



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE VERACRUZ
VERACRUZ, VERACRUZ



CARRERA
INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA
LENGUAJES Y AUTOMATAS II

DOCENTE
LIC. MARTHA MARTÍNEZ MORENO

EVIDENCIAS
INDIVIDUALES

ESTUDIANTE
JOSÉ MANUEL MIRANDA VILLAGRÁN

15 de noviembre de 2018

Índice

Unidad 1 Análisis semántico	3
<i>Lenguajes de programación actuales</i>	<i>3</i>
Unidad 2 Generación de código intermedio	4
<i>Del alto nivel a ensamblador</i>	<i>4</i>
Unidad 3 Optimización	5
Bibliografía	6

Unidad 1 Análisis semántico

Lenguajes de programación actuales

La participación en el foro fue la siguiente.



EXPERIANCIA EN LOS LENGUAJES DE PROGRAMACIÓN ACTUALES
de JOSE MANUEL MIRANDA VILLAGRAN - miércoles, 29 de agosto de 2018, 01:35



¿Qué lenguajes de programación conoces?...

Los lenguajes de programación que conozco y he utilizado a lo largo de mis estudios han sido: Pascal, Java, C#, C, C++, PHP, Ensamblador de Intel.

Del lenguaje elegido para desarrollar ¿qué sabes del mismo? ¿es estructurado? ¿orientado a objetos?

El lenguaje elegido para desarrollar su compilador es Ruby, es un lenguaje de programación de scripts, moderno y orientado a objetos. Incorpora algunas de las mejores características de otros lenguajes como Smalltalk, Java y Perl, ya que su creador Yukihiro Matsumoto trabajaba como programador en lenguajes como Perl y PHP y en un principio, su intención era crear un Perl avanzado, pero en lugar de eso decidió apostar por desarrollar uno propio a 'partir de lenguajes preferidos.

Algunas de las características de Ruby son:

- Es un lenguaje multiplataforma; puede correr incluso en dispositivos móviles.
- Es un lenguaje dinámico e interpretado.
- Es altamente extensible, no solo mediante librerías escritas en Ruby, si no que podemos ampliarlo utilizando el lenguaje C.


¿Qué harías o propondrías para crear un nuevo lenguaje de programación?

Esta pregunta es ambigua ya que depende 100% del propósito para el cual se quiera diseñar. Si mi propósito es un lenguaje que sirva solo para explicar la programación en las universidades, deberá ser simple, en mi idioma y, además, la sintaxis al nivel más alto, para que los estudiantes no empleen más tiempo en entenderla que en la propia lógica de programación.

Adjunto un enlace de una documentación/tutorial: <http://rubytutorial.wikidot.com/>

Figura 1 – Participación en el foro.

Las siguientes capturas hacen referencia a replicas de dos comentarios.




RE: LENGUAJES DE PROGRAMACIÓN Y MI EXPERIENCIA
de JOSE MANUEL MIRANDA VILLAGRAN - miércoles, 29 de agosto de 2018, 02:19

Si ya has trabajado con Java, entonces hacer el compilador de C# será sencillo para ti. La sintaxis es muy similar, ambos son orientados a objetos y los tipos de datos, delimitadores, fin de línea, comentarios, etc.

Y tu propuesta para el diseño de tu lenguaje es interesante, ya que no todos tienen la ventaja de egresar de un bachillerato con carrera tecnica en programacion o a fin. Todos deberían empezar a la par.

Mostrar mensaje anterior | Responder (réplica)

Figura 2 – Respuesta 1 a participación en el foro.



RE: LENGUAJES DE PROGRAMACIÓN ACTUALES
de JOSE MANUEL MIRANDA VILLAGRAN - miércoles, 29 de agosto de 2018, 02:05

Respecto a la pregunta 3, hay bastante razón en "sencilla instalación", algunos lenguajes son complicados de instalar y ademas, muchas veces no tienen documentación oficial y no hay material en internet.

Actualmente las interfaces graficas de los editores o IDEs, si bien son agradables, consumen muchos recursos de la maquina, deberían ser un entorno básico, pero eficaz, eficiente y agradable.

Mostrar mensaje anterior | Responder (réplica)

Figura 3 – Respuesta 2 a participación en el foro.

Unidad 2 Generació de código intermedio

Del alto nivel a ensamblador

La participación en el foro fue la siguiente.

En este espacio se comenta una retroalimentación acerca de que es necesario saber antes de poder programar en ensamblador.



Ruby

José Manuel Miranda Villagrán

De alto nivel a ensamblador

Cosas que hay que saber antes de programar en ensamblador

Antes de empezar a codificar en cualquier lenguaje es necesario conocerlo, en la mayoría de los casos la estructura de cada uno no cambia, pero en bajo nivel es distinto, significa que vamos a interactuar directamente con el hardware y por ende es necesario tener conocimiento de arquitectura básica de computadoras.

Algunos temas que hay que tener en cuenta para poder entender ensamblador son:

- Manejo de memoria.
- Modos de direccionamiento.
- Conjunto de instrucciones.
- Arquitectura interna del CPU.
- Registros.
- Tipos de segmentos en la memoria.

Figura 4 – Participación en el foro.

Unidad 3 Optimización

La participación en el foro fue la siguiente.

En este espacio se comenta la experiencia obtenida como retroalimentación al ejercicio de la optimización de código y las experiencias.

Ruby

José Manuel Miranda Villagrán



Durante el ejercicio de optimización de código intermedio, se creó un algoritmo para ordenar elementos de un arreglo de n dimensiones, los valores del arreglo era generados aleatoriamente. Se diseñó de esta manera ya que al interactuar con el usuario, el tiempo de ejecución se vería afectado.

Si aplicaría un modelo de diseño del compilador el costo dependería de el objetivo, hay compiladores muy complejos y otros mas simples, unos son de código abierto y otros son de paga. En principio el compilador sería Open Source, de esta manera más personas colaborarían.

Ruby contiene una herramienta para hacer test de ejecuciones, es muy útil ya que el programador solo hace uso de ella, y no tiene que implementar un procedimiento extra. Los resultados se comprobaron con la misma implementación haciendo comparaciones de costo espacial y temporal.

Figura 5 – Participación en el foro.

Bibliografía

- Irvine, K., Romero Elizondo, A., Villar Cuesta, M., Leal Flores, A. and Morán Loza, J. (2007). *Lenguaje ensamblador para computadoras basadas en Intel®*. 5th ed. México: Pearson Educación.
- Gálvez, S. and Mora Mata, M. (2005). *Java a tope: Traductores y compiladores con LEX/YACC, JFlex/Cup y JavaCC*. 1st ed. Málaga.
- Aho, A., Ullman, J., Sethi, R. and Lam, M. (2008). *Compiladores, principios, técnicas y herramientas*. 2nd ed. México: Pearson Education.
- Ruby-lang.org. (2018). *Documentación*. [online] Available at: <https://www.ruby-lang.org/es/documentation/> [Accessed 15 Nov. 2018].