**LABORATORIO 2**

**ORGANIZACIÓN DE COMPUTADORES**

|  |  |
| --- | --- |
| Nombre: | José Maureira |
| Profesor: | Leonel Medina |
| Ayudante: | Ricardo Álvarez |
| Fecha de Entrega: | 23-nov.-18 |

Santiago de Chile

2 - 2018

Tabla de contenido

[Capítulo 1. Introducción 1](#_Toc530782577)

[1.1 Enunciado del problema 1](#_Toc530782578)

[1.2 Motivación 4](#_Toc530782579)

[1.3 Objetivos 4](#_Toc530782580)

[1.3.1 Objetivo general 4](#_Toc530782581)

[1.3.2 Objetivos específicos 4](#_Toc530782582)

[1.4 Herramientas 4](#_Toc530782583)

[1.5 Estructura del informe 5](#_Toc530782584)

[Capítulo 2. Marco teórico 7](#_Toc530782585)

[Capítulo 3. Desarrollo 9](#_Toc530782586)

[Capítulo 4. Experimentos a realizar 11](#_Toc530782587)

[4.1 Resultados 13](#_Toc530782588)

[4.2 Análisis de resultados 14](#_Toc530782589)

[Capítulo 5. Conclusión 15](#_Toc530782590)

[Capítulo 6. Referencias 17](#_Toc530782591)

Índice de Figuras

Ilustración 1: Ejemplo archivo de entrada 2

Ilustración 2: Archivo resultado del juego 3

Ilustración 3: Archivos resultado de etapas 3

Ilustración 4: Camino de datos 8

Ilustración 5: Experimento.1 11

Ilustración 6: Experimento.2 12

Ilustración 7: Experimento.3 12

Ilustración 8: Experimento.4 13

Ilustración 9: Resultados experimento 13

# Introducción

Este informe se desarrolla bajo el marco del laboratorio de la asignatura de Organización de Computadores, en el cual se analiza la estructura y el funcionamiento de los computadores, en particular aquí se hará incapie en lo que es el camino de datos de uno de ellos, es decir, el camino que toman las diferentes señales eléctricas a través de las diferentes etapas de un circuito electrónico cuando ejecutan una instrucción.

Para el estudio y análisis llevado a cabo presentado en este informe se ha trabajado con el lenguaje de programación MIPS y en especifico se analizará el camino de datos de ciertas instrucciones MIPS en un procesador monociclo.

## Enunciado del problema

Para este proyecto de laboratorio en especifico se pide un programa que sea capaz de simular el famoso juego conocido por “Tic Tac Toe” o “Gato”, esto en el lenguaje de programación C, para esto las jugadas del juego serán ingresadas mediante un archivo de texto que contiene instrucción del lenguaje de programación MIPS. El programa desarrollado debe ser capaz de leer estas instrucciones y con ellas simular una partida del juego mencionado y los resultados del mismo.

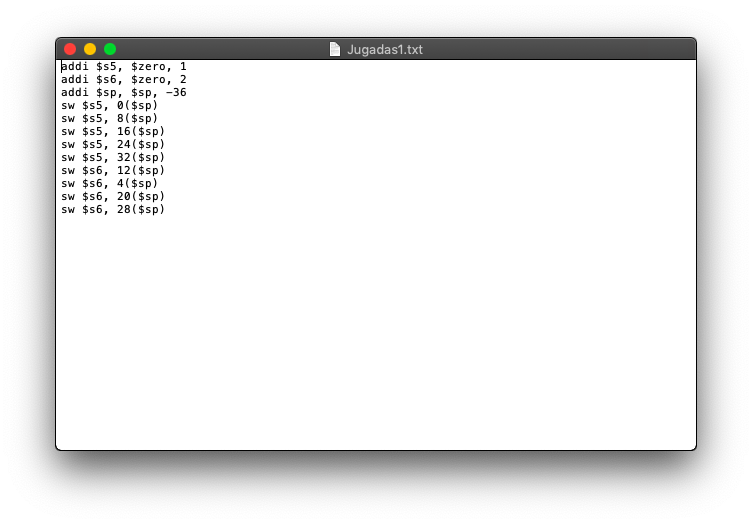
El archivo de texto contiene siempre al inicio del mismo la definición de jugadores mediante la instrucción add, luego puede contener o no las instrucciónes para la creación del tablero mediante la instrucción add y a cada instrucción siguiente hace referencias a las jugadas de cada uno de los jugadores mediante add, sw y subi (Ilustración 1). 

Ilustración : Ejemplo archivo de entrada

El programa debe generar uno o más archivos donde se indique el estado final del juego, el resultado del mismo y cuantas veces se pasó por cada etapa del camino de datos de un procesador monociclo al ejecutar las instrucciones MIPS. (Ilustración 2 e Ilustración 3).

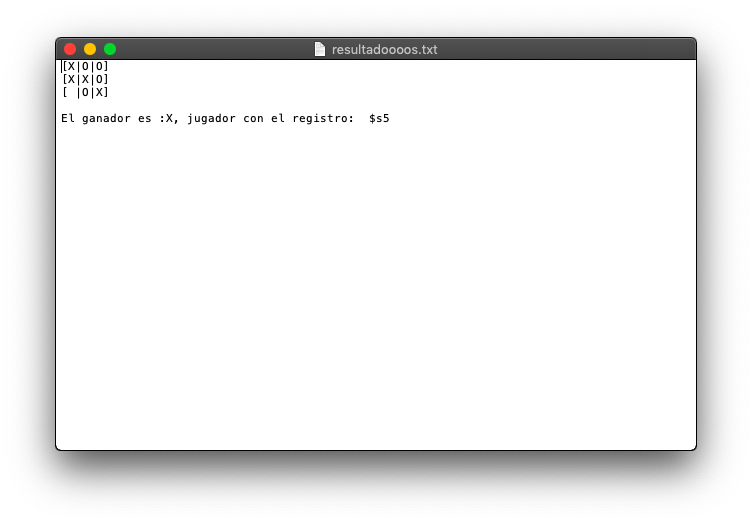


Ilustración : Archivo resultado del juego

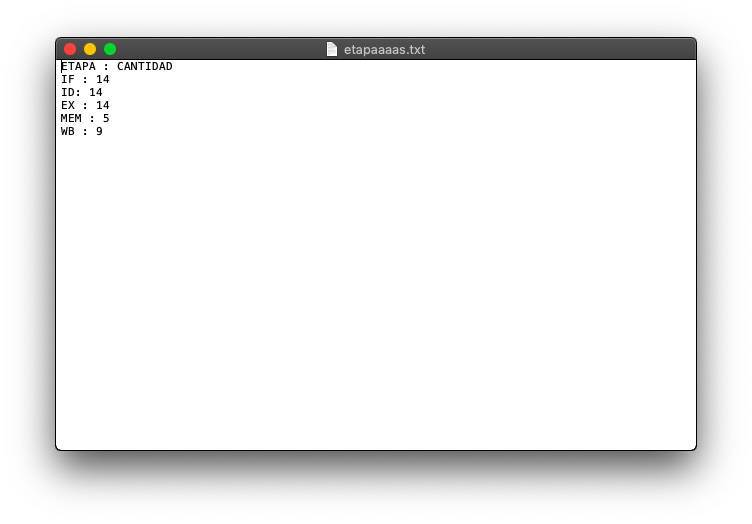


Ilustración : Archivos resultado de etapas

## Motivación

La motivación para este proyecto ha sido llevar a algo tan cercano como lo es el juego del “TicTacToe” la teoría y lo practico que se ha podido aprender hasta el momento en la asignatura de Organización de Computadores, más en especifico el como funciona el camino de datos y sus etapas en un procesador monociclo.

## Objetivos

### Objetivo general

En grandes rasgos el objetivo perseguido en este proyecto es desarrollar un programa que simule el juego “TicTacToe” mediante instrucciones MIPS en un archivo de texto que será recibido como entrada.

Generar archivos de texto capaces de representar el resultado del juego, el ganador, el registro del ganador y un conteo de las etapas del camino de dato usadas.

### Objetivos específicos

* Traducir instrucciones de un lenguaje de bajo nivel como MIPS a uno de más alto nivel como C (estándar ANSI C).
* Poder tener un registro de cuantas veces las instrucciones pasaron por las etapas del camino de datos de un procesador monociclo.
* Usar estructura de datos para almacenar registros.
* Desarrollar el programa con usabilidad.
* Desarrollar el programa cumpliendo un mínimo de calidad de software.

## Herramientas

Las herramientas utilizas para el desarrollo de este programas fueron las siguientes:

* Editor de texto y código *Sumblime Text* Versión 3.1.1
* Sistema operativo *macOS Mojave*
* Compilador de C nativo de la consola de *macOS Mojave*
* Librerías math.h, stdlib.h, stdio.h, string.h.

## Estructura del informe

La estructura de este informe se compone de manera tal que, primero se da cuenta del contexto en el cual está situado este proyecto de laboratorio y una breve introducción de lo que se espera más adelante haciendo incapié en los conceptos a tratar.

Se da a conocer el enunciado de lo que se debe llevar acabo en este proyecto, seguido de la motivación, los objetivos generales, especificos y las herramientas a utilizar para llevar a cabo todo lo planteado.

Posteriormente se detalla una estructura de informe para guiar al lector en este informe y darle paso a un marco teórico el cual entregará la información necesaria sobre MIPS, camino de datos y el procesador monociclo.

Luego se desarrolla el proceso de la solución a lo solicitado en el enunciado, para más tarde mostrar experimentaciones éxitosas del programa.

Para finalizar se procede a hacer un análisis de resultados y las conclusiones del proyecto.

# Marco teórico

* **MIPS (***Microprocessor without Interlocked Pipeline Stages*)**:** Se refiere a una familia de microprocesadores, estos son usados en las más distintas computadoras, desde uso cientifico hasta las embebidas en consolas de videojuegos.

Las computadoras que poseen esta arquitectura, son programadas en el lenguaje ensamblador MIPS. El gran punto a favor que tienen lenguajes de bajo nivel como éste, es que nos permiten dominar el diseño y el funcionamiento de cada una de las partes de los procesadores con esta arquitectura.

* **Registros**: MIPS posee registros, los cuales sirven para almacenar y cargar direcciones de memoria u otros registros, también operaciones básicas como operaciones aritméticas y de control. Posee entre otros los siguientes registros:
  + 32 registros en la CPU, cada uno de 32 bits.
  + 32 registros en la unidad de coma flotante, cada uno de 32 bits.
  + Un contador de programa (PC) de 32 bits, que indica, al principio de cada ciclo, la dirección de memoria de la instrucción del programa que se va a ejecutar.
  + Dos registros de 32 bits para multiplicaciones y divisiones (HI y LO).
* **Camino de datos:** El camino de datos se le denomina al conjunto de elementos que componen una CPU, ya sea sus buses de memoria, unidades aritmeticas logicas entre otras. Y se hace referencia a que es un camino, ya que, se puede describir como un mapa por donde se moverán las distintas instrucciones que asignemos a un procesador (Ilustración 4).

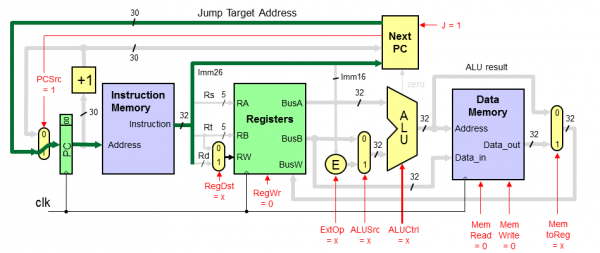


Ilustración 4: Camino de datos

* **Procesador monociclo:** La caracteristica más relevante para este proyecto de esste procesador es que el período del reloj se ajusta con la ruta crítica combinacional, que es la establecida por la instrucción más lenta en completar sus transferencias.

# Desarrollo

El desarrollo de este programa fue realizado en el lenguaje C, en el cual se usaron diferentes tipos de estructuras y funciones, estas funciones realizan una parte acotada del programa ya que se usó el principio de “divide y vencerás” para llevar a cabo el desarrollo del mismo, de esta forma, la sinergia de funciones logra la solución esperdada. A continuanción se detallarán las funciones más relevantes:

* **char\* crear\_tablero\_defecto():** función que crea un arreglo de caracteres que representa el tablero del juego, este tablero se crea por defecto como uno de 9 posiciones.
* **char\* crear\_tablero(FILE \*archivo):** función que crea un arreglo de caracteres que representa el tablero del juego, este tablero se crea según las posiciones que hayan sido indicadas en las instrucciones MIPS.
* **jugador\* crear\_jugador(FILE \*archivo):** función que crea una estructura jugador, el cual tiene como atributos el registro MIPS con el que fue indicado en las instrucciones y el simbolo que usará en la partida a modo de identificador.
* **char\* operacion\_addi(FILE \*archivo, char\* tablero, jugador \*jugador1, jugador \*jugador2):** función que recrea la operación addi de ensamblador y suma a las etapas en las cuales corresponda una unidad para indicar por donde pasaron las instrucciones, esta función añade el simbolo de un jugador al tablero.
* **char\* operacion\_subi(FILE \*archivo, char\* tablero, jugador \*jugador1, jugador \*jugador2):** función que recrea la operación subi de ensamblador y suma a las etapas en las cuales corresponda una unidad para indicar por donde pasaron las instrucciones, esta función borra el simbolo de un jugador en el tablero.
* **char\* operacion\_sw(FILE \*archivo, char\* tablero, jugador \*jugador1, jugador \*jugador2):** función que recrea la operación sw de ensamblador y suma a las etapas en las cuales corresponda una unidad para indicar por donde pasaron las instrucciones, esta función añade el simbolo de un jugador al tablero.
* **void generar\_archivos\_de\_salida(char\* tablero,jugador \*jugador1, jugador \*jugador2):** función que genera los dos archivos con los resultados de la partida y con los contadores de las etapas del camino de dato.

# Experimentos a realizar

A continuación se muestra la operación exitosa del programa. El primer paso a seguir es abrir la consola y una vez ubicado en la carpeta contenedora de los archivos del programa, escribir lo siguiente “make” (Ilustración 5).

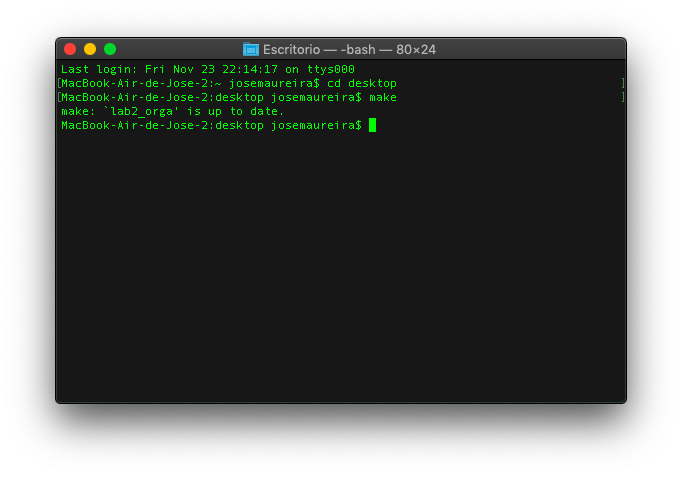


Ilustración 5: Experimento.1

Luego escribir “./lab\_orga2.c” y el programa ya estará corriendo (Ilustración 6).

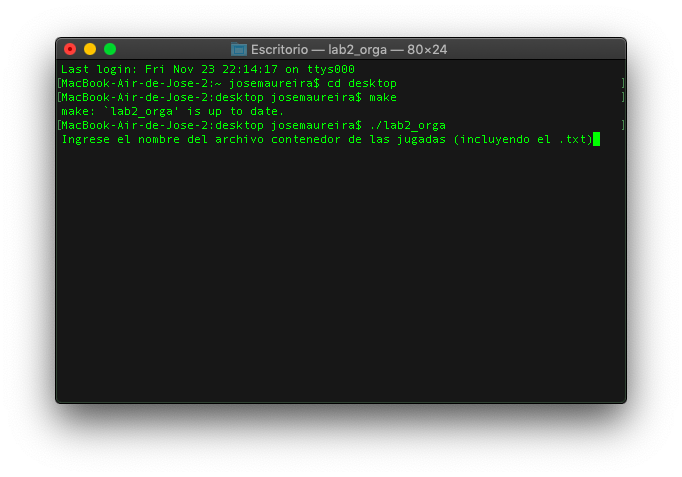


Ilustración 6: Experimento.2

Debido a que fue creado con usabilidad, deberá indicar el nombre del archivo de instrucciones (Ilustración 7).

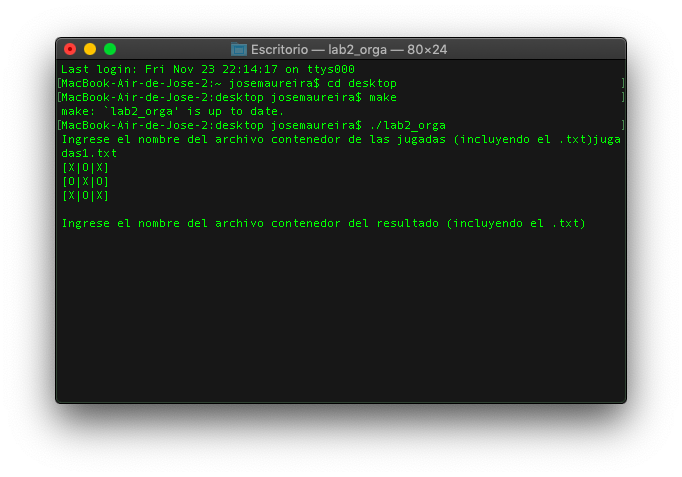


Ilustración 7: Experimento.3

Por la misma razón anterior se debe indicar el nombre de los archivos de salida (Ilustración 8).

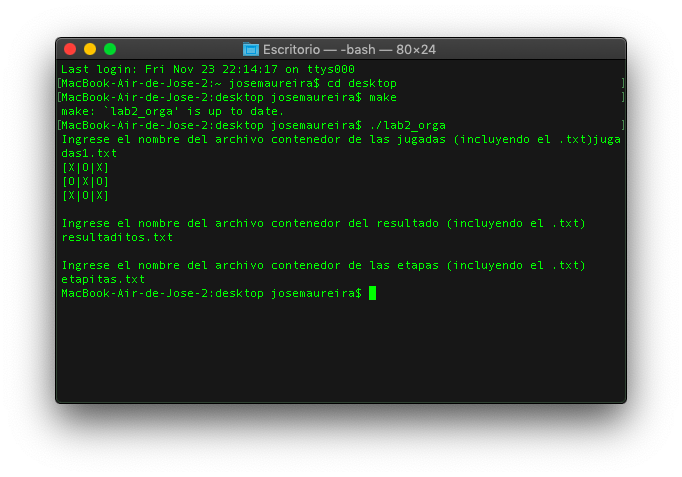


Ilustración 8: Experimento.4

## Resultados

El programa ha finalizado Los archivos de salida son los siguientes (Ilustración 9).

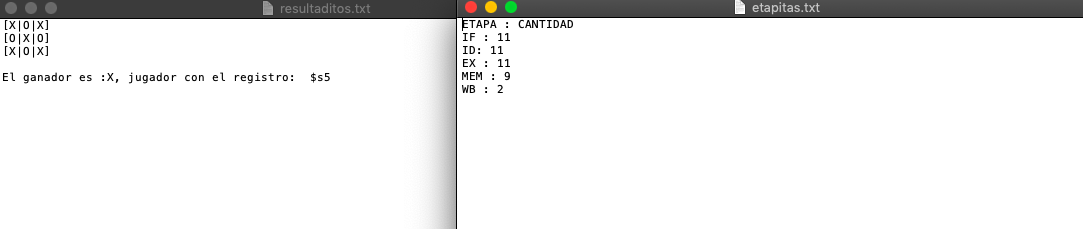


Ilustración : Resultados experimento

## Análisis de resultados

Midiendo los resultados entregados por el programa se puede constatar que cumple con emular las instrucciones de MIPS; addi, subi y sw a cabalidad. En el caso de cada una contabiliza las etapas que realizan estas instrucciones en el camino de datos de un procesador monociclo.

También logra jugar “TicTacToe” de manera óptima, pudiendose realizar triunfos por cualquiera de los dos jugadores y empates según las reglas del juego lo indican.

Logra mostrar en sus archivos de salida tanto el tablero con las jugadas realizadas, el ganador del juego, su registro y también mostrar cuantas veces pasó por cada etapa el conjunto de instrucciones MIPS.

Por todo lo anterior se tiene un análisis de resultados positivo, ya que, cumple con todas las funcionalidades esperadas.

# Conclusión

Teniendo en cuenta el positivo análisis de resultados y constatando que cumple con el objetivos general:

* Desarrollar un programa que simule el juego “TicTacToe” mediante instrucciones MIPS en un archivo de texto que será recibido como entrada y que genere archivos de texto capaces de representar el resultado del juego, el ganador, el registro del ganador y un conteo de las etapas del camino de dato usadas.

Y también cumpliendo con cada uno de los objetivos especificos:

* Traducir instrucciones de un lenguaje de bajo nivel como MIPS a uno de más alto nivel como C (estándar ANSI C).
* Poder tener un registro de cuantas veces las instrucciones pasaron por las etapas del camino de datos de un procesador monociclo.
* Usar estructura de datos para almacenar registros.
* Desarrollar el programa con usabilidad.
* Desarrollar el programa cumpliendo un mínimo de calidad de software.

Además se complen a cabalidad las siguientes exigencias del proyecto de laboratorio:

* Junto con el código fuente se debe encontrar un archivo makefile, válido para la compilación del programa.
* Los procedimientos y funciones poseen nombres que son representativos.

En resumen se concluye que el proyecto de laboratorio de la asignatura Organización de computadores se ha desarrollado de manera óptima, ya que, el programa cumple con todo lo requerido; los objetivos planteados y exigencias.

# Referencias

Patterson, D., Hennessy, J. and Alexander, P. (2013). *Computer organization and design*. 5th ed. Morgan Kaufmann.

Medina, L., Alvarez, R., Santana, B. and Undurraga, A. (2018). *Organización de Computadores*.[ebook]Santiago.Availableat: https://drive.google.com/file/d/16yOvRnZk3WEjrWXPOh\_Pe0l8YLBzQkZD/view [Accessed 20 Nov. 2018].