Asignatura: Sistemas Operativos

~Práctica 2~

Programación de un intérprete de mandatos (Minishell)

Autores:

- *Gonzalo Martín Jiménez - 100333375 - Grupo 80*
- *Manuel Del Sol Nova - 100329912 - Grupo 80*

- Índice de contenidos -

- Descripción del código

El minishell programado nos proporciona la resolución de diferentes mandatos que pongamos por pantalla, se puden realizar tanto comandos simples, como comandos concatenados conectados por pipes, en la que la información guardada en un mandato pasa al siguiente.

Dentro del minishell también podemos realizar dos mandatos internos especiales, mycalc y mybak, creados en el mismo código.

El mandato interno mycalc realiza la suma o módulo (según se pase por argumento) de dos enteros pasados por argumentos. Éste muestra por pantalla la suma/módulo de ambos con su respectivo resultado junto con la variable de entorno “acc”, en la que se guardan la suma de los resultados de las distintas operaciones (solo suma) que se hayan hecho.

El mandato interno mybak copia el contenido de un fichero pasado por argumento en otro fichero que se crea en la ruta pasada por argumento, con el mismo tamaño que el original, si el fichero ya existe en la ruta de destino, se borra su contenido truncándolo a 0. El nuevo fichero tendrá permisos tanto de lectura como escritura para el usuario y el grupo.

- Batería de pruebas

Mandato interno mycalc

-Operación correcta de add-

msh> mycalc 1 add 3
[OK] 1 + 3 = 4; Acc = 4

-Operación correcta de add con actualización de acc-

msh> mycalc 43 add 13
[OK] 43 + 13 = 56; Acc = 60

-Operación correcta de mod-

msh> mycalc 12 mod 45
[OK] 12 % 45 = 45 * 0 + 12

-Número de argumentos invalido-

msh> mycalc 12 mod
[ERROR] La estructura del comando es <operando 1> <add/mod> <operando 2>

-Estructura del comando errónea-

msh> mycalc 7 23 5
[ERROR] La estructura del comando es <operando 1> <add/mod> <operando 2>

Mandato interno mybak

-Copia de fichero nueva-

msh> mybak y.c /home/gonza/Escritorio
[OK] Copiado con éxito el fichero y.c a la carpeta /home/gonza/Escritorio

-Fichero que ya existe en la ruta destino-

msh> mybak y.c /home/gonza/Escritorio
El fichero y.c ya existe en /home/gonza/Escritorio; El fichero y.c se ha truncado a 0
[OK] Copiado con exito el fichero y.c a la carpeta /home/gonza/Escritorio

-Estructura del comando errónea-

msh> mybak y.c
[ERROR] La estructura del comando es mybak <fichero origen> <directorio destino>

-Fichero erroneo-

msh> mybak asd.xzcasd /home/gonza/Escritorio
[ERROR] Error al abrir el archivo origen

-Ruta indefinida-

```
msh> mybak y.c
[ERROR] Error al abrir el fichero destino
```

-Ruta inexistente-

```
msh> mybak y.c /zdsf/hbd
La ruta /zdsf/hbd no existe
```

Mandatos simples

-Mandato simple sin argumentos correcto-

```
msh> ls
Makefile  msh.c  parser.o  scanner.l  y.c  y.tab.h
msh         msh.o  parser.y  scanner.o  y.o
```

-Mandato simple con argumentos correcto-

```
msh> ls -1
Makefile
msh
msh.c
msh.o
parser.o
parser.y
scanner.l
scanner.o
y.c
y.o
y.tab.h
```

-Mandato inexistente-

```
msh> l
[ERROR] Error en la ejecución del proceso hijo: No such file or directory
[ERROR] Ejecución anormal del proceso hijo: Success
```

-Redirección de entrada-

```
msh> cat < y.c
/*-
 * DO NOT MODIFY THIS FILE
 */

#include <stdio.h>

void yyerror(char *s)
{
        fprintf(stderr, "%s\n", s);
}

#undef yywrap

int yywrap()
{
        return 1;
}
```

-Redirección de entrada inexistente-

msh> cat < l
[ERROR] El archivo de redirección no existe: No such file or directory
[ERROR] Ejecución anormal del proceso hijo: Success

-Redirección de salida-
msh> ls > prueba

-Crea un fichero llamado prueba que contiene:-
Makefile
msh
msh.c
msh.o
parser.o
parser.y
prueba
scanner.l
scanner.o
y.c
y.o
y.tab.h

-Redirección de salida de error-
msh> l >& prueba2
[ERROR] Ejecución anormal del proceso hijo: Success

Crea un fichero llamado prueba2 que contiene:
[ERROR] Error en la ejecución del proceso hijo: No such file or directory

-Operación con una tubería de dos argumentos-
msh> ls | more
Makefile
msh
msh.c
msh.o
parser.o
parser.y
scanner.l
scanner.o
y.c
y.o
y.tab.h

- Conclusiones

- Para algunas funciones implementadas en los mandatos internos se hecha en falta alguna que otra librería, ya que facilitarían bastante más a la hora de programarlos y quedaría la solución más elegante, sobre todo a la hora de manejar rutas, como la librería <dirent.h>.