

Universidad de Buenos Aires - FIUBA
66.20 Organización de Computadoras
Trabajo práctico 0: Infraestructura básica
1^{er} cuatrimestre de 2015

\$Date: 2015/03/23 13:41:08 \$

1. Objetivos

Familiarizarse con las herramientas de software que usaremos en los siguientes trabajos, implementando un programa y su correspondiente documentación que resuelvan el problema descripto 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, un informe impreso de acuerdo con lo que mencionaremos en la sección 6, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

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].

Durante la primera clase del curso presentaremos brevemente los pasos necesarios para la instalación y configuración del entorno de desarrollo.

5. Implementación

5.1. Programa

El programa, a escribir en lenguaje C, es una versión minimalista del comando `tac` de UNIX. El mismo concatena y escribe en `stdout` el contenido de uno o mas archivos en reversa,

invirtiendo el orden de las líneas.

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

5.1.1. Parámetros

El programa recibe como parámetros los archivos de entrada sobre los que va a trabajar. En caso contrario, deberá operar sobre los datos provenientes de `stdin`. El tamaño y estructura de los archivos de entrada es arbitrario; y el programa deberá utilizar memoria dinámica para realizar la transformación de líneas. 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 > return.txt
Upon successful completion a value of 0 is returned.  Otherwise, a value
of -1 is returned and errno is set to indicate the error.
```

```
$ tp0 return.txt
of -1 is returned and errno is set to indicate the error.
Upon successful completion a value of 0 is returned.  Otherwise, a value
```

```
$ cat >status.txt
The status information word st_flags has the following bits:
```

Constant	Description
UF_NODUMP	do not dump a file
UF_IMMUTABLE	file may not be changed

```
$ tac return.txt status.txt
of -1 is returned and errno is set to indicate the error.
Upon successful completion a value of 0 is returned.  Otherwise, a value
```

UF_IMMUTABLE	file may not be changed
UF_NODUMP	do not dump a file
Constant	Description

The status information word `st_flags` has the following bits:

```
$ cat return.txt | tp0
of -1 is returned and errno is set to indicate the error.
Upon successful completion a value of 0 is returned.  Otherwise, a value
$
```

Por último, si tenemos disponible el diccionario de palabras en `/usr/share/dict/words` (una palabra por línea, todas las líneas terminadas con `\n`):

```
$ cat words | md5sum
cbbcded3dc3b61ad50c4e36f79c37084 -
$ tac words | tac | md5sum
cbbcded3dc3b61ad50c4e36f79c37084 -
```

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

Fecha de vencimiento: martes 7/4/2015.

Referencias

- [1] tac (Wikipedia). [http://en.wikipedia.org/wiki/Tac_\(Unix\)](http://en.wikipedia.org/wiki/Tac_(Unix)).
- [2] GXemul, <http://gavare.se/gxemul/>.
- [3] The NetBSD project, <http://www.netbsd.org/>.