

Files

Search

foca

hacker-lab

scanners.sh

test

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_not...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md

linux3_scripting.md

mod.sh

mkod.sh

linux2_commands.md

Open in Editor

Ejemplo 13: Manejo de un Banco con Bash

Este script de Bash proporciona una solución para gestionar una pequeña base de datos de clientes de un banco, permitiendo agregar, borrar, consultar, consignar y retirar dinero, así como generar un reporte.

El script utiliza un archivo llamado `banco.txt` para almacenar la información de los clientes y sus saldos. Los clientes pueden interactuar con el sistema mediante un menú que ofrece varias opciones para realizar diferentes operaciones.

```
#!/bin/bash

BANCO="banco.txt"

# Función para mostrar el menú
mostrar_menu() {
    echo "-----"
    echo "          Banco          "
    echo "-----"
    echo "1. Agregar cliente"
    echo "2. Borrar cliente"
    echo "3. Consultar saldo de cliente"
    echo "4. Listar clientes" # Nueva opción
    echo "5. Consignar"
    echo "6. Retirar"
    echo "7. Total de saldos"
    echo "8. Generar reporte"
    echo "0. Salir"
}
```

~/workspace: BANCO=~/banco.txt

```
if grep -q "$nombre:" "$BANCO"; then
    echo "El cliente ya existe."
else
    echo -n "Ingrese el saldo inicial: "
    read saldo
    echo "$nombre:$saldo" >> "$BANCO"
    echo "Cliente agregado con éxito."
fi
doneesac*) echo "Opción no válida, intenta de nuevo." ;; >> reporte.txtBANCO
```

```
-----
Banco
-----
1. Agregar cliente
2. Borrar cliente
3. Consultar saldo de cliente
4. Listar clientes
5. Consignar
6. Retirar
7. Total de saldos
8. Generar reporte
0. Salir
-----
Selecciona una opción: 1
Ingrese el nombre del cliente: huan
Ingrese el saldo inicial: 200
Cliente agregado con éxito.
-----
Banco
-----
1. Agregar cliente
2. Borrar cliente
3. Consultar saldo de cliente
4. Listar clientes
5. Consignar
6. Retirar
7. Total de saldos
8. Generar reporte
0. Salir
-----
Selecciona una opción: 
```

Files

Search

foca

hacker-lab

scanner.sh

test

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md


linux3_scripting.md

mcd.sh

mikcd.sh

linux2_commands.md

do ... done: Cuerpo del bucle que ejecuta el código entre do y done.



Taller

Taller de Agenda Telefónica

Ahora que has visto cómo funciona un sistema de gestión de banco con **bash**, te invitamos a aplicar estos conceptos para resolver un nuevo problema. Imagina una **agenda telefónica** que debe permitir gestionar contactos y teléfonos. Deberás crear un menú similar al del script del banco que permita:

1. **Agregar contacto:** Añadir un nuevo contacto a la agenda.
2. **Consultar contacto:** Buscar y mostrar el número de teléfono de un contacto específico.
3. **Borrar contacto:** Eliminar un contacto de la agenda.
4. **Listar contactos:** Mostrar todos los contactos almacenados.
5. **Ordenar por nombre:** Organizar los contactos alfabéticamente por nombre.
6. **Ordenar por teléfono:** Organizar los contactos por número de teléfono.
7. **Generar reporte:** Crear un archivo con una lista ordenada de contactos.

Este ejercicio te permitirá reforzar tus habilidades en scripting y te ayudará a ver cómo se aplican en situaciones prácticas. ¡Buena suerte y disfruta del desafío!

~/workspace\$ AGENDA="agenda.txt"

~/workspace\$ # Función para mostrar el menú

~/workspace\$ mostrar_menu() {

> echo "-----"

> echo "AGENDA TELEFÓNICA"

> echo "-----"

> echo "1. Agregar contacto"

> echo "2. Consultar contacto"

> echo "3. Borrar contacto"

> echo "4. Listar contactos"

> echo "5. Ordenar por nombre"

> echo "6. Ordenar por teléfono"

> echo "7. Generar reporte"

> echo "8. Salir"

> echo "-----"

> echo -n "Selecciona una opción: "

> }

~/workspace\$ # Función para agregar un contacto

~/workspace\$ agregar_contacto() {

> echo -n "Ingrese el nombre del contacto: "

> read nombre

> echo -n "Ingrese el número de teléfono: "

> read telefono

> # Crear el archivo si no existe

> if [! -f "\$AGENDA"]; then

> touch "\$AGENDA"

> echo "Archivo de agenda creado."

> fi

> # Verificar si ya existe el contacto

> if grep -q "\$nombre:" "\$AGENDA"; then

> echo "El contacto ya existe."

> else

> echo "\$nombre:\$telefono" >> "\$AGENDA"

> echo "Contacto agregado con éxito."

> fi

> }

Files

Search

foca

hacker-lab

scanner.sh

test

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md


linux3_scripting.md

mcd.sh

mikcd.sh

linux2_commands.md

do ... done: Cuerpo del bucle que ejecuta el código entre do y done.



Taller

Taller de Agenda Telefónica

Ahora que has visto cómo funciona un sistema de gestión de banco con **bash**, te invitamos a aplicar estos conceptos para resolver un nuevo problema. Imagina una **agenda telefónica** que debe permitir gestionar contactos y teléfonos. Deberás crear un menú similar al del script del banco que permita:

1. **Agregar contacto:** Añadir un nuevo contacto a la agenda.
2. **Consultar contacto:** Buscar y mostrar el número de teléfono de un contacto específico.
3. **Borrar contacto:** Eliminar un contacto de la agenda.
4. **Listar contactos:** Mostrar todos los contactos almacenados.
5. **Ordenar por nombre:** Organizar los contactos alfabéticamente por nombre.
6. **Ordenar por teléfono:** Organizar los contactos por número de teléfono.
7. **Generar reporte:** Crear un archivo con una lista ordenada de contactos.

Este ejercicio te permitirá reforzar tus habilidades en scripting y te ayudará a ver cómo se aplican en situaciones prácticas. ¡Buena suerte y disfruta del desafío!

~/workspace\$ # Función para consultar un contacto

~/workspace\$ consultar_contacto() {

> echo -n "Ingrese el nombre del contacto: "

> read nombre

> telefono=\$(grep "\$nombre:" "\$AGENDA" | cut -d ':' -f2)

> if [-z "\$telefono"]; then

> echo "Contacto no encontrado."

> else

> echo "El número de \$nombre es: \$telefono"

> fi

> }

~/workspace\$ # Función para borrar un contacto

~/workspace\$ borrar_contacto() {

> echo -n "Ingrese el nombre del contacto a borrar: "

> read nombre

> sed -i "/\$nombre:/d" "\$AGENDA"

> echo "Contacto borrado (si existía)."

> }

~/workspace\$ # Función para listar todos los contactos

~/workspace\$ listar_contactos() {

> echo "Lista de contactos:"

> echo "-----"

> echo "Nombre" "Teléfono"

> echo "-----"

> awk -F ':' '{printf "%-16s | %-12s\n", \$1, \$2}' "\$AGENDA"

> echo "-----"

> }

~/workspace\$ # Función para ordenar por nombre

~/workspace\$ ordenar_por_nombre() {

> sort -t ':' -k1,1 "\$AGENDA" -o "\$AGENDA"

> echo "Agenda ordenada por nombre."

> }

~/workspace\$ # Función para ordenar por teléfono

~/workspace\$ ordenar_por_telefono() {

> sort -t ':' -k2,2n "\$AGENDA" -o "\$AGENDA"

> }

Files

Search

foca

hacker-lab

scanner.sh

test

banco.txt

colores.txt

contactos.txt

estudiantes_por_id_de...

estudiantes_por_nom...

estudiantes_por_nota...

estudiantes.txt

frutas.txt

hello.txt

linux1_shell.md

linux2_commands.md


linux3_scripting.md

mcd.sh

mikcd.sh

linux2_commands.md

do ... done: Cuerpo del bucle que ejecuta el código entre do y done.



Taller

Taller de Agenda Telefónica

Ahora que has visto cómo funciona un sistema de gestión de banco con **bash**, te invitamos a aplicar estos conceptos para resolver un nuevo problema. Imagina una **agenda telefónica** que debe permitir gestionar contactos y teléfonos. Deberás crear un menú similar al del script del banco que permita:

1. **Agregar contacto:** Añadir un nuevo contacto a la agenda.
2. **Consultar contacto:** Buscar y mostrar el número de teléfono de un contacto específico.
3. **Borrar contacto:** Eliminar un contacto de la agenda.
4. **Listar contactos:** Mostrar todos los contactos almacenados.
5. **Ordenar por nombre:** Organizar los contactos alfabéticamente por nombre.
6. **Ordenar por teléfono:** Organizar los contactos por número de teléfono.
7. **Generar reporte:** Crear un archivo con una lista ordenada de contactos.

Este ejercicio te permitirá reforzar tus habilidades en scripting y te ayudará a ver cómo se aplican en situaciones prácticas. ¡Buena suerte y disfruta del desafío!

~/workspace\$ while true; do mostrar_menu; read opcion; case \$opcion in 1..

2. Consultar contacto

3. Borrar contacto

4. Listar contactos

5. Ordenar por nombre

6. Ordenar por teléfono

7. Generar reporte

8. Salir

Selecciona una opción: 6

Opción no válida. Intenta de nuevo.

AGENDA TELEFÓNICA

1. Agregar contacto

2. Consultar contacto

3. Borrar contacto

4. Listar contactos

5. Ordenar por nombre

6. Ordenar por teléfono

7. Generar reporte

8. Salir

Selecciona una opción: 1

Ingrese el nombre del contacto: huan

Ingrese el número de teléfono: 2

Archivo de agenda creado.

Contacto agregado con éxito.

AGENDA TELEFÓNICA

1. Agregar contacto

2. Consultar contacto

3. Borrar contacto

4. Listar contactos

5. Ordenar por nombre

6. Ordenar por teléfono

7. Generar reporte

8. Salir

Selecciona una opción:

