UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Asistente de Programción en lenguaje C

Presentado por Rubén Marcos González
en Universidad de Burgos — 13 de mayo de 2020
Tutor: Carlos Pardo Aguilar

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Carlos Pardo Aguilar, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Rubén Marcos González, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de mayo de 2020

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general III

Índice de figuras IV

Índice de tablas V

1 Introducción 1

2 Objetivos del proyecto 3

Conceptos teóricos 5

Técnicas y herramientas 7

Aspectos relevantes del desarrollo del proyecto 9

Trabajos relacionados 11

Conclusiones y Líneas de trabajo futuras 13

Índice de figuras

Índice de tablas

2.1. Ejemplo de visualización de la tabla de variables 3
4.2. Herramientas y tecnologías utilizadas en cada parte del proyecto 8

Capítulo 1

Introducción

La intención de este proyecto es crear un asistente de programación para el lenguaje C que permita ver a los nuevos alumnos como se van modificando las variables linea a linea.

Para ello se utilizar a un interprete de C que se modificará convenientemente para que vaya mostrando de forma gráfica como las variables se van modificando.

Capítulo 2

Objetivos del proyecto

- El principal objetivo del proyecto es la creación de una interfaz limpia y clara que permita monitorizar las variables para nuevos alumnos que nunca hayan estudiado programación. Para ello nos basaremos en interfaces de desarrollo vistos en distintas clases prácticas: MatLab, Spyder y Eclipse.

- Uno de los objetivos es que el interprete de C muestre por pantalla el estado de todas la variables utilizadas en la ejecución, mostrando su valor, su espacio en memoria y su tipo ver 2.1.

Nombre	**Tipo**	**Hex**	**valor**
Num	int	0x5F8	1528
Letra	char	0x3F	?
Vector	int[3]	-	-
Vector[0]	int	0x17	23
Vector[1]	int	0x39	57
Vector[2]	int	0x8D	141
Estructura	Struct[3]	-	-
Estructura[0]	int	0x5D	93
Estructura[1]	char	0x40	@
Estructura[2]	char	0x20	

Tabla 2.1: Ejemplo de visualización de la tabla de variables

- Otro de los objetivos es que el interprete nos permita ejecutar las lineas de código una a una
- Siguiendo con el objetivo anterior permitir la inclusión de breakpoints para facilitar la depuración de código

Conceptos teóricos

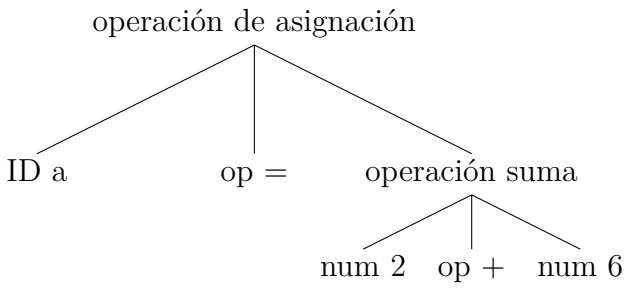
Compilador : un compilador es un traductor que se encarga de convertir el código fuente a código máquina para poder ser entendido por una maquina

Interprete : a diferencia de los compiladores que traducen todo el código de un vez, los interpretes, van convirtiendo el código poco a poco, linea a linea, instrucción a instrucción, etc.

Lexer o analizador léxico : el lexer analiza una secuencia de caracteres y la convierte en una secuencia de tokens, por ejemplo: a = 2 + 6. El lexer lo traduciría como:

1. identificador [a]
2. operador de asignación [=]
3. constante numérica [2]
4. operador de suma [+]
5. constante numérica [6]

Parser o analizador sintáctico : agrupa los token resultados del lexer formando frases gramaticales que posteriormente serán analizadas por el compilador. Utilizando el ejemplo visto con el lexer (a = 2 + 6), así quedaría con el parser:



[1]

Algunos conceptos teóricos de LaTeX[2].

[1]Definiciones extraídas del libro: Jiménez Millán, José Antonio. (2014). Compiladores y procesadores del lenguaje. Cádiz, España: Servicio de Publicaciones de la Universidad de Cádiz.

[2]Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

Técnicas y herramientas

Para la realización de este proyecto se usará un debbuger de C y se adaptará a una nueva interfaz mas clara y limpia para los alumnos nuevos que den sus primeros pasos en programación.

Para la realización del proyecto me he ayudado de un parser de C para la librería PLY de Python, pycparser [3]

Para programar se ha utilizado Spyder del paquete de Anaconda 3 [4]

Para la documentación en LaTeXse ha utilizado Overleaf[5], he utilizado Overleaf porque al tener el proyecto en nube me es mas cómodo y fácil acceder a él desde cualquier dispositivo además de incluir un previsualizador online del pdf resultante del documento de LaTeX

Utilizo el compilador de gcc, que al no venir de base en Windows se ha utilizado el paquete de herramientas de MinGW [6]

[3]Repositorio de GitHub: https://github.com/eliben/pycparser
[4]Sitio web de anaconda para su descarga: https://www.anaconda.com
[5]Sitio web de Overleaf: https://www.overleaf.com
[6]Sitio web de MinGW: http://mingw.org/

Herramientas	Aplicación	Memoria
Spyder	X	
Pycparser	X	
GitHub Desktop	X	X
Overleaf		X

Tabla 4.2: Herramientas y tecnologías utilizadas en cada parte del proyecto

Aspectos relevantes del desarrollo del proyecto

En un principio se contemplo la posibilidad de realizar el trabajo en Google Colaboratory[7] pero por problemas de comunicación entre el notebook de Google Colab y la maquina local se terminó descartando la idea.

Posteriormente se intento realizar el trabajo en un notebook local utilizando Jupyter, incluido en el paquete de Anaconda 3, pero debido a la restricción de permisos que tienen los exploradores de internet se terminó descartando también.

Para el interprete de C se pensó aprovechar la herramienta gdb incluida en Linux de base y en Windows mediante MinGW pero ante la imposibilidad de crear una comunicación continua entre la consola de la maquina y la aplicación se terminó por descartar esta opción.

Se tenía la intención de diseñar el parser y lexer enteros utilizando esta gramática: enlace a gramática y este léxico: enlace a léxico

[7]Sitio web de Googe Colab: https://colab.research.google.com/notebooks/intro.ipynb

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.