

# Operating Systems System Calls: Minishell msh. 2020b

(C)FranciscoRosales,frosal@fi.upm.es

4deseptiembrede2020

## 1 Objetivo de la práctica

The student must design and code, in the C language and on the Unix operating system, a program that acts as a simple interpreter of shell commands. The program should follow strictly the specifications and requirements contained in this document.

Con la realización de este programa el alumno adquirirá valiosos conocimientos de programación en Unix environment. SIGNAL, PIPE, DUP, etc), como en el manejo de herramientas como el visualizador de páginas de manual (man), el compilador de C.

NOTE: While reading through this document, you will encounter the notation "man-s#xxxxx" which suggests the manual page number to obtain more information.

## 2 Descripción de la práctica

The minishell utilizes the standard input (descriptor 0), parses the commands that it receives and executes them. It also presents the results of the commands. 2) It also shows the special variables prompt by bgpid (see the special variables) as a way to notify the user of errors or of a library error.

Blank: A character that separates words.

Separator: A character with a special meaning (<>), the end of a line or the end of a file (keyboard Ctrl-D).

Text: Is any sequence of characters delimited by blank or separator.

Command: A sequence of text separated by blanks. The first text specifies the name of the command to execute. The remaining are the arguments of the invoked command.

Enumerate mandates pass as argument 0 (manexecvp). Each mandate runs as a direct child process of the minishell (manfork). The value of a command is stored in determination (man-s2wait). If the execution fails, notify the error (bye! estandar error).

Secuencia Es una secuencia de dos o más mandatos separados por el carácter '|'. La salida de cada mandato se conecta por un tubo normalmente espera la finalización del último comando de esta secuencia antes de solicitar la siguiente línea de entrada. El valor de una secuencia es el

Redirección La entrada y la salida de un mandato o secuencia pueden ser redirigidas a un archivo o a un dispositivo de salida. In case of any errors during redirects, notification (por la estandar error) yes suspende la ejecución del archivo.

```
< fichero Usafichero como entrada estandar abrir el archivo para lectura (manopen).
> fichero Usafichero como salida estandar. Si el fichero no existe se crea, si existe se
truncated (mancreat), modo de creación 0666
> & fichero Usafichero como estandar error. Si el fichero no existe se crea, si existe se
truncated (mancreat), modo de creación 0666
```

Background Un mandato o secuencia terminada en & supone la ejecución en segundo plano del mismo, esto es, el minishell no queda

Señales El minishell no lanza los mandatos en background, deben morir por señales de teclado (SIGINT, SIGQUIT) (man signal). Por contrast, los comandos lanzados en primer plano deben morir por señales de teclado, por lo tanto mantienen la acción default.

Mandato Interno Es aquel que bien se corresponde directamente con una llamada al sistema o bien es un complemento que ofrece el propio minishell

Todo mandato interno comprueba el número de argumentos con que se le invoca si encuentra este o cualquier otro error, lo reporta (por el estándar error) y termina con un código distinto de cero.

Los mandatos internos del minishell son:

cd [Directory] Cambia el directorio default (man-s2chdir). If Directory appears should change to it. If it does not appear, it changes to the directory specified in the variable de entorno HOME. Presenta (por la salida estandar) como resultado de la llamada al directorio actual de trabajo (mangetcwd) con el formato: "%s\n".

umask [Value] Changes the file creation mask

(man-s2umask). (por las salidas estándar) como resultado el valor de la actual máscara con el formato: "%o\n". Además, si aparece Valo

time[Mandate]Consumo de tiempo desde el juicio (man-s2times). Si aparece Manda-  
to, lo ejecutará e informará sobre su consumo. Sin argumentos, informará sobre el consumo  
del minishell and your children. Represents (on stdout) time in user mode,  
time in system mode and real time, in seconds and milliseconds, in the format:  
"%d.%03du %d.%03ds %d.%03dr\n"

readVariable[Variable...] Da valor a las variables de entorno no especificadas (mínimo una) (manputenv). Lee un carácter de la entrada esta

Metacharacters El minishell interpreta ciertos caracteres de una manera especial

way. that begins with ~ or that contains \$ undergoes the substitution of the expression that forms part of the escape sequence to its corresponding value.

~[Usuario] Un nombre breve de usuario tiene el formato "%[\_a-zA-Z0-9]" (manss-conf). Si aparece Usuario,  
se substituye por el directorio home de dicho usuario, follow-up passwd entry is indicated (mangetpwnam). Bell, is replaced by the  
value of a variable de entorno HOME.

\$Variable Un nombre breve de variable tiene el formato "%[\_a-zA-Z0-9]" (manss-  
conf). Se substituye por el valor de la variable de entorno Variable (mangetenv).

Variables especiales El minishell conoce y trata de forma especial algunas variables de en-  
torno (manputenvgetenv).

prompt Mensaje de apremio antes de leer cada línea. Por defecto será "msh>".

my pid Identificador de proceso del propio minishell.

bg pid Identificador de proceso del último proceso arrancado en background.

status Valor de terminación del último mandato o pipeline ejecutado en foreground.

Expansión de nombres de ficheros Antes de la ejecución de cada orden, cada texto es examinado en busca de caracteres comodín. Si aparece

? Casa con cualquier carácter individual.

Se establecen las siguientes restricciones. Si en el texto aparece también algún carácter /, no se tratará a los comodines. Esto reduce la tar

### 3 Códigos de apoyo

Para facilitar la realización de estas prácticas se dispone del fichero `sh.2020b.tgz` que contiene el código de apoyo. Cuando se extraiga el código de apoyo...

`Makefile` Fichero de apoyo para la herramienta `make`. No debe ser modificado. Con él se consigue la compilación automática de los ficheros fuente que se modifiquen.

`scanner.l` Fichero de apoyo para la herramienta `lex`. No debe ser modificado. Con él se genera automáticamente el código C que implementa un analizador léxico.

`parser.y` Fichero de apoyo para la herramienta `yacc`. No debe ser modificado. Con él se genera automáticamente el código C que implementa un analizador sintáctico.

`main.c` Fichero de apoyo de C que muestra cómo usar el parser. Este fichero es el que se DEBE MODIFICAR. It is recommended to study in detail for the correct understanding of the use of the function interface, obtain order. La versión que se ofrece hace el código de las líneas que son sintácticamente correctas. This feature must be deleted and substituted by the execution of the lines that are syntactically correct.

### 4 General recommendations

Develop the minishell in stages, progressively complicating it. Start by implementing a basic version. Continúa introduciendo las funcionalidades en el orden en que se describen en el apartado Descripción.

Para ello se le pide que tenga un documento de especificación de la información contenida, en concreto con los formatos de presentación de la información.

Para probar la práctica habrá que entregar la práctica. Se le contestará con un correo electrónico. La raza resultante puede darle información.

### 5 Documentación a entregar

LA ÚLTIMA VERSIÓN REGISTRADA DE LA PRÁCTICA ES LA ÚNICA CUYA EVALUACIÓN TIENE VALIDEZ. IT IS THE ONLY AND DEFINITIVE VALIDITY.

ALL PREVIOUS DELIVERY IS CONSIDERED THAT IT IS ONLY FOR DEBUGGING.

Seguía añadiendo nuevas funcionalidades, deberá entregarla usando el mandato `delivery.somsh.2020b`. This mandate will carry out the collection of the following files:

authors.txt Fichero con los datos de autor.

bitacora.txt Es conveniente llevar una memoria del desarrollo de la práctica.

main.c Código fuente del minishell, implementando todas las funcionalidades que se requieren.

## References

[Bou83] “The UNIX System”, S.R. Bourne Addison-Wesley, 1983.

[Roc85] “Advanced Unix Programming”, M.J. Rochkind  
Prentice-Hall, 1985.

[Sun90] “Programming Utilities and Libraries”, SUN Microsystems  
Sun Microsystems, 1990.