

66:20/86.37 Organización de Computadoras
Primer cuatrimestre de 2018
Trabajo práctico 0: Infraestructura básica

1. Objetivos

Familiarizarse con las herramientas de software que usaremos en los siguientes trabajos, implementando un programa (y su correspondiente documentación) que resuelva el problema piloto que presentaremos más abajo.

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, 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 en este y otros trabajos prácticos, una máquina MIPS corriendo una versión reciente del sistema operativo NetBSD [2].

En éste utilizaremos los programas `gcc` [3] y `time` para compilar y para examinar el desempeño de los programas a desarrollar.

5. El comando `wc`

El comando de Unix `wc` [4] toma como entrada un archivo o `stdin`, y cuenta las palabras, las líneas y la cantidad de caracteres que contiene.

6. Programas a desarrollar

El programa a escribir, en lenguaje C, recibirá un nombre de archivo que contiene texto (o el archivo mismo por `stdin`) e imprimirá por `stdout` la cantidad de líneas, palabras y caracteres que contiene, junto con el nombre del archivo.

6.1. Ejemplos

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

```
$ tp0 -h
Usage:
  tp0 -h
  tp0 -V
  tp0 [options] file
Options:
  -V, --version      Print version and quit.
  -h, --help         Print this information.
  -l, --lines Print number of lines in file.
  -w, --words Print number of words in file.
  -c, --chars Print number of characters in file.
  -i, --input        Path to input file.
Examples:

  tp0 -w -i input.txt
```

Luego, lo usamos con un pequeño fragmento de texto:

```
$echo -n "El tractorcito rojo que silbo y bufo" > entrada.txt
$tp0 -w -i entrada.txt
7 entrada.txt
$
```

7. Mediciones

Se deberá medir el tiempo insumido por el programa para el caso de los archivos `alice.txt`, `beowulf.txt`, `cyclopedia.txt` y `elquijote.txt`. Graficar el tiempo insumido contra el tamaño de muestra. Se deberá comprobar que el programa acepta las opciones dadas, y que reporta un error ante situaciones anómalas (como no encontrar el archivo solicitado). La ejecución del programa debe realizarse bajo el entorno MIPS.

8. Informe

El informe deberá incluir:

- Documentación relevante al diseño e implementación del programa;
- Las corridas de prueba, con los comentarios pertinentes;
- Gráficos de tiempo completo para los diferentes archivos, en función del tamaño de la entrada.
- El código fuente, en lenguaje C;
- Este enunciado.

9. Fecha de entrega

El cierre de este trabajo práctico es el jueves 5 de abril de 2018.

Referencias

- [1] GXemul, <http://gavare.se/gxemul/>
- [2] The NetBSD project, <http://www.netbsd.org/>
- [3] GCC, <https://gcc.gnu.org/onlinedocs/gcc-3.3.6/gcc/>
- [4] Manual de `wc`, <http://man7.org/linux/man-pages/man1/wc.1.html>