

Universidad de Buenos Aires - FIUBA
66.20 Organización de Computadoras
Trabajo práctico 1: Programación MIPS
1^{er} cuatrimestre de 2021

\$Date: 2021/05/08 22:49:22 \$

1. Objetivos

Familiarizarse con el conjunto de instrucciones MIPS y el concepto de ABI, extendiendo un programa que resuelva el problema descrito a continuación.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes, un informe confeccionado de acuerdo con lo que mencionaremos en la sección 6, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

El trabajo y eventuales reentregas deberán ser presentadas exclusivamente a través del campus según las fechas preestablecidas para el mismo en la planificación del curso presentada en la primera clase del cuatrimestre. Asimismo todas las devoluciones se realizarán en horario de clase, en persona, para lo cual deberán estar presentes todos los integrantes.

4. Descripción

El programa a desarrollar deberá procesar un *stream* de texto compuesto por una cantidad arbitrariamente grande de líneas de longitud arbitraria.

La entrada del programa estará compuesta por una secuencia de pares de números enteros. Luego de leer estos pares de números, deberá calcular e imprimir el máximo común divisor (GCD) de cada par siguiendo los lineamientos indicados más abajo.

Por ejemplo, dado el siguiente flujo de entrada:

```
$ cat input.txt
60 45
90 9
17 13
```

Al ejecutar el programa la salida será:

```
$ tp1 -i input.txt -o -
GCD(60, 45) = 15
GCD(90, 9) = 9
GCD(17, 13) = 1
```

4.1. Ejemplos

Primero, usamos la opción `-h` para ver el mensaje de ayuda:

```
$ tp1 -h
Usage:
  tp1 -h
  tp1 -V
  tp1 -i in_file -o out_file
Options:
  -V, --version      Print version and quit.
  -h, --help         Print this information and quit.
  -i, --input        Specify input stream/file, "-" for stdin.
  -o, --output        Specify output stream/file, "-" for stdout.
Examples:
  tp1 < in.txt > out.txt
  cat in.txt | tp1 -i - > out.txt
```

A continuación, ejecutamos algunas pruebas: primero, veamos qué sucede cuando el archivo de entrada está vacío,

```
$ ./tp1 -o salida.txt </dev/null
$ ls -l salida.txt
-rw-r--r-- 1 leandro leandro 0 Oct 20 20:14 salida.txt
```

Aquí puede verse que el programa se comporta según lo esperado, ya que cuando la entrada está vacía, la salida lo estará también.

Veamos qué ocurre al ingresar un archivo con una única línea, la cual contiene un único par de valores:

```
$ echo 1 1 | ./tp1 -o -
1
```

Lo mismo debería ocurrir si la entrada se encuentra alojada en el sistema de archivos:

```
$ echo 1 1 >entrada.txt
$ ./tp1 -i entrada.txt -o -
1
```

5. Implementación

El programa a desarrollar constará de una mezcla entre código MIPS32 y C, siendo la parte escrita en *assembly* la encargada de calcular los GCDs utilizando el algoritmo de Euclides. El formato de dicha función será:

```
void euclides(struct gcd *, size_t);
```

En donde el primer parámetro, de tipo `struct gcd *`, es un puntero a un arreglo de estructuras que describen los números de la entrada, mientras que el segundo parámetro es la cantidad de elementos presentes en el arreglo.

```
struct gcd {  
    int num_a; /* input */  
    int num_b; /* input */  
    int gcd_ab; /* output */  
};
```

De esta manera, con una única invocación a `euclides()`, el programa podrá calcular y retornar todos los GCDs usando el arreglo pasado por parámetro.

6. Informe

El informe deberá incluir:

- Documentación relevante al diseño e implementación del programa.
- Comando(s) para compilar el programa.
- Las corridas de prueba, con los comentarios pertinentes.
- El código fuente, en lenguaje C y MIPS.
- El código MIPS32 generado por el compilador¹.
- Este enunciado.

7. Regresiones

El programa deberá pasar todas las regresiones definidas en el código fuente suministrado en este TP [1]:

```
$ make  
cc -Wall -g -o regressions regressions.c gcd.c gcd.S  
:  
$ make test  
./regressions
```

Asimismo deberá usarse el modo 1 del sistema operativo para manejo de acceso no alineado a memoria [2].

¹Por motivos prácticos, en la copia impresa sólo es necesario incluir la primera página del código assembly MIPS32 generado por el compilador.

8. Entrega de TPs

La entrega de este trabajo deberá realizarse usando el campus virtual de la materia [3] dentro del plazo de tiempo preestablecido. Asimismo, en todos los casos, estas presentaciones deberán ser realizadas durante los días martes. El *feedback* estará disponible de un martes hacia el otro, para lo cual deberán estar presentes los integrantes de cada grupo tal como ocurre durante la modalidad presencial de cursada.

Por otro lado, la última fecha de entrega y presentación para esta trabajo será el martes 18/5.

Referencias

- [1] Código fuente para realizar el Trabajo Práctico.
https://drive.google.com/file/d/1iTAF_7Dyyp0JHvZk5DHICd4aRd_ZcveX/view
- [2] Controlling the kernel unalignment handling via debugfs,
<https://www.linux-mips.org/wiki/Alignment>.
- [3] Aula Virtual - Organización de Computadoras 86.37/66.20 - Curso 1 - Turno Martes.
<https://campus.fi.uba.ar/course/view.php?id=649>