

Práctica 02

DOCENTE	CARRERA	CURSO
MSc. Vicente Enrique Machaca Arceda	Escuela Profesional de Ingeniería de Software	Fundamentos de Lenguajes de Programación

PRÁCTICA	TEMA	DURACIÓN
02	Practica 2	3 horas

1. Datos de los estudiantes

- Grupo: xxxxxx
- Integrantes:
 - Roberto Heredia Garland
 - Gabriela Pacco Huamani

2. Ejercicios

1. ¿Cuál es la diferencia entre lenguaje de maquina y lenguaje ensamblador?
El lenguaje de maquina es una secuencia de bits que controlan directamente el procesador, haciendo que este sume, compare, mueva información de un lugar a otro, y entre otros en el tiempo adecuado. Scott (2009)
Por otro lado los lenguajes Assambly fueron inventados para poder expresar operaciones con abreviaciones mnemotecnias.Scott (2009)
2. ¿En qué circunstancia un lenguaje de alto nivel es superior al lenguaje ensamblador?
Los lenguajes de alto nivel son superiores a los lenguajes ensamblador en cuanto a la reutilizacion de codigo en diferentes maquinas
3. ¿En qué circunstancia un lenguaje ensamblador es superior al lenguaje de alto nivel?
Anteriormente los compiladores no podían generar lenguaje de ensamblador mas rápidos que programar directamente en ensamblador, pero esto ya no es asi.
4. ¿Por qué hay tantos lenguajes de programación?
Debido a la evolución de Ciencias de la computación, propósitos específicos de los lenguajes, preferencia personal, facilidad de uso y entre otros.

5. Nombre 3 lenguajes en las categorías de von Neumann, funcional y orientado a objetos. Dos lenguajes lógicos y 2 concurrentes.

a) von Neumann

- C
- Ada
- Fortran

b) Funcional

- Lisp
- ML
- Haskell

c) Orientado a Objetos

- SmallTalk
- Eiffel
- Java

d) Logicos

- Prolog
- Spreadsheets

e) Logicos

- Id
- Val

6. ¿Qué distingue a los lenguajes declarativos e imperativos?

Los lenguajes declarativos se enfocan en que tiene que hacer la computadora y los lenguajes imperativos se enfocan en como la computadora debe hacerlo.

7. ¿Cuál es considerado el primer lenguaje de alto nivel?

Fortan

8. ¿Cuál es considerado el primer lenguaje funcional?

Lambda calculus

9. ¿Por qué los lenguajes concurrentes no están considerados en la clasificación de Scott (2000)(Figura 1.1).?

Debido a que en realidad las categorías están abiertas a debate y los autores no se ponen de acuerdo.

10. Lista las principales fases de un compilador y describe la función de cada fase.

a) Scanner (Análisis Léxico)

Genera un flujo de Tokens. Se simplifica el trabajo del Parser reduciendo el tamaño del flujo.

b) Parser (Análisis sintáctico)

Genera un árbol Parse, basado en gramaticas libres de contexto.

c) Semantic analysis

En el analisis semantico se genera un árbol de sintaxis abstracta. En esta fase se reconoce las cuando los identificadores apuntan a una misma entidad de programa. Se genera una tabla de símbolos.

- d) Machine-independent code improvement
Se mejora el árbol generado en el análisis semántico
 - e) Target code generation Se genera el código en el lenguaje objetivo (assembler) recorriendo el árbol de sintaxis.
 - f) Machine-specific code improvement Se mejora el código generado en el paso anterior.
11. ¿En qué circunstancias tiene sentido que un compilador pase o revise el código varias veces?
Cuando se quiere optimizar el código.
12. ¿Cuál es el propósito de la tabla de símbolos en un compilador?
Se utiliza para determinar reglas que no estén en la estructura de la gramática de libre contexto.
13. ¿En la actualidad, que programa es más eficiente, uno desarrollado desde cero en ensamblador o uno generado por un compilador?
Uno generado por un compilador.

3. Investigación

FORTRAN Fortran es un lenguaje de programación que está orientado y adaptado para aplicaciones numéricas y computación científica. Con Fortran nació la programación moderna. A través de él se han puesto en práctica conceptos como la computación científica, o la complicación de código, entre otros. El origen de este lenguaje de programación se remonta al año 1954, y se le atribuye a John Backus, un experimentado científico de computadores estadounidense que pertenecía a la empresa IBM. Su propuesta se centraba en poner en marcha un lenguaje de programación cuyo objetivo era traducir de una manera sencilla, y accesible, diferentes fórmulas matemáticas en código que pudiese entender un ordenador. Como curiosidad, este especialista en computación, estuvo trabajando en un proyecto previo denominado SSEC (Selective Sequence Electronic Calculator) para conseguir que este programa pudiese calcular las posiciones de la luna. Actualmente el lenguaje FORTRAN es utilizado, por una parte, debido a la existencia de numerosas bibliotecas de funciones utilizables en FORTRAN, por otra parte, porque existe compiladores FORTRAN potentes que producen ejecutables muy rápidos. No obstante, se reemplaza cada vez más, incluso para aplicaciones científicas, por los lenguajes C y C ++.

Dado que el FORTRAN se creó en la época de las tarjetas perforadas (en particular con el sistema FMS), mantiene una determinada rigidez en la compaginación del programa fuente, hasta el FORTRAN 90. El código por mucho tiempo debió comenzar a partir de la 7ª columna y no sobrepasar de la 72 (las columnas 73 a 80 se reservan para la numeración de las tarjetas perforadas).

Además, se escribieron desde hace tiempo numerosos códigos industriales en FORTRAN, y la compatibilidad de las nuevas versiones con las anteriores es indispensable, al precio de la conservación de conceptos anticuados.

EL lenguaje BASIC, en su versión original (1964) se concibió como un pequeño lenguaje de carácter pedagógico que permitía iniciar a los estudiantes en la programación, antes de pasar a los lenguajes "serios" de aquella época: FORTRAN y ALGOL. Se encuentran en él algunas características del lenguaje FORTRAN. Evolución de Fortran A la hora de presentarse, hubo algunas reticencias, ya que todos estaban acostumbrados a su antecesor, el lenguaje ensamblador que surgió en el año 1949. Pero pronto cambió la percepción general ya que eran muchas las ventajas que conllevaba utilizar Fortran. Fue considerado como un lenguaje de programación de alto nivel, que conseguía traducir programas enteros sin necesidad de hacerlo de forma manual como con sus predecesores. Además, su uso era más sencillo, no tan restrictivo como lo fueron los anteriores lenguajes de programación existentes. Una de las cosas que consiguió revolucionar el mundo de la programación fue el hecho de poder permitir que el código se escribiese de manera más rápida, y además no requería de profesionales tan especializados, lo que lo hacía más accesible a cualquiera. Es un lenguaje que nunca ha dejado de evolucionar. Ha ido

variando a lo largo de los años hasta llegar a Fortran 2018 al que se le han incluido nuevas funcionalidades y mejoras desde su origen. Fortran ha servido de inspiración y base para la creación de otro tipo de lenguajes de programación como: Lisp (1958), COBOL (1959) o ALGOL (1958). Sin duda, es uno de los lenguajes que todavía se tienen en cuenta a la hora de trabajar con ellos, y que ha servido de información para crear otros aspectos de programación derivados en base a él. Ventajas y desventajas de Fortran

1. Entre sus ventajas destacan las siguientes:

- Más sencillo de aprender que sus antecesores.
- Todavía se utiliza como uno de los lenguajes más destacados a la hora de realizar cálculo numérico.
- Se considera una revolución y el principio de la programación moderna.
- Su puesta en práctica, y los años de uso han dado lugar a librerías probadas y eficientes que constatan su eficacia como lenguaje de programación.

2. Sus desventajas también han de tenerse en cuenta a la hora de ser utilizado:

- Es un lenguaje de programación en el que no existen clases, o estructuras.
- Imposibilita el hecho de hacer una reserva de memoria dinámica.
- Para el proceso de textos, listas y estructuras de datos de alto grado de complejidad es un lenguaje algo primitivo.

Elementos básicos de un programa en FORTRAN Un programa en FORTRAN tiene los siguientes elementos básicos:

- Nombre del programa. El nombre del programa es en realidad opcional, pero es muy buena idea tenerlo.
- Declaraciones de variables utilizadas en el programa.
- Cuerpo del programa.
- Comandos a ejecutar en el código. Los comandos se ejecutan en orden de aparición. El programa siempre debe terminar con el comando END.

Listing 1: Fortran

```
1
2 REAL T, SUM
3   INTEGER M
4   T = 6.
5   SUM = 0.0 + T
6   WRITE (*,*) 'SUM IS ', SUM
7   STOP
8   END
```

Referencias

M. L. Scott, *Programming Language Pragmatics, Third Edition*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009.