**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**



Trabajo monográfico para la optar al

Título en Ingeniería en Computación

**Herramienta de generación de código para definición y gestión de lenguaje ensamblador como recurso didáctico para la asignatura de Máquinas Computadoras II de la Universidad Nacional de Ingeniería (UNI)**

**AUTOR:**

**Br. Harvin Manuel Toledo Polanco**

**Br. Jorge Manuel Potosme Álvarez**

**TUTOR:**

**Ing. José Díaz Chow**

**Managua, Nicaragua**

**2016**

Contenido

[Dedicatoria 3](#_Toc444347578)

[Agradecimientos 4](#_Toc444347579)

[Resumen 5](#_Toc444347580)

[Introducción 6](#_Toc444347581)

[Antecedentes 7](#_Toc444347582)

[Metodología de desarrollo 7](#_Toc444347583)

[Objetivos 8](#_Toc444347584)

[General 8](#_Toc444347585)

[Especifico 8](#_Toc444347586)

[Justificación 9](#_Toc444347587)

[Marco Teórico 10](#_Toc444347588)

[Análisis y Presentación de Resultados 11](#_Toc444347589)

[Bibliografía 12](#_Toc444347590)

[Anexos 12](#_Toc444347591)

[Introducción al lenguaje ensamblador 12](#_Toc444347592)

[Lenguaje de Bajo Nivel 12](#_Toc444347593)

[Ventajas y desventajas de los lenguajes ensambladores 12](#_Toc444347594)

[El procesador 14](#_Toc444347595)

[Unidades Funcionales Básicas 14](#_Toc444347596)

[Unidad Aritmética Lógica (ALU) 14](#_Toc444347597)

[Unidad de Control (CU) 14](#_Toc444347598)

[Plataforma de desarrollo 18](#_Toc444347599)

# Dedicatoria

# Agradecimientos

# Resumen

# Introducción

## Antecedentes

## Metodología de desarrollo

# Objetivos

## General

Generar una interfaz de programación de aplicaciones (API ) para definición y gestión (de código fuente) de lenguaje ensamblador con el fin de proporcionar a los estudiantes de la asignatura de Arquitectura de Máquinas II, de la Universidad Nacional de Ingeniería, una herramienta que les facilite realizar simulaciones de arquitectura de máquinas para distintas gammas de procesadores.

## Especifico

* Determinar los requerimientos para la herramienta de generación de código a desarrollar a partir de las características requeridas por la asignatura y las necesidades de los estudiantes para el desarrollo del gestor de código fuente un simulador de arquitectura.
* Establecer los parámetros básicos con los cuales se pueda definir la arquitectura objetivo de la herramienta de simulación.
* Identificar las plataformas de ejecución del programa simulador de microprocesador para la herramienta de simulación.

# Justificación

# Marco Teórico

## Herramientas de desarrollo

### QT Creator

Es un entorno de desarrollo integrado IDE, creado y mantenido por la empresa Trolltech; es ampliamente utilizado para desarrollar aplicaciones basadas en librerías QT, es multiplataforma y puede correr en sistemas Windows, Linux y Mac.

Incluye un manejador de proyectos que puede usarse en una variedad de formatos de proyecto tales como .pro, CMake, Autotools entre otros.

Integra un editor de código y un interfaz de diseño (Qt Designer) para construir interfaces graficas de usuarios (GUI) utilizando los Qt widgets. El editor de código suporta resaltado para varios lenguajes además puede analizar código en c++ y en QML, dando como resultado un editor de código con completación de código, también provee navegación semántica.

La herramienta de diseño Qt Designer es una herramienta que permite crear interfaces graficas utilizando los controles de Qt (Qt Widgets), es posible crear controles y diálogos personalizados utilizando diferentes estilos y resoluciones directamente en el editor.

Los Widgets o controles creados con Qt Designer están integrados con código programado usando el mecanismo de señales(signals) y slots de Qt.

Qt Quick Designer es una herramienta para desarrollar animaciones usando el lenguaje declarativo QML.

Provee soporte para compilar aplicaciones para entornos de escritorio windows, Linux, freebsd y mac os, dispositivos móviles (Android, BlackBerry, Maemo y MeeGo) y dispositivos con sistemas Linux embebidos.

Es fácilmente integrable a sistemas de control de versiones entre los cuales están: GIT, Subversion, Preforce, Bazaar, CVS y Mercurial.

Posee un simulador (Qt Simulator) para probar aplicaciones destinadas a dispositivos móviles, simulando un entorno similar al dispositivo de destino.

Qt creator no incluye un debugger de manera nativa, pero posee un plugin que actua de interface entre el Core de Qt Creator y un debugger nativo de c++, los debugger soportados son los siguientes: Gnu Symbolic Debugger (GDB), Microsoft Console Debugger(CDB), LLVM debugger(LLDB)

### C++

Es un lenguaje de programación utilizado ampliamente en su paradigma orientado a objetos fue diseñado a mediados de los años 80 por Bjarne Stroustrup con la intención de extender el lenguaje de programación C.

El nombre C++ fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera del laboratorio científico. Antes se había usado el nombre “C con clases”. La expresión “C++” significa incremento de C.

### QT

Es una biblioteca multiplataforma ampliamente usada para desarrollar aplicaciones con interfaz gráfica de usuario, así como también para el desarrollo de programas sin interfaz gráfica, como herramientas para la línea de comandos y consolas para servidores.

# Análisis y Presentación de Resultados

# Bibliografía

# Anexos

## Introducción al lenguaje ensamblador

### Lenguaje de Bajo Nivel

Los lenguajes de bajo nivel, también llamados lenguajes de ensamblador, permiten al programador escribir los programas usando palabras nemotécnicas que ejercen control completo sobre el hardware, razón por la cual son muy dependientes del hardware donde se trabaje, no es lo mismo el lenguaje ensamblador para procesadores Intel que el lenguaje ensamblador para procesadores PowerPC(IBM) o procesadores ARM etc,

El uso de la palabra bajo en su denominación no implica que el lenguaje sea menos potente que un lenguaje de alto nivel, si no que se refiere a la reducida abstracción entre el lenguaje y el hardware. Por ejemplo, se utiliza este tipo de lenguajes para programar tareas críticas de los sistemas operativos, de aplicaciones en tiempo real o controladores de dispositivos.

### Ventajas y desventajas de los lenguajes ensambladores

Los lenguajes ensambladores tienen algunas ventajas sobre los lenguajes de alto nivel, por ejemplo:

Velocidad: debido a que el lenguaje ensamblador trabaja directamente con el hardware permite más rápido acceso a los recursos de la computadora por esta razón la velocidad, no obstante esto también puede depender mucho de la calidad del código fuente.

Flexibilidad: el lenguaje ensamblador es flexible ya que puede utilizar el hardware de en su totalidad, sin restricciones de ningún tipo, también puede trabajar independiente del sistema operativo por ejemplo un programa en ensamblador puede correr tanto en sistemas Unix como en sistemas Windows,

Desventajas

Portabilidad mínima: los programas creados en ensamblador al depender tanto de la arquitectura del procesador, no pueden ejecutarse en computadoras que posean una arquitectura diferente de donde fueron creados.

Mantenimiento de código es muy complicado: modificar, agregar o eliminar alguna instrucción y/o función resulta mucho más complicado debido a que el lenguaje ensamblador no permite mucha abstracción por esta razón el código fuente es más grande.

## El procesador

## Unidades Funcionales Básicas

### Unidad Aritmética Lógica (ALU)

La ALU es la encargada de realizar operaciones aritméticas sobre los datos de la memoria tales como suma, resta, multiplicación y división, también se encarga de realizar las operaciones lógicas, tales como AND,OR, NOT etc.

Unidad Aritmética: todas las operaciones pueden realizarse mediante tres sistema:

* Sistema serie: se procesa bit a bit en forma secuencial
* Sistema paralelo: se procesan simultáneamente todos los bits.
* Sistema paralelo-serie: se procesa por grupos. Los grupos se procesan en serie y los bits que componen los grupos en paralelo.

Unidad Lógica: establece comparaciones para facilitar la toma de decisiones. Estas comparaciones son mucho más sencillas que las operaciones aritméticas, puesto que no necesitan considerar el resultado de la operación realizada con los bits anteriores, pudiéndose efectuar en paralelo, El resultado de las operaciones puede ser vedadero o falso, que en representación binaria es 1 o 0.

### Unidad de Control (CU)

La unidad de control se encarga de buscar las instrucciones en la memoria principal, interpretarlas y ejecutarlas, empleando para ello la unidad empleando para ello la unidad de procesos,

Coordina todos los componentes del computador de modo que los eventos tomen lugar en la secuencia apropiada en el momento correcto. Además de realizar esta labor de sincronización, la unidad de control, es decir entiende las instrucciones del programa que obtiene de la memoria y dirige la acción para realizarlas.

La secuencia lógica que la unidad de control debe realizar para ejecutar una instrucción es la siguiente:

1. Localizar y extraer de la memoria principal la instrucción correspondiente.
2. Transferir la instrucción de la memoria a la Unidad de Control.
3. Determinar qué tipo de operación se debe ejecutar.
4. Ejecutar la instrucción, enviando las señales de control u órdenes a los elementos pertinentes.
5. Supervisar la operación anterior para determinar si ha finalizado correctamente.
6. Localizar la siguiente instrucción a ejecutar.

Elementos de la Unidad de Control

El reloj: consiste en un circuito eléctrico capaz de generar una sucesión de pulsos a intervalos de tiempo constantes. El intervalo entre dos puntos de reloj se denomina *ciclo*.

Contador de programa (PC): también denominado registro contador de instrucción, (RCI). Su misión es controlar el orden de ejecución de las instrucciones del programa, de acuerdo con su contenido. Un programa no siempre ejecuta las instrucciones secuencialmente. Puede haber instrucciones de salto o bifurcación.

Registro de Instrucción: es una unidad de almacenamiento temporal, este registro guarda la instrucción cuando se extrae de la memoria principal y se mantiene mientras se realiza la decodificación o interpretación.

Decodificador: habitualmente, toda instrucción contiene un campo conocido como código de operación, que indica el tipo de operación que hay que realizar; el decodificador es el elemento encargado de realizar el análisis del código de operación.

Secuenciador: Es un generador de ordenes simples, denominadas microordenes que sincronizadas con el reloj y distribuidas a los elementos necesarios permiten la ejecución de la instrucción.

1. Arquitectura de registros dos direcciones

Esta arquitectura se caracteriza por solo definir 2 operandos, uno corresponde a fuente y destino a la vez por tanto se da una lectura destructiva.

1. Código de instrucciones
   1. Instrucciones de transferencia de datos

MOV

Copia el operando 2 al operando 1

Operandos posibles:

Registro, Memoria

Memoria, Registro

Registro, Registro

Memoria, Inmediato

PUSH

Almacena un valor de 16 bits en la pila

Operandos posibles

Registro

-sreg

Memoria

Inmediato

POP

Obtiene un valor de 16 bits de la pila

Operandos posibles:

Registro

Memoria

-sreg

* 1. Instrucciones Lógicas

AND

Instrucción lógica AND entre todos los bits de 2 operandos, el resultado se almacena en el operando 1.

Operandos posibles

Registro, Memoria,

Memoria, Registro

Registro, Registro

Memoria, Inmediato

Registro, Inmediato

OR

Operación lógica OR entre todos los bits de 2 operandos, el resultado es almacenado en el primer operando.

Operandos posibles:

Registro, Memoria

Memoria, Registro

Registro, Registro

Memoria, Inmediato

Registro, Inmediato

XOR

Operación lógica XOR (OR exclusivo) entre todos los bits de un operando, el resultado es almacenado en el primer operando.

Operandos posibles:

Registro, Memoria

Memoria, Registro

Registro, Registro

Memoria, Inmediato

Registro, Inmediato

NOT

Invierte cada bit del operando.

Operandos posibles:

Registro

Memoria

* 1. Instrucciones Aritméticas

ADD

Suma los operandos

Operandos posibles:

Registro, Memoria,

Memoria, Registro

Registro, Registro

Memoria, Inmediato

Registro, Inmediato

## Plataforma de desarrollo

Java

Java es el nombre de un entorno o plataforma de computación originaria de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación Java u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo. En este caso, la plataforma no es un hardware específico o un sistema operativo, sino más bien una máquina virtual encargada de la ejecución de las aplicaciones, y un conjunto de bibliotecas estándar que ofrecen una funcionalidad común.

Máquina Virtual de Java

El núcleo de la plataforma Java es el concepto común de un procesador virtual que ejecuta programas escritos en el lenguaje de programación Java. En concreto ejecuta el código resultante de la compilación del código fuente, conocido como bytecode.

Bytecode

Es el resultado de compilar un código fuente de java, el código binario de Java no es un lenguaje de alto nivel, sino un verdadero código de máquina de bajo nivel, viable incluso como lenguaje de entrada para microprocesador físico, pero los productos de derivados de construir microprocesadores que aceptaran el Java bytecode como lenguaje de maquina fueron infructuosos,

Fuente Java

(.java)

Bytecode

(.class)

Máquina Virtual de Java

Sistema Operativo

C#

Código Fuente

Recursos

Referencias

Compilador

Ensamblado (.exe o .dll)

Metadatos MSIL

.NET Framework

Common Language Runtime

Garbage Collection

JIT Compiler

Clases y librerías de .NET Framework

Sistema Operativo

Convierte a código de maquina nativo

Crea

Metadatos IL y referencias cargadas por CLR