Proyecto Administración de SO

Dra. KARLA E. VÁZQUEZ ORTIZ
* CRISTIAN ALEXIS GÓNZALES QUIÑONES
* JENNY ARLETH PÉREZ VALDEZ
* CLAUDIA LIZBETH CARRIZALES PIÑA
UNIVERSIDAD POLITECNICA DE VICTORIA
ADMINISTRACIÓN DE SISTEMAS OPERATIVOS

1730048@upv.edu.mx, 1730035@upv.edu.mx, 1730117@upv.edu.mx

Lunes 06 de Agosto del 2018

ÍNDICE

RESUMEN

Linux es un sistema operativo: un conjunto de programas que permiten interactuar con el ordenador y ejecutar otros programas. La parte ms importante de un sistema operativo es el ncleo. En un sistema GNU/Linux, Linux es el ncleo.
El Shell es el programa-interface y es un script para la terminal que solo tenemos que agregar la extensión .sh que provee la comunicacin entre el usuario y el Sistema operativo, sus funciones son: Servir como traductor de comandos, recibe los comandos internos del mismo sistema operativo y Shell se encarga de su ejecucin, en cambio si hablamos de comandos realizados por el usuario entre la funcin del kernel, el los ejecuta y se encarga de procesar sus rdenes.

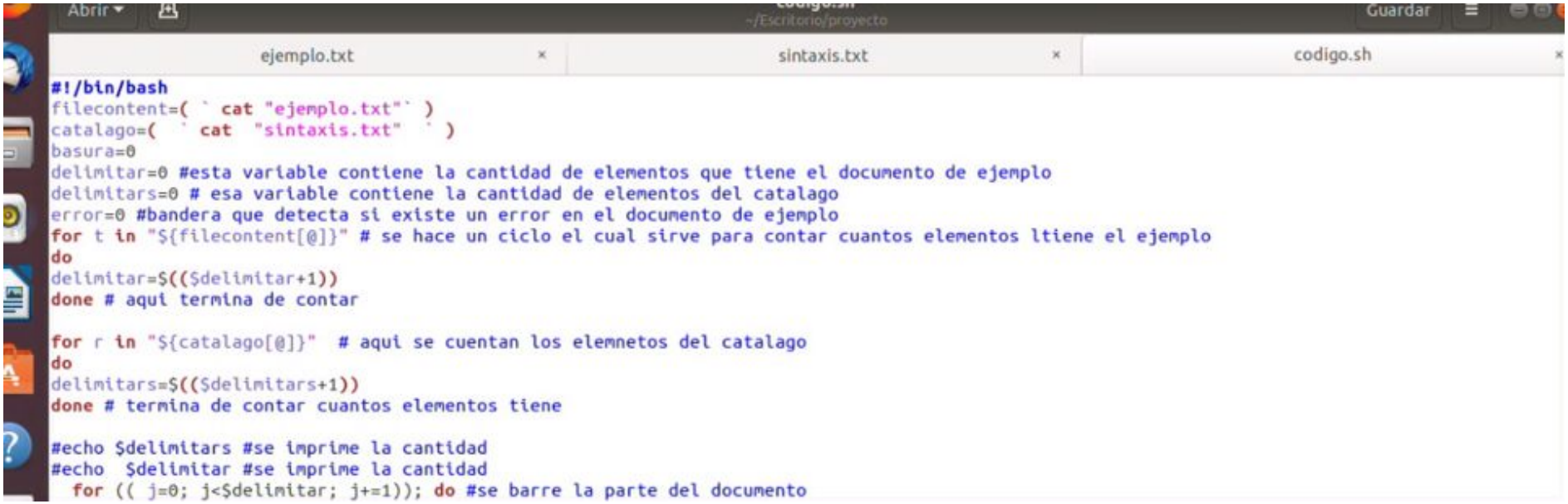
INTRODUCCIÓN

Este reporte tiene el propsito de dar una breve y detallada explicacin sobre lo que realizamos en nuestro cdigo, mostrando fragmentos de Cdigo de nuestro Script Shell. En la consola, los comandos son analizados y ejecutados por el Shell, o interprete de comandos. Existen muchos intrpretes de comandos distintos en Linux, los ms utilizados son el sh y el bash. Cada uno de ellos se diferencia del anterior en que mejora y complementa las rdenes existentes y aade nuevas posibilidades. Nosotros solamente utilizaremos Shell (.sh) en este proyecto. El proyecto lo que realiza es recibir un archivo de texto con la sintaxis propuesta, al realizar esto, por consiguiente, a travs de validaciones y algoritmos, se identifican los errores de la misma sintaxis que no concuerden con la sintaxis ya propuesta. En caso de que no haya errores en la sintaxis se ejecuta el programa y mostraremos la salida.

DESARROLLO

En esta parte del código cargamos lo que son los archivos de texto en donde ejemplo.txt corresponde a codigo que se va a traducir en base a la sintaxis propuesta y se guarda en el vector de filecontent y en sintaxis.txt es donde se tiene el catálogo de comandos propuestos por el equipo y se guarda en el vector catálogo se tiene una variable llamada basura la cual hace la funcion de omitir pasos de impresión en consola u otros procesos en la parte de la validación de la sintaxis la variable delimitar y delimitars sirven para guardar cuantas palabras se han almacenado en el vector y con estas variables se limitan los ciclos en donde barremos los ciclos y asi evitar problemas la variable de error nos sirve para la hora de validar la sintaxis darnos cuenta de que hay un error y evitar correr el programa en seguida se hacen ciclos en donde la limitante son los vectores de catalago y filecontent en ellos se cuenta la cantidad de palabras de cada 1.

DESARROLLO


Abrir
~/Escritorio/proyecto
Guardar
ejemplo.txt
sintaxis.txt
codigo.sh
#!/bin/bash
filecontent=(` cat "ejemplo.txt"`)
catalago=(` cat "sintaxis.txt" `)
basura=0
delimitar=0 #esta variable contiene la cantidad de elementos que tiene el documento de ejemplo
delimitars=0 # esa variable contiene la cantidad de elementos del catalago
error=0 #bandera que detecta si existe un error en el documento de ejemplo
for t in "${filecontent[@]}" # se hace un ciclo el cual sirve para contar cuantos elementos ltiene el ejemplo
do
delimitar=$(($delimitar+1))
done # aqui termina de contar
for r in "${catalago[@]}" # aqui se cuentan los elemnetos del catalago
do
delimitars=$(($delimitars+1))
done # termina de contar cuantos elementos tiene
#echo $delimitars #se imprime la cantidad
#echo $delimitar #se imprime la cantidad
for ((j=0; j<$delimitar; j+=1)); do #se barre la parte del documento

DESARROLLO

En esta parte se muestra un fragmento del bloque de validación de sintaxis y lo que se hace es que verifica si lo que va leyendo concuerda con la sintaxis propuesta por ejemplo en esta parte se esta validadndo lo que es el comando escribir el cual si todo esta correcto se traducira por echo si en dado caso se escribe mal lo escribir marcara error y se encenderá la variable bandera error y esto evitara que el programa se ejecute a menos que el error se corrija se tocan varios puntos clave cómo lo es la impresión de variables y para ello se deberia poner el cáracter $ y se debera escribir el nombre de la variable si el usuario ingreso mal en nombre de la variable el compilador marcara el error y se encenderá o mantendrá la bandera error encendida.

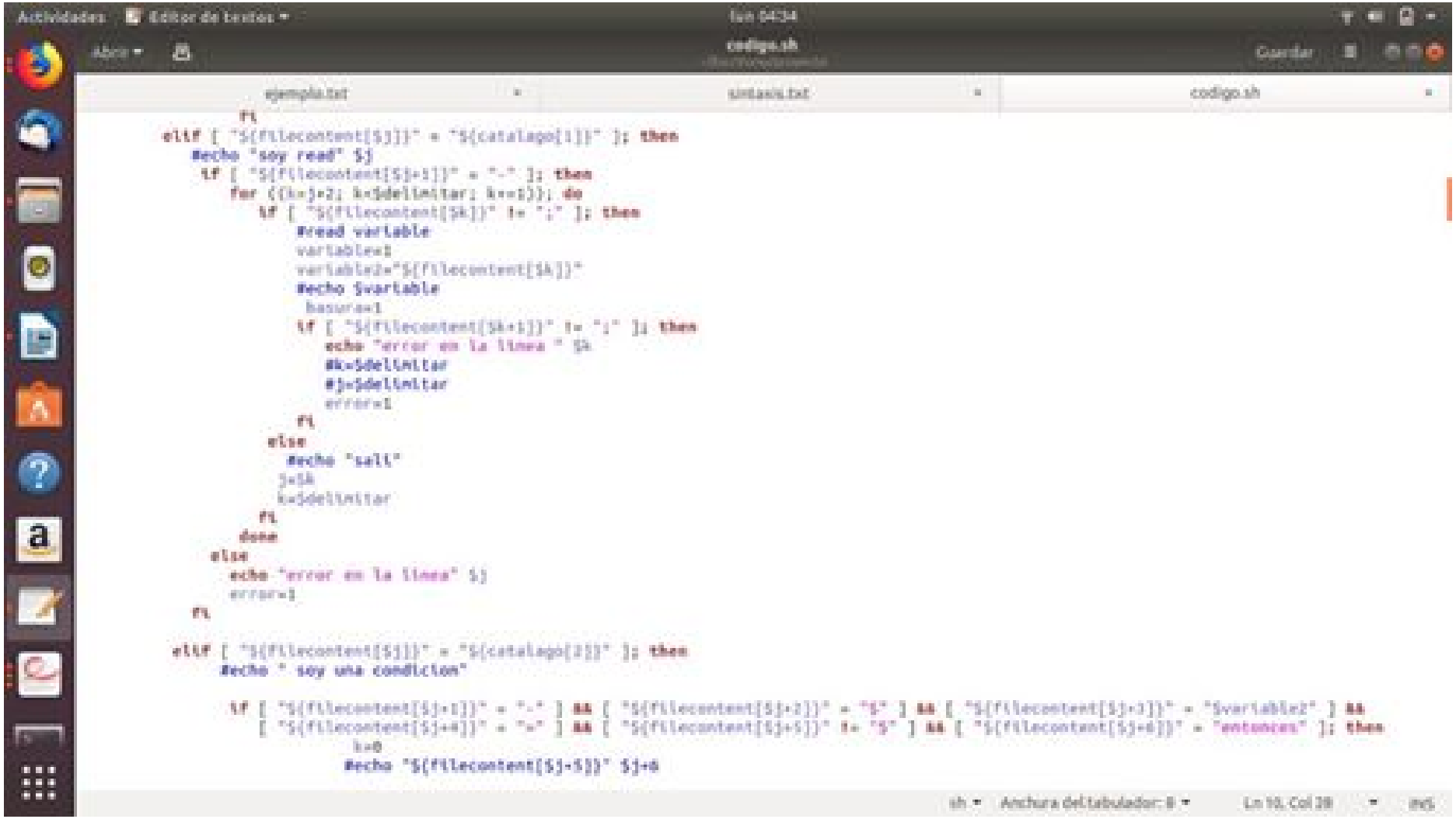
DESARROLLO


Actividades
Editor de textos
lun 04:23
codigo.sh
Abrir
Guardar
ejemplo.txt
sintaxis.txt
codigo.sh
#echo $delimitar #se imprime la cantidad
for ((j=0; j<$delimitar; j++)); do #se barre la parte del documento
#echo "j==" $j
if ["${filecontent[$j]}" = "${catalogo[0]}"]; then #si se cumaple con la primera parte del comando
#echo " soy escribir" $j
k=0
if ["${filecontent[$j+1]}" = "-"]; then # si se cumple entonces es correcto y sigue para darle argumentps al eco
#echo "es correcto"
for ((k=$j+2; k<$delimitar; k++)); do #se hace ujn ciclo en cual se le asigna el indice j+2 para darle argjumentos
#echo "ciclo"
if ["${filecontent[$k]}" != ";"]; then # la condicon es que siempre yt cuando lo que este en el vector en esta
parte
#echo "//" #sea diferente de ; se agregaran los parametros para que pueda imprimir
if ["${filecontent[$k]}" != "$"]; then
basura=1
else
if ["$variable2" = "${filecontent[k+1]}"]; then
#echo $variable
k=$k+1
j=$k
else
echo "error en linea " $k
error=1
fi
fi
else
#echo "se encontro ; " # si ya encontro ; quiere decir que el analisis debe terminar
j=$k #j toma el indice de k para continuar donde se quedo
k=$delimitar #se rompe el ciclo
#echo "ahora j vale " $j
fi
done
else
echo "error en la linea " $j
error=1
sh
Anchura del tabulador: 8
Ln 10, Col 28
INS

DESARROLLO

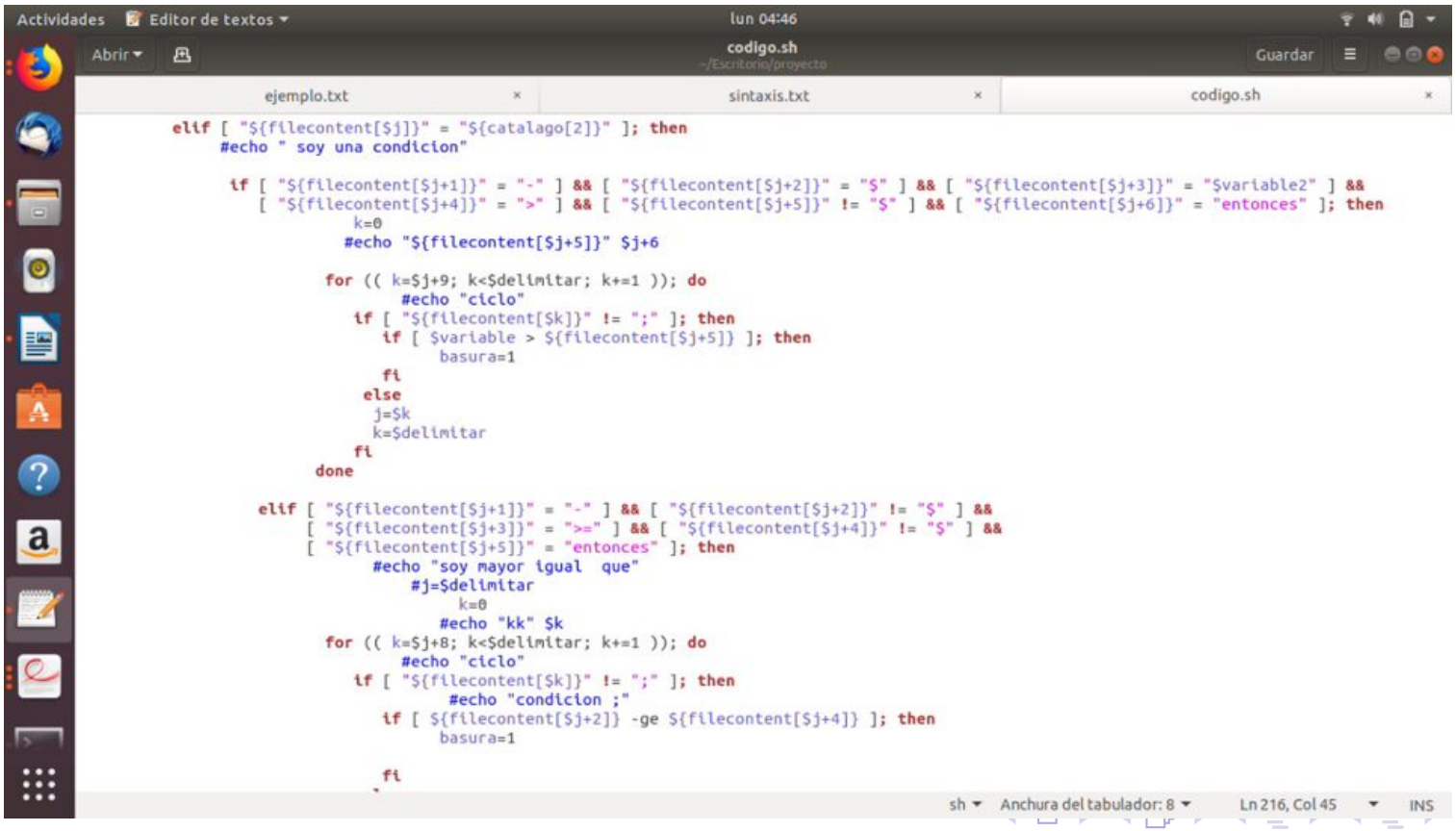
Tomando en cuenta otro ejemplo en esta parte se valida si la sintaxis si es correcta en este caso se esta evaluando el comando read el cual en nuestro catalago de comandos es leer si se encuentra el comando leer se traducirá al comando read en dado caso que este mal escrito se activara la bandera y el programa no se ejecutara.

DESARROLLO


Actividades
Editor de textos
lun 04:34
codigo.sh
Abrir
Guardar
ejemplo.txt
sintaxis.txt
codigo.sh
fi
elif ["${filecontent[$j]}" = "${catalago[1]}"]; then
#echo "soy read" $j
if ["${filecontent[$j+1]}" = "-"]; then
for ((k=j+2; k<$delimitar; k++)); do
if ["${filecontent[$k]}" != ";"]; then
#read variable
variable=$j
variable2="${filecontent[$k]}"
#echo $variable
basura=1
if ["${filecontent[$k+1]}" != ";"]; then
echo "error en la linea " $k
#k=$delimitar
#j=$delimitar
error=1
fi
else
#echo "salI"
j=$k
k=$delimitar
fi
done
else
echo "error en la linea" $j
error=1
fi
elif ["${filecontent[$j]}" = "${catalago[2]}"]; then
#echo " soy una condicion"
if ["${filecontent[$j+1]}" = "-"] && ["${filecontent[$j+2]}" = "$"] && ["${filecontent[$j+3]}" = "$variable2"] &&
["${filecontent[$j+4]}" = "="] && ["${filecontent[$j+5]}" != "$"] && ["${filecontent[$j+6]}" = "entonces"]; then
k=0
#echo "${filecontent[$j-5]}" $j+6
sh
Anchura del tabulador: 8
Ln 16, Col 28
INS

DESARROLLO

En esta parte se valida lo que son las condiciones esta paarte se validan si alguna parte de los operadores no se encuentra o si esta mal escrita o si le falta una parte por ejemplo un guion o un entonces los cuales son esenciales para que el comando se ejecute correctamente de lo contrario NO SE EJECUTARA EL PROGRAMA.



DESARROLLO

En esta parte se verifica si al final del análisis no hubo error o si hubo algún error en la condición if [$error = 0] se ejecutará el programa ingresado.

CONCLUSIÓN

El proyecto que realizamos ha contribuido de manera muy importante para conocer y adentrarnos más en lo que es el sistema operativo Linux. Antes de este proyecto nuestra opinin sobre Linux era que es un sistema operativo demasiado difícil de utilizar, pero a medida que fuimos investigando sobre el mismo, descubrimos que no es un sistema tan difícil de usar. En este proyecto sobre programación en Shell, fue como una guía breve para nosotros que vamos comenzando como usuarios de sistemas de Linux, este proyecto nos permitió comprender, ejecutar y empezar a programar en Shell, haciendo referencia especialmente a .Sh a base de prueba y error fuimos desarrollando este pequeño proyecto que aunque al principio no entendíamos bien como hacerlo al final lo comprendimos bien. Este proyecto nos sirve como experiencia para cuando estemos en un grado mas avanzado de nuestra vida profesional.