

66:20 Organización de Computadoras

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 6), 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 grupo¹, y se valorarán aquellos escritos usando la herramienta \TeX / \LaTeX .

4. Recursos

Usaremos el programa GXemul [2] 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 [3].

En la clase del 20/8 hemos repasado, brevemente, los pasos necesarios para la instalación y configuración del entorno de desarrollo.

¹<http://groups.yahoo.com/group/orga-comp>

5. Implementación

5.1. Programa

El programa, a escribir en lenguaje C, es una versión del comando `rev` de UNIX. El mismo concatena y escribe en `stdout` el contenido de uno o mas archivos, invirtiendo el orden de los caracteres de cada línea. En nuestro caso, se asume que cada caracter mide 1 byte.

Retorna 0 sólo en caso de éxito. Todos los errores deben ser impresos por `stderr`.

5.1.1. Parámetros

El programa debe recibir como parámetros los archivos sobre los que va a trabajar. En caso de no recibir ningún parámetro, entonces deberá operar sobre los datos provenientes de `stdin`. Debido a que el tamaño de la información a recibir es desconocida, debe utilizarse memoria dinámica. Cualquier error debe ser informado debidamente.

5.2. Ejemplos

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

```
$ tp0 -h
Usage:
  tp0 -h
  tp0 -V
  tp0 [file...]
Options:
  -V, --version      Print version and quit.
  -h, --help         Print this information and quit.
Examples:
  tp0 foo.txt bar.txt
  tp0 gz.txt
```

A continuación, ejecutamos algunas pruebas:

```
$ cat > file1
Esta es la primera línea de file1
y esta es la segunda.

$ ./tp0 file1
1elif ed aeníl aremirp al se atsE
.adnuges al se atse y

$ cat > file2
Ahora es el turno
de file2.

$ ./tp0 return.txt status.txt
1elif ed aeníl aremirp al se atsE
.adnuges al se atse y
onrut le se arohA
.2elif ed

$ cat file1 | ./tp0
1elif ed aeníl aremirp al se atsE
.adnuges al se atse y

$ ./tp0 file1 | rev
Esta es la primera línea de file1
y esta es la segunda.
```

5.3. Portabilidad

Como es usual, es necesario que la implementación desarrollada provea un grado mínimo de portabilidad. Para satisfacer esto, el programa deberá funcionar al menos en NetBSD/pmax (usando el simulador GXemul [2]) y la versión de Linux (Knoppix, RedHat, Debian, Ubuntu) usada para correr el simulador, Linux/i386.

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;
- El código MIPS32 generado por el compilador;
- Este enunciado.

7. Fechas

- Entrega: 10/9/2013.
- Última revisión: 17/9/2013.
- Vencimiento: 24/9/2013.

Referencias

- [1] NetBSD rev manual page. <http://netbsd.gw.com/cgi-bin/man-cgi?rev+1.NONE+NetBSD-current>.
- [2] GXemul, <http://gavare.se/gxemul/>.
- [3] The NetBSD project, <http://www.netbsd.org/>.