



**Herramienta de perfilado para una arquitectura basada en
RISC-V**

Manual de instalación y de usuario

Arturo Salas Delgado - 201229804

Manual de instalación

Las herramientas necesarias para la utilización de la herramienta de perfilado basada en una arquitectura basada en RISC-V son el Toolchain de RISC-V y un simulador llamada rv8. Además, se debe de descargar la herramienta de perfilado para poder ser usada. Los pasos para su instalación son los que se muestran a continuación:

Toolchain RISC-V

Para la instalación de este toolchain, que contiene el compilador de RISC-V para poder programar un procesador desde código C, se van a requerir de las siguientes bibliotecas:

- autoconf
- automake
- autotools-dev
- curl
- libmpc-dev
- libmpfr-dev
- libgmp-dev
- gawk
- build-essential
- bison
- flex
- texinfo
- gperf
- libtool
- patchutils
- git

El siguiente comando debe ser ejecutado en una terminal de Linux para la instalación de las bibliotecas mencionadas anteriormente:

```
$ sudo apt-get install autoconf automake autotools-dev curl libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils git
```

Se debe clonar el repositorio del toolchain de RISC-V mediante el siguiente comando:

```
$ git clone https://github.com/riscv/riscv-gnu-toolchain riscv-gnu-toolchain-rv32i
```

Una vez clonado el repositorio, se entra al directorio creado usando el siguiente comando:

```
$ cd riscv-gnu-toolchain-rv32i
```

Para verificar que ha sido clonado de manera correcta, se usa el siguiente comando:

```
$ git checkout bf5697a
```

Para la búsqueda de actualizaciones se usa el siguiente comando:

```
$ git submodule update --init --recursive
```

Y una vez completado los pasos anteriores, se procede a instalar la herramienta mediante el siguiente comando:

```
$ sudo ./configure --with-arch=rv32i
```

En caso de existir un error en la ejecución del comando anterior, se procede a ejecutar el siguiente comando:

```
$ sudo ./configure --with-xlen=32 --with-arch=I
```

Y para completar la instalación se ejecuta el siguiente comando:

```
$ sudo make -j$(nproc)
```

Una vez instalado el toolchain de RISC-V, se procede a la instalación de la herramienta elf2hex para el tratamiento de archivos .elf que contienen el ensamblado de un programa compilado. Se debe de clonar el repositorio mediante el uso del siguiente comando:

```
$ git clone https://github.com/riscv/riscv-fesvr riscv-fesvr
```

Se entra al nuevo directorio creado:

```
$ cd riscv-fesvr/
```

Se configura la herramienta mediante el uso del siguiente comando:

```
$ sudo ./configure
```

Y finalmente se procede a la instalación:

```
$ sudo make install
```

Simulador rv8

Además del toolchain de RISC-V, es necesario la instalación de una herramienta llamada rv8, el cual es un simulador basado en las diferentes arquitecturas de RISC-V para obtener métricas según el usuario desee.

El primer comando a ejecutar para la instalación del simulador es el siguiente:

```
$ export RISCV=/opt/riscv/toolchain
```

Ahora se procede a clonar la herramienta mediante el uso del siguiente comando:

```
$ git clone https://github.com/rv8-io/rv8.git
```

Se entra al directorio creado:

```
$ cd rv8
```

Se procede a actualizar los submódulos de la herramienta usando el siguiente comando:

```
$ git submodule update --init --recursive
```

Se prepara la herramienta para su instalación:

```
$ make
```

Y se instala la herramienta usando el siguiente comando:

```
$ sudo make install
```

Herramienta de perfilado

Para el uso de la herramienta de perfilado, se debe de clonar el repositorio mediante el uso del siguiente comando:

```
$ git clone https://github.com/artsaldo/ProfilingTool.git
```

Se entra al directorio nuevo:

```
$ cd ProfilingTool
```

Y se configuran la compilación necesaria mediante la ejecución del siguiente comando:

```
$ make
```

Manual de Usuario

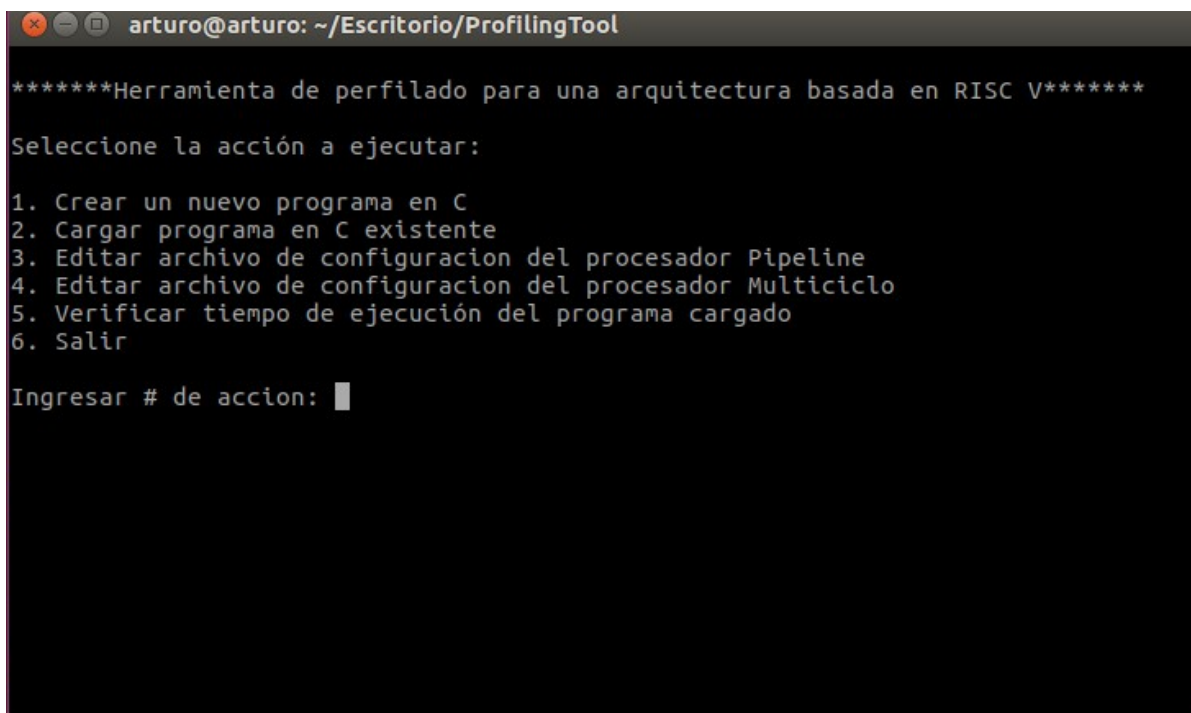
Para el uso de la herramienta de perfilado para una arquitectura basada en RISC-V se debe de abrir una terminal de linux e ingresar al directorio de la herramienta mediante el siguiente comando:

```
$ cd ProfilingTool
```

Una vez ahí, solamente se debe de ejecutar un script llamada tool.sh mediante el siguiente comando:

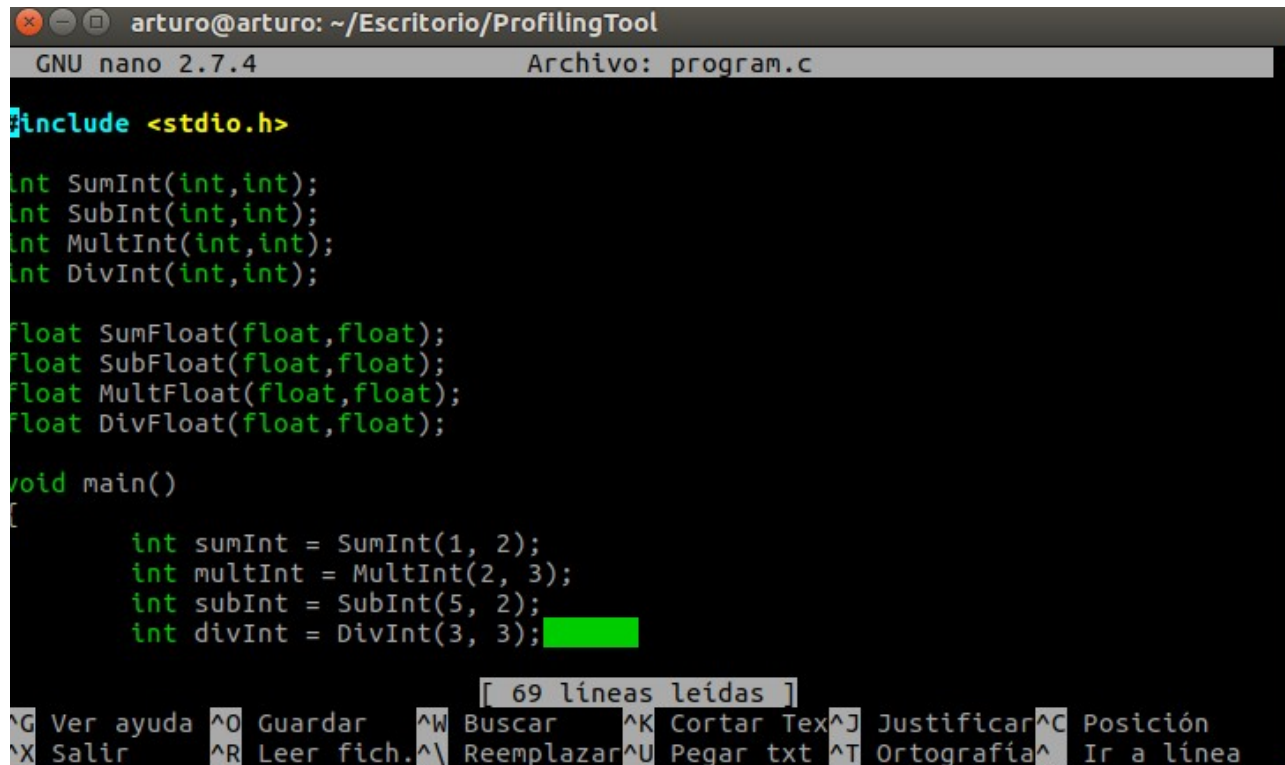
```
$ sh tool.sh
```

La terminal mostrará la siguiente interfaz de usuario:

A screenshot of a terminal window titled 'arturo@arturo: ~/Escritorio/ProfilingTool'. The terminal displays a menu for the 'Herramienta de perfilado para una arquitectura basada en RISC V'. The menu options are: 1. Crear un nuevo programa en C, 2. Cargar programa en C existente, 3. Editar archivo de configuracion del procesador Pipeline, 4. Editar archivo de configuracion del procesador Multiciclo, 5. Verificar tiempo de ejecución del programa cargado, and 6. Salir. Below the menu, it prompts 'Ingresar # de accion:' with a cursor. The terminal background is black with white text.

```
arturo@arturo: ~/Escritorio/ProfilingTool
*****Herramienta de perfilado para una arquitectura basada en RISC V*****
Seleccione la acción a ejecutar:
1. Crear un nuevo programa en C
2. Cargar programa en C existente
3. Editar archivo de configuracion del procesador Pipeline
4. Editar archivo de configuracion del procesador Multiciclo
5. Verificar tiempo de ejecución del programa cargado
6. Salir
Ingresar # de accion: █
```

Si se desea crear un nuevo programa en C para ser simulado, se debe de ingresar la opción #1 al sistema, donde se habilitará la edición de un programa llamado program.c que podrá ser simulado por la herramienta después. Se edita el archivo, se guarda pulsando “ctrl + O” y se sale de la edición pulsando “ctrl + X”. Una vez seleccionado se mostrará lo siguiente:



```
arturo@arturo: ~/Escritorio/ProfilingTool
GNU nano 2.7.4 Archivo: program.c

#include <stdio.h>

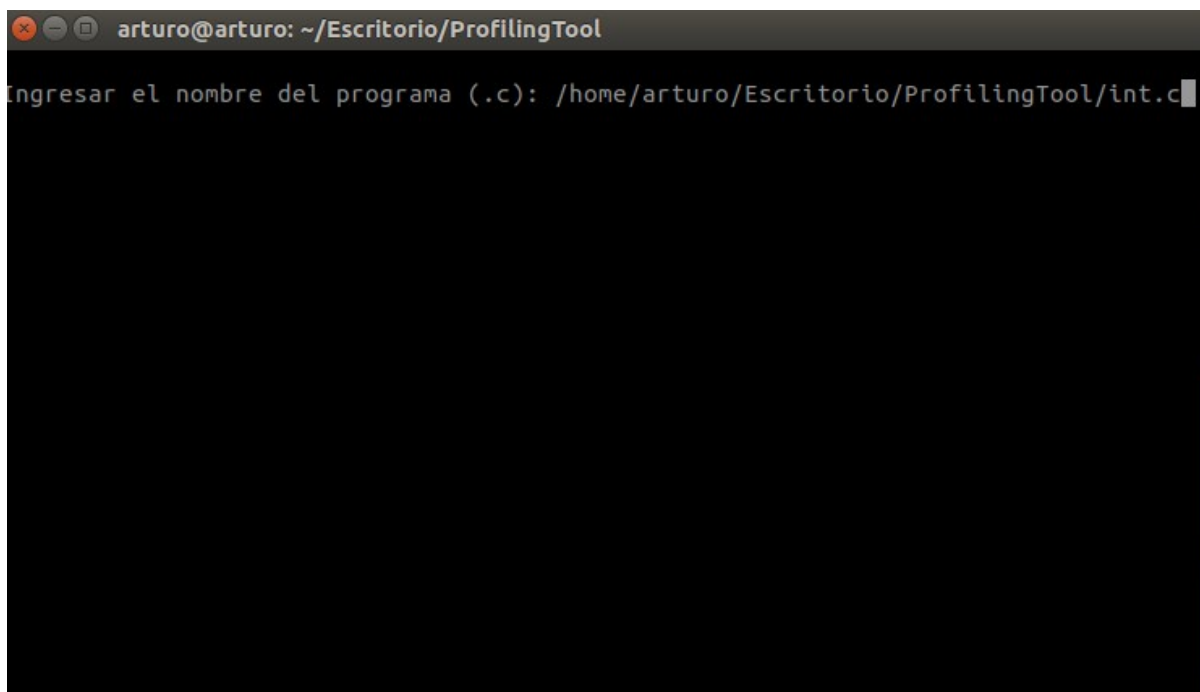
int SumInt(int,int);
int SubInt(int,int);
int MultInt(int,int);
int DivInt(int,int);

float SumFloat(float,float);
float SubFloat(float,float);
float MultFloat(float,float);
float DivFloat(float,float);

void main()
{
    int sumInt = SumInt(1, 2);
    int multInt = MultInt(2, 3);
    int subInt = SubInt(5, 2);
    int divInt = DivInt(3, 3);

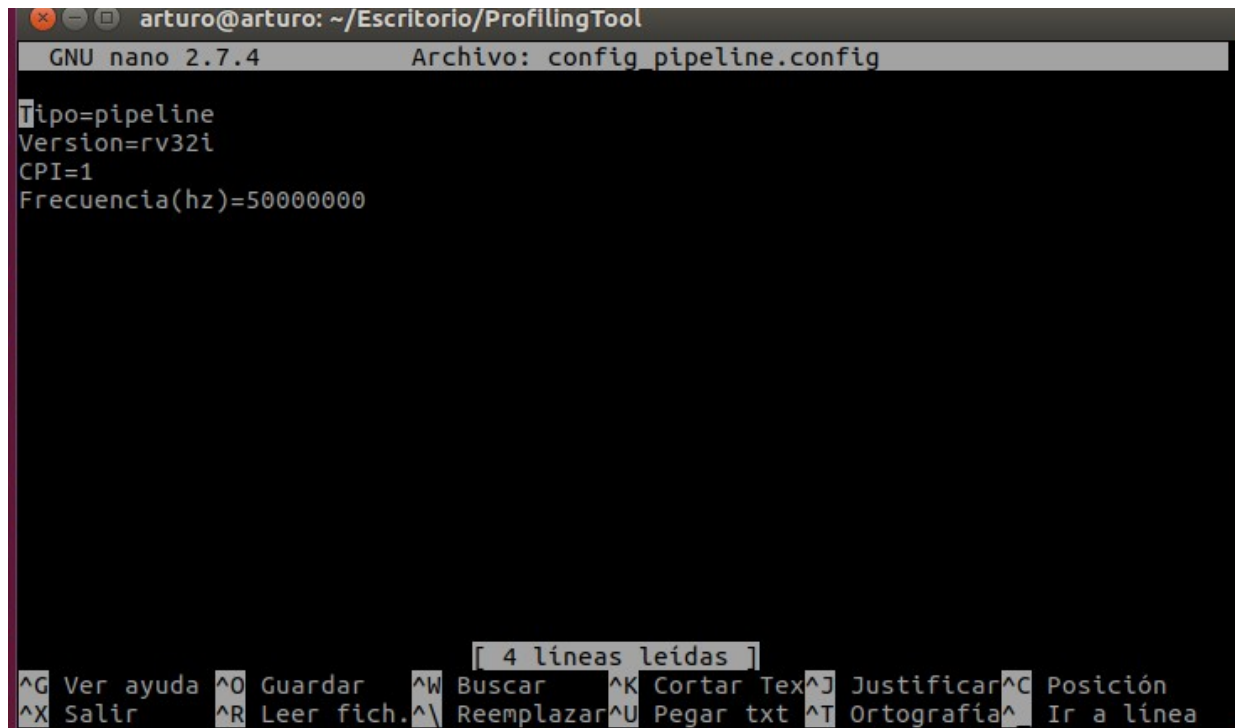
    [ 69 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

En caso de tener un programa en C existente y funcional, solamente se debe de seleccionar la opción #2 e ingresar el path donde se encuentra el archivo .c, tal como se muestra a continuación:



```
arturo@arturo: ~/Escritorio/ProfilingTool
Ingresar el nombre del programa (.c): /home/arturo/Escritorio/ProfilingTool/int.c
```

Si se desea editar el archivo de configuración para una arquitectura basada en pipeline, se debe de ingresar la opción #3. La edición de este archivo es tal como se muestra a continuación:



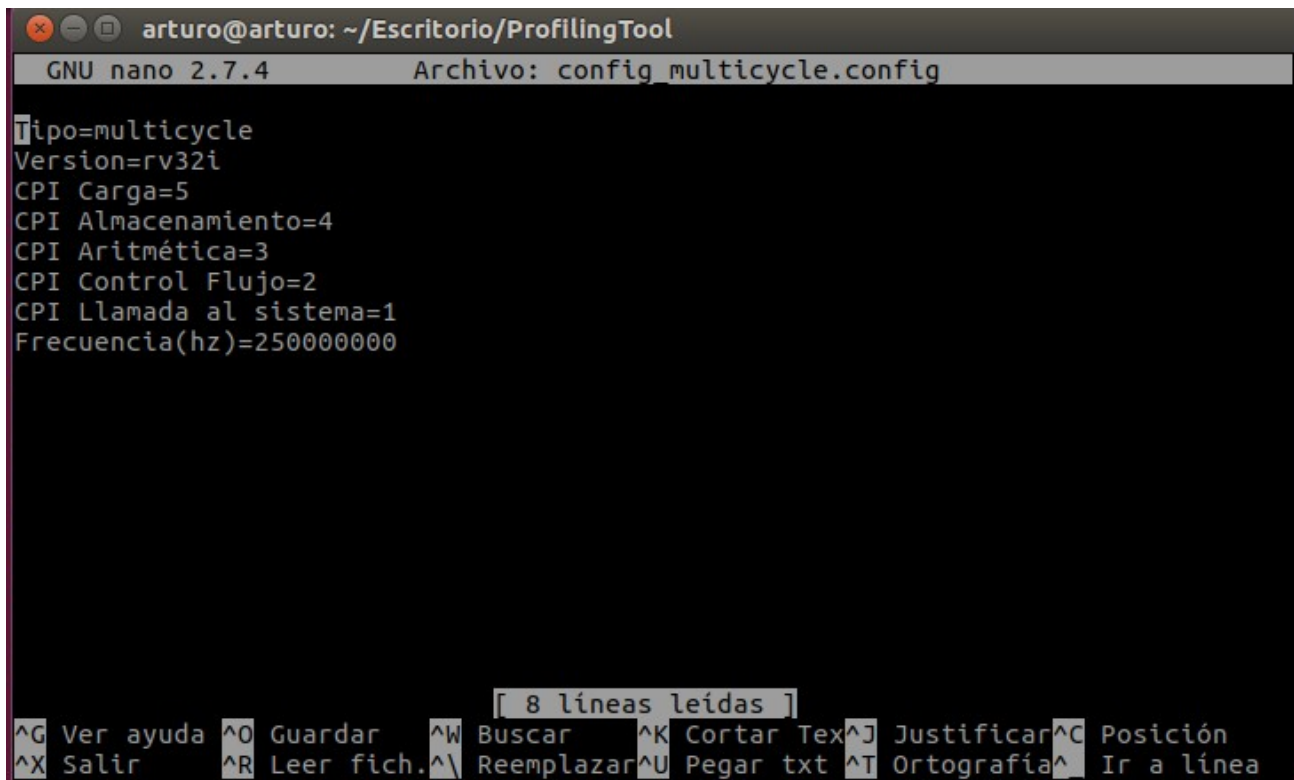
```
arturo@arturo: ~/Escritorio/ProfilingTool
GNU nano 2.7.4 Archivo: config_pipeline.config

Tipo=pipeline
Version=rv32i
CPI=1
Frecuencia(hz)=50000000

[ 4 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Nota: Es necesario que el archivo de configuración tenga el formato mostrado anteriormente, en caso de no ser así, la aplicación no obtendrá las métricas necesarias para la simulación. Además, el archivo correspondiente a esta configuración se encuentra en el directorio raíz con el nombre de config_pipeline.config.

Si se desea editar el archivo de configuración para una arquitectura basada en multicycle, se debe de ingresar la opción #4. La edición de este archivo es tal como se muestra a continuación:



```
arturo@arturo: ~/Escritorio/ProfilingTool
GNU nano 2.7.4 Archivo: config_multicycle.config

Tipo=multicycle
Version=rv32i
CPI Carga=5
CPI Almacenamiento=4
CPI Aritmética=3
CPI Control Flujo=2
CPI Llamada al sistema=1
Frecuencia(hz)=250000000

[ 8 líneas leídas ]
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^T Ortografía ^_ Ir a línea
```

Nota: Es necesario que el archivo de configuración tenga el formato mostrado anteriormente, en caso de no ser así, la aplicación no obtendrá las métricas necesarias para la simulación. Además, el archivo correspondiente a esta configuración se encuentra en el directorio raíz con el nombre de config_multicycle.config.

Y una vez completado el programa en C y los archivos de configuración, se procede a la simulación para obtener tiempos de ejecución seleccionando la opción #5. El sistema mostrará la opción de seleccionar la arquitectura sobre la cual se desea basar la simulación, tal como se muestra a continuación:


```
arturo@arturo: ~/Escritorio/ProfilingTool
Sobre cuál arquitectura desea simular?

1. Pipeline
2. Multiciclo

Ingrese la opcion: █
```

Para simular sobre una arquitectura pipeline, se debe de ingresar la opción #1. Para simular sobre una arquitectura multiciclo, se debe ingresar la opción #2. Los tiempos de ejecución resultantes se guardan en un archivo llamado `timeResults.txt` ubicado en la raíz del directorio. Además, el set de instrucciones que podrá ser ejecutado por una arquitectura basada en RISC-V se encuentra en un archivo llamado `executableBinary.txt` ubicado en la raíz del directorio. También, los tiempos resultantes se muestran en la terminal tal como se muestra a continuación:

```
arturo@arturo: ~/Escritorio/ProfilingTool
Sobre cuál arquitectura desea simular?
1. Pipeline
2. Multiciclo

Ingrese la opcion: 1

***** EXECUTION TIME *****

Execution time = 0.011060 ms

*****

Listo, ver el archivo timeResults.txt!!!!
```

```
arturo@arturo: ~/Escritorio/ProfilingTool
Sobre cuál arquitectura desea simular?
1. Pipeline
2. Multiciclo

Ingrese la opcion: 2

***** EXECUTION TIME *****

Execution time = 0.006392 ms

*****

Listo, ver el archivo timeResults.txt!!!!
```

Una vez ejecutada la simulación en el menú principal se debe de seleccionar la opción #6 para salir de la herramienta de perfilado.