

BUSCADOR DE FICHEROS (BUFI)

Sergio Esteban Tarrero

LABORATORIO DE REDES Y SISTEMAS OPERATIVOS - 2ºZ Curso 2019-2020

Introducción:

BUFI es un **BU**scador de **F**icheros creado y desarrolla en Linux (CentOS 7), después de tener un par de ideas sobre el proyecto final de la asignatura, al final decidí crear un buscador de ficheros que funcione por SHELL, dejando atrás las otras ideas que debido a falta de conocimiento del lenguaje de programación no pude realizar (Keylogger y controlador de Google Drive / Dropbox por línea de comandos).

Finalmente me decanté por este proyecto y he conseguido desarrollar gran parte de lo que quería hacer al principio.

Desarrollo:

El proyecto se ha desarrollado en CentOS 7 desde el principio, una vez avanzado el proyecto realicé pruebas en otras máquinas virtuales de Linux (Debian 9 y Ubuntu 18.01) y no funcionaban correctamente, solo en CentOS 7 me ha funcionado el proyecto de manera correcta, por lo que realicé todo el desarrollo del programa en esta partición de Linux.

Los principales problemas con los que me he encontrado han sido la diferente forma del uso de variables dentro del uso de Bash dentro de los editores de texto (VScode y Vim) debido a que ciertas variables no se expandían, he tenido que investigar la causa de esto y finalmente encontré una solución óptima dentro del código (Ej: En vez de poner \${FICHERO} tenía que poner "\$fichero" para que la máquina me lo reconociese correctamente.

El programa contiene principalmente un contenido para una dirección de fichero y un par de switch case para especificar la opción de búsqueda, dentro de que está bastante revisado el control de errores.

Instrucciones:

Forma de usar el programa:

bash bufi.sh <directorio> <opción de búsqueda> <opción extra> <imprimir (opcional)>

Ejemplo: bash bufi.sh /home/sergio/descargas/pruebas -t -d -print

→ Comandos del programa <opción de búsqueda> <opción extra> :

-t) Caso buscar por **TIPO DE ARCHIVO**:

- **-t -f** : Caso buscar por tipo de archivo y mira solo los ficheros normales
- **-t -d** : Caso buscar por tipo de archivo y mira solo los directorios
- **-t -** : Caso buscar por tipo de archivo y muestra todo
- **-t -*** : Caso de error

-n) Caso buscar por **NOMBRE DE ARCHIVO**

-p) Caso buscar por **PERMISOS DE ARCHIVO**

- **-p -x** : Caso buscar por permisos de archivo y que el usuario pueda escribir y leer
- **-p -r** : Caso buscar por permisos de archivo y que el usuario pueda leer
- **-p -w** : Caso buscar por permisos de archivo y que el usuario pueda escribir
- **-p -*** : Caso de error

***)** Todo lo demás

→ Comandos del programa <imprimir> :

"-print"): Imprimir la búsqueda realizada

"-exec"): Ejecutar el fichero después de buscarlo (no funciona)

***)** Todas las **demás opciones**

Entonces en el ejemplo dado:

bash bufi.sh /home/sergio/descargas/pruebas -t -d

Busca en el directorio **/home/sergio/descargas/pruebas** por **TIPO DE ARCHIVO** y **MIRA SOLO LOS DIRECTORIOS**

→ Casos de error:

- **Error 1:** “El directorio introducido no existe”
- **Error 2:** “Solo se pueden introducir los comandos ‘-f’, ‘-d’ y ‘-’ “
- **Error 3:** “Solo se pueden introducir los comandos ‘-t’, ‘-n’ y ‘-w’ “
- **Error 4:** “Solo se pueden introducir los comandos ‘-x’, ‘-r’, ‘-p’ y ‘-c’ “
- **Error 5:** “Solo se pueden introducir el comando ‘-print’ “

```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
[sergio@localhost profi]$ bash bufi.sh /home/sergio -t -
/home/sergio/awsd
/home/sergio/Descargas
/home/sergio/Desktop
/home/sergio/LRS0_Z_Actividad_1_Sergio_Esteban
/home/sergio/LRS0_Z_Actividad_1_Sergio_Esteban.zip
/home/sergio/LRS0_Z_Actividad_2_Sergio_Esteban
/home/sergio/practicass
/home/sergio/profi
/home/sergio/proyectofinal
/home/sergio/Proyecto Final Laboratorio
```

Caso buscar por tipo de archivo y muestra todo

```
[sergio@localhost profi]$ bash bufi.sh /home/sergio -t -f
/home/sergio/LRS0_Z_Actividad_1_Sergio_Esteban.zip
```

Caso buscar por tipo de archivo y mira solo los ficheros normales

```
[sergio@localhost profi]$ bash bufi.sh /home/sergio -t -x
ERROR 2: Solo se pueden introducir los comandos -f, -d y -
Forma de usar el programa:
<directorío> <opción de búsqueda> <opción extra> <acción>
Ejemplo: /home/sergio/downloads/pruebas -t -d
Ctrl + l para limpiar la shell
IMPORTANTE: Todos los comandos introducidos deben estar escritos en minúsculas
```

Caso de error, muestra los comandos que se deben introducir para que el programa funcione correctamente

Control de errores:

```
[sergio@localhost profil]$ bash bufi.sh /home/sergioooo -t -x  
ERROR 1: El directorio introducido no existe, pruebe otro
```

Caso de error 1 – El directorio introducido no existe

```
[sergio@localhost profil]$ bash bufi.sh /home/sergio -t -x  
ERROR 2: Solo se pueden introducir los comandos -f, -d y -
```

Caso de error 2 – Control de errores en <opción extra>

```
[sergio@localhost profil]$ bash bufi.sh /home/sergio -p -a  
ERROR 3: Solo se pueden introducir los comandos -t, -n y -w
```

Caso de error 3 - Control de errores en <opción extra>

```
[sergio@localhost profil]$ bash bufi.sh /home/sergio -a  
ERROR 4: Solo se pueden introducir los comandos -x, -r, -p y -c
```

Caso de error 4 - Control de errores en <opción de búsqueda>

```
[sergio@localhost profil]$ bash bufi.sh /home/sergio -t - -aa  
ERROR 5: Solo se pueden introducir los comandos -print o -exec
```

Caso de error 5 - Control de errores en < imprimir (opcional) >

Conclusiones:

Como conclusión final, me ha gustado conocer un lenguaje de programación nuevo, aunque no pude realizar todas las cosas que me hubiesen gustado hacer, como introducir un comando **-exec** para ejecutar un fichero en concreto al poner el comando o añadir una interfaz gráfica más intuitiva, ha sido un proyecto de investigación bastante útil.