**Ejercicio 1:**

Tomando en cuenta el siguiente script responda las preguntas que se encuentran más abajo. **Importante:** como parte del resultado se deberá entregar el script en un archivo tipo sh y las respuestas en el mismo código.

#!/bin/bash

ErrorS() {

echo "Error. La sintaxis del script es la siguiente:"

echo "**El comando para saber la cantidad de líneas del archivo nombre\_archivo es**: $0 nombre\_archivo L" # COMPLETAR

echo " **El comando para saber el número de caracteres del archivo nombre\_archivo es**: $0 nombre\_archivo C" # COMPLETAR

echo “**El comando para saber la longitud de la línea más larga del archivo nombre\_archivo es**: $0 nombre\_archivo M" # COMPLETAR

}

ErrorP() {

echo "Error. nombre\_archivo **no existe o no tiene permiso de lectura**" # COMPLETAR

}

if test $# -lt 2; then

ErrorS fi

if ! test $1 -r; then ErrorP

elif test -f $1 && (test $2 = "L" || test $2 = "C" || test $2 = "M"); then if test $2 = "L"; then

res=`wc –l $1`

echo "**la cantidad de líneas del archivo es**: $res" # COMPLETAR

elif test $2 = "C"; then

res=`wc –m $1`

echo “**el número de caracteres del archivo es**: $res" # COMPLETAR

elif test $2 = "M"; then

res=`wc –L $1`

echo “**la longitud de la línea más larga del archivo es**: $res" # COMPLETAR

fi

else

ErrorS

Fi

**Responda:**

1. ¿Cuál es el objetivo de este script?

**El objetivo del script es mostrar por pantalla el número de líneas, el numero de caracteres o el numero de caracteres de la línea mas larga del archivo que se le mande (si se tiene permiso de lectura sino lanza error), dependiendo del segundo parámetro que se le mande.**

1. ¿Qué parámetros recibe?

**Recibe dos parámetros.**

**El primero es el archivo sobre el que se quiere consultar**

**El segundo parámetro indica que es lo que se quiere consultar del archivo**

**L= número de líneas**

**C=número de caracteres**

**M=numero de caracteres de la línea más larga**

1. Comentar el código según la funcionalidad (no describa los comandos, indique la lógica)

**Verifica la cantidad de parámetros y si no cumple con tener mínimo 2 parámetros tira ErrorS**

**Verifica la existencia del archivo indicado y los permisos de lectura y en caso de no tenerlos arroja ErrorP**

**Verifica la existencia del archivo indicado y si es un archivo regular y si el segundo parámetro es L,C,M.En el caso verdadero imprime el dato solicitado que corresponda según el segundo parámetro.En el caso falso lanza el ErrorS.**

1. Completar los “echo” con el mensaje correspondiente.
2. ¿Qué información brinda la variable “$#”? ¿Qué otras variables similares conocen? Explíquelas.

**$#: Número de parámetros pasados ​​al script o función.**

**$\* / $@: Parámetros de la función / script (argumentos).**

* + **"$\*" es lo mismo que "$1 $2 ..." 1**
  + **"$@" es lo mismo que "$1" "$2" ...**

**$! : El Pid del último proceso hijo ejecutado en segundo plano.**

**$$: Pid del proceso actual.**

**$?: Valor de ejecución del último comando.**

**$n: Parámetros enviados, donde n = 1, 2, 3, ..., 9.**

**${n} Parámetros enviados (igual que arriba), pero n puede ser> 9.**

**$0 En los scripts, ruta con la que se invocó el script.**

**$\_ Último parametro del último comando.**

**Las variables de entorno son lo mismo que las variables internas?**

1. Explique las diferencias entre los distintos tipos de comillas que se pueden utilizar en Shell scripts.

**Comillas Dobles ("): Se utilizan para definir textos y "se expanden". Es decir, las variables dentro de las comillas dobles son interpretadas (y no se muestran como el nombre de la variable).**

**Comillas Simples ('): Se utilizan para definir textos y "no se expanden". Es decir, las variables dentro de las comillas simples se muestran como el nombre de la variable (y no se muestran como su valor).**

**Acento Grave (`): Se utilizan para indicar a bash que interprete el comando que hay entre los acentos.**