

Trabajo Práctico X

Título del informe

Nombre Apellido, *Padrón xxxxx*
mail@mail.com

Nombre Apellido, *Padrón xxxxx*
mail@mail.com

1er. Cuatrimestre de 2019

86.37 / 66.20 Organización de Computadoras – Práctica Jueves
Facultad de Ingeniería, Universidad de Buenos Aires

Resumen

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1. Introducción

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```
-V, --version  
-h, --help  
-i, --ignore-case
```

2. Diseño e Implementación

2.1. Análisis de la línea de comandos

Las opciones válidas ingresadas por línea de comandos como fue mencionado, pueden ser:

```
-V, --version  
-h, --help  
-i, --ignore-case
```

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```
const char *const short_opt = "hvi";
const struct option long_opt[] = {
    {"help", 0, NULL, 'h'},
    {"version", 0, NULL, 'v'},
    {"ignore-case", 0, NULL, 'i'},
    {NULL, 0, NULL, 0}
};
```

- COD_HELP 0: Se solicitó la ayuda por medio de la opción -h.
- COD_VERSION 1: Se solicitó la versión por medio de la opción -v.

2.2. Desarrollo del Código Fuente

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1. **optfunctions.h**: Funciones para el manejo de las opciones.
2. **line.h**: Funciones útiles para el manejo de cadenas de caracteres.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

3. Proceso de Compilación

Se cuenta con un archivo Makefile que indica las reglas de compilación ejecutadas por el comando make.

El código fuente se compila ejecutando el comando make con alguna de las posibles reglas. En todos los casos se compila con los flags de warnings y de debbuging prendidos.

make tp: Crea el ejecutable.

make tp.mips: Genera el código de máquina para MIPS.

make all: Crea el ejecutable join.

Además se cuenta con las siguientes reglas adicionales.

indent: Indenta los archivos fuentes que componen nuestra aplicación.

Para eliminar todos los archivos generados por el comando make, se puede ejecutar con la regla clean de la forma:

make clean

4. Portabilidad

Dado que se utilizó el lenguaje de programación C, sin hacer uso de funciones específicas de sistemas operativos, o de bibliotecas comerciales, los programas pueden ser compilados en varios de ellos. De todas formas, el Makefile presentado fue hecho particularmente para sistemas de tipo UNIX.

Los casos de prueba presentados fueron llevados a cabo en una arquitectura MIPS emulada. El programa de emulación utilizado fue GXemul y el sistema operativo en la máquina emulada fue NetBSD.

Los programas también fueron probados en Ubuntu-Linux corriendo en una arquitectura Intel x86 satisfactoriamente.

5. Casos de Prueba

A continuación mostraremos los casos básicos de prueba realizados a la aplicación.

5.1. Línea de comandos

Ejecutamos la aplicación sin parámetros.

```
# ./tp ; echo $?
join: missing operand
Try 'tp --help' for more information.
3
```

Pasamos una opción inválida.

```
./tp invalid ; echo $?
Error: opening file invalid
6
```

No pasamos un argumento cuando la opción lo requiere.

```
./tp - ; echo $?
join: missing operand after '-'
Try 'join --help' for more information.
4
```

Tratamos de abrir un archivo inexistente como entrada.

```
# ls -l null
ls: null: No such file or directory

# ./tp - null
Error: opening file null
6
```

Tratamos de utilizar dos veces stdin.

```
# ./tp - - ; echo $?
Error: stdin can be used only once
5
```

5.2. Funcionamiento de la aplicación

Finalmente mostramos el caso de prueba proporcionado en el enunciado.

Salida del caso de prueba

Dado que trabajamos con memoria dinámica, tenemos que verificar que no haya fugas de memoria, ni ningún otro error posible. Para ello utilizamos la herramienta Valgrind en la máquina Host, en este caso es una máquina Linux con Ubuntu.

```
$ uname -a
Linux dell 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i686 GNU/Linux

~/6620-tp0$ valgrind -v --tool=memcheck --leak-check=yes ./join-6620 -i ApellidoNombre Puntaje

[...]% \newpage

==12692== HEAP SUMMARY:
==12692==      in use at exit: 0 bytes in 0 blocks
==12692==    total heap usage: 24 allocs, 24 frees, 1,144 bytes allocated
==12692==
==12692== All heap blocks were freed -- no leaks are possible
==12692==
==12692== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 12 from 7)
--12692--
--12692-- used_suppression:      12 dl-hack3-cond-1
==12692==
==12692== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 12 from 7)
```

6. Conclusión

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Referencias

- [1] GXemul. <http://gavare.se/gxemul/>.
- [2] The NetBSD project. <http://www.netbsd.org/>.
- [3] Vim the Editor. <http://www.vim.org/>.
- [4] B. Kernighan, D. Ritchie, *The C Programming Language*. Person Prentice Hall
- [5] B. Kernighan, R. Pike, *The Unix Programming Environment*. Prentice Hall
- [6] Valgrind, valgrind-3.6.0.SVN-Debian, <http://valgrind.org/>

A. Enunciado

B. Código fuente