

# 66:20 Organización de computadoras

## Trabajo práctico 1: Conjunto de instrucciones MIPS.

### 1. Objetivos

El objetivo de este trabajo es familiarizarse con la programación en assembler MIPS y con la ABI(Application Binary Interface).

### 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.

Además, es necesario que el trabajo práctico incluya (entre otras cosas, ver sección 8), la presentación de los resultados obtenidos, explicando, cuando corresponda, con fundamentos reales, las causas o razones de cada resultado obtenido.

El informe deberá respetar el modelo de referencia que se encuentra en el grupo<sup>1</sup>, y se valorarán aquellos escritos usando la herramienta  $\text{T}_{\text{E}}\text{X}$  /  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ .

### 4. Recursos

Usaremos el programa GXemul [1] para simular el entorno de desarrollo que utilizaremos, una máquina MIPS corriendo una versión reciente del sistema operativo NetBSD [2].

En la clase del 11/4 hemos repasado, brevemente, los pasos necesarios para la instalación y configuración del entorno de desarrollo.

### 5. Descripción.

En el presente trabajo práctico, se deben implementar en lenguaje C dos algoritmos de ordenamiento: *Bubblesort*[3] y *Shellsort*[4], que deben ordenar palabras en un archivo. Una vez implementados, procederemos a realizar mediciones para evaluar el desempeño relativo entre ambas implementaciones, utilizando el programa `time`[5].

---

<sup>1</sup><http://groups.yahoo.com/group/orga6620>

---

También se deberá proveer una versión en assembler MIPS del algoritmo de *Shellsort*, y comparar su desempeño con el de la versión en C.

## 6. Implementación.

El programa debe leer los datos de entrada desde *stdin* o bien desde uno o más archivos. La salida del programa debe imprimirse por *stdout*, mientras que los errores deben imprimirse por *stderr*. El algoritmo de ordenamiento puede seleccionarse mediante las opciones **-b** o **-s** para *bubblesort* o *shellsort* respectivamente. La implementación de *shellsort* debe consistir de una función con el siguiente prototipo:

```
void shellsort (char ** words, int arraysize)
```

donde *words* es un puntero a una lista de punteros a caracter, que son las palabras contenidas en el archivo, y *arraysize* es el tamaño de esta lista. La confección de esta lista corre por cuenta del programa en C.

### 6.1. ABI

La implementación en assembler MIPS debe respetar la ABI usada por la cátedra[6], que difiere de la utilizada por el GCC.

Mostramos el mensaje de ayuda mediante la opción **-h**.

```
$tp0 -h
tp0 [OPTIONS] [file...]
-h, --help          display this help and exit.
-V, --version       display version information and exit.
-m, --bubble        use the bubblesort algorithm.
-s, --sel           use the shellsort algorithm.

$echo -n echo "El tractorcito rojo que silbo y bufo" > entrada.txt
$tp1 -b entrada.txt
bufo El que rojo silbo tractorcito y

$cat letters.txt
aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ$

$tp0 letters.txt
aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ
$

$tp0 letters.txt entrada.txt
aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ bufo El que rojo silbo tractorcito y
$
```

---

## 7. Mediciones.

En el ambiente MIPS emulado, utilizar `time` para tomar el tiempo que tardan ambos algoritmos en ordenar los siguientes archivos: `alice.txt`, `beowulf.txt`, `cyclopedia.txt`, `elquijote.txt`. Graficar para cada algoritmo el tiempo insumido contra el tamaño de muestra. Graficar el speedup de *Bubblesort* contra *Shellsort* para los diversos tamaños de archivo. Graficar el speedup de *Shellsort* versión C contra *Shellsort* versión assembler MIPS para los diversos tamaños de archivo.

## 8. Informe.

Se debe entregar:

- Informe describiendo el desarrollo del trabajo práctico.
- CD conteniendo todo el material digital.
- Código fuente.
- Este enunciado.

## 9. Fechas de entrega.

- Primera entrega: Jueves 2 de Mayo.
- Revisión: Jueves 9 de Mayo.
- Vencimiento: Jueves 16 de Mayo.

## Referencias

- [1] GXemul, <http://gavare.se/gxemul/>.
- [2] The NetBSD project, <http://www.netbsd.org/>.
- [3] Bubble-sort [http://en.wikipedia.org/wiki/Bubble\\_sort](http://en.wikipedia.org/wiki/Bubble_sort)
- [4] Shell-sort [http://en.wikipedia.org/wiki/Shell\\_sort](http://en.wikipedia.org/wiki/Shell_sort)
- [5] time man page <http://unixhelp.ed.ac.uk/CGI/man-cgi?time>
- [6] func\_call\_conv.pdf en <http://groups.yahoo.com/group/orga6620/files/>.