final y los detalles a pulir del proyecto, en los cuales podremos saber si los encuestados tienen una facilidad o dificultad a la hora de trabajar con un lenguaje de programación compilado.
Los hitos planificados del proyecto es trabjarlo con base en el modelo de prototipos para despúes poder refinar los hitos consiguientes, debido a que es un modelo más sencillos para la realización del producto.
Tambien se planea el uso de paradigmas como el orientado a objetos y el estructurado para el desarrollo de los modulos/prototipos.