PROYECTO SHELL

PRESENTADO A:
JOHN ALEXANDER SANABRIA

CARLOS ESTEBAN MURILLO
1526857

JUAN CAMILO SÁNCHEZ
1527749

HERNEY QUINTERO
1528556

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
SISTEMAS OPERATIVOS
SANTIAGO DE CALI
MAYO 2018

Para la realización del shell en el lenguaje C, implementamos varios métodos y funciones los cuales repasamos a continuación, mostrando la solución al

Se implementan las siguientes variables globales:
instruction, instruction2, pip1, pip2 y **filename** son arreglos de caracteres usados para almacenar partes de un comando. para un comando simple se hace uso del arreglo *"instruction",* a la hora de encontrarse con un ">", la parte anterior y posterior a este símbolo se almacenan en *"instruction2"* y *"filename"* respectivamente. Para comandos con pipe, hace la separación en dos variables, donde lo que está antes del pipe se almacena en pip1 y lo posterior en pip2. para un comando combinado con pipe y el operador ">", la operación con pipe se ejecuta normalmente y el resultado se guarda en el archivo "filename".

En el shell se implementan los siguientes archivos y métodos:

Archivo mainShell.c

main: Este es el método principal donde se hace el llamado a todas las funciones heredadas por la librería "shell.c", aquí existe un bucle infinito donde se le piden los comandos al usuario, el programa borra los espacios en blanco y guarda cada palabra del comando en un archivo, después con el método Buscar Operador, verifica lo que se encuentre en el comando bien sea un pipe o un asignador de archivos. Después ejecuta el método correspondiente heredado por la librería antes mencionada.

Archivo shell.c

Remover Espacios: Este método no recibe ni retorna nada, pero modifica el arreglo inicial *"instruction"*, donde cada elemento de este es una palabra separada por espacios proveniente del String escaneado al principio, el cual recibe las entradas del usuario.

Buscar Operador: Este método recibe como entrada un String, el cual será utilizado para comparar cada parte del arreglo *"instruction"*, esto se realiza con el fin de saber si se encuentran caracteres como ">", "|".

Ejecutar: este método no recibe nada como parámetro, pero usa la llamada al sistema fork para ejecutar el comando almacenado en el arreglo instruction. Este método ejecuta los comandos más simples (ls, top, ps etc). En caso de que el comando no exista muestra un mensaje notificando.

Guardar en archivo: Este método no recibe ni retorna nada, se usa si se encuentra un ">" en el arreglo, mediante un ciclo guarda en instruction2 lo que está antes del ">" y en filename lo que se encuentra después del ">", luego ejecuta lo guardado previamente en instruction2 y se direcciona la salida a un archivo; para esto mediante el comando close cerramos el flujo de salida de datos por consola, y mediante open lo guardamos en un archivo, en el cual se usaron los permisos: `S_IRWXU|S_IXOTH`. El archivo tendrá por nombre el contenido de filename.

Pipe: Este método se encarga de almacenar en un buffer la ejecución del primer comando siendo esta la entrada del segundo comando. para esto se tiene un buffer del tamaño 2 el cual va utilizarse para almacenar las dos ejecuciones de los comandos. el comando “pipe()” lo que hace es preparar el buffer para que los datos que se encuentren en la primera posición sean la entrada del nuevo comando. Para que este funcione correctamente se emplea el método “dup2()” el cual duplica un registro mediante se hace el cambio del proceso padre a proceso hijo, es por esto que carga primero el comando del padre y después recibe de entrada el proceso del comando hijo.

Ejecutar Pipe: Inicialmente este comando separa lo que se encuentra antes y después del pipe “|”, guardandolos en los arreglos pip1 y pip2 respectivamente. después de esto crea un proceso con el método “fork()” y realiza la ejecución del método Pipe, anteriormente mencionado**.**

Pipe Archivo: Este método combina parte de lo que realiza el método “Guardar en archivo” y “Ejecutar Pipe”, donde separa cada parte del comando en 3 subarreglos, los pipes los almacena en pip1 y pip2, lo que queda despues del simbolo “>” se almacena en el filename. Al de cerrar el flujo por consola y redireccionar la salida al archivo se implementa el método Pipe para su funcionamiento.

Vaciar: Este método no recibe ni retorna nada, se encarga de inicializar los arreglos globales para prevenir usar residuos de comandos ejecutados anteriormente en nuevas ejecuciones

Archivo shell.h

Es el archivo de cabecera, y en él están definidos todos los métodos usados en el shell.c

Makefile

Este archivo contiene las instrucciones:

-clean: se encarga de limpiar del directorio de los archivos .a, .o, librerías creadas y el binario del shell.

-shell, libshell.a y shell.o: que se encargan de compilar e importar el shell como librería.

-ejecutar: que se encarga llamar a clean y shell, es decir, primero limpia el directorio para evitar que se hayan realizado cambios en los archivos y luego ejecuta la instrucción shell, finalmente ejecuta la línea ./shell, esto para que se ejecute el binario directamente.

Enlace a vídeo demo de la Shell

enlace