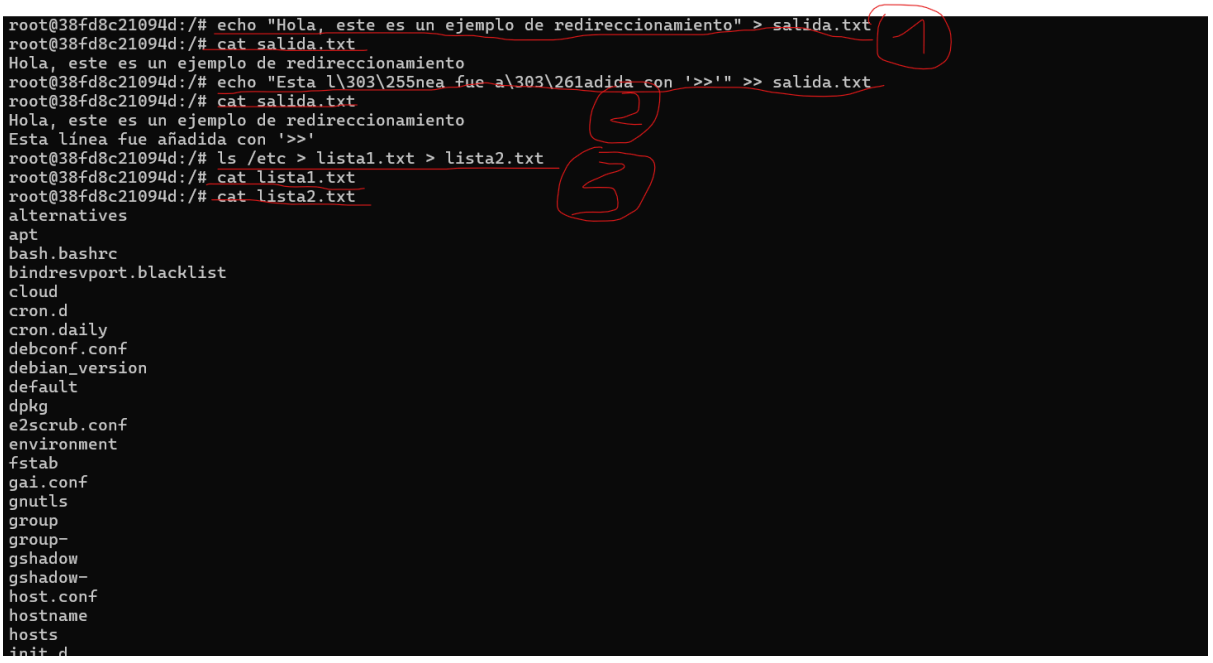
Sistemas Informáticos

Bloque 3
Práctica 5: Redireccionamiento de Entrada, Salida y Errores en Linux

Ejercicio 1: Redireccionamiento Básico de Salida Estándar (> y >>)

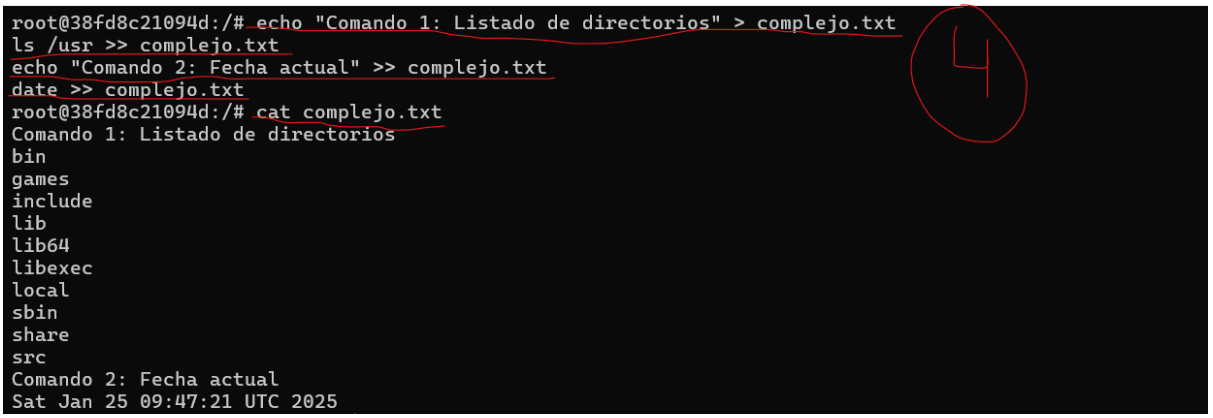
1. **Crea un archivo con contenido redirigido:**
   - Ejecuta el siguiente comando para guardar el texto en un archivo:
     `echo "Hola, este es un ejemplo de redireccionamiento" > salida.txt`
   - Verifica el contenido del archivo:
     `cat salida.txt`
2. **Añade contenido al archivo existente:**
   - Usa >> para agregar más texto al archivo:
     `echo "Esta línea fue añadida con '>>'" >> salida.txt`
   - Verifica nuevamente el contenido del archivo:
     `cat salida.txt`
3. **Sobrescribe y crea múltiples archivos:**
   - Ejecuta un comando que genere una lista de archivos en tu sistema y guárdala en dos archivos diferentes al mismo tiempo.
     `ls /etc > lista1.txt > lista2.txt`



4. Redirige la salida estándar de varios comandos consecutivos a un solo archivo:

- Encadena varios comandos y redirige todas las salidas al mismo archivo.

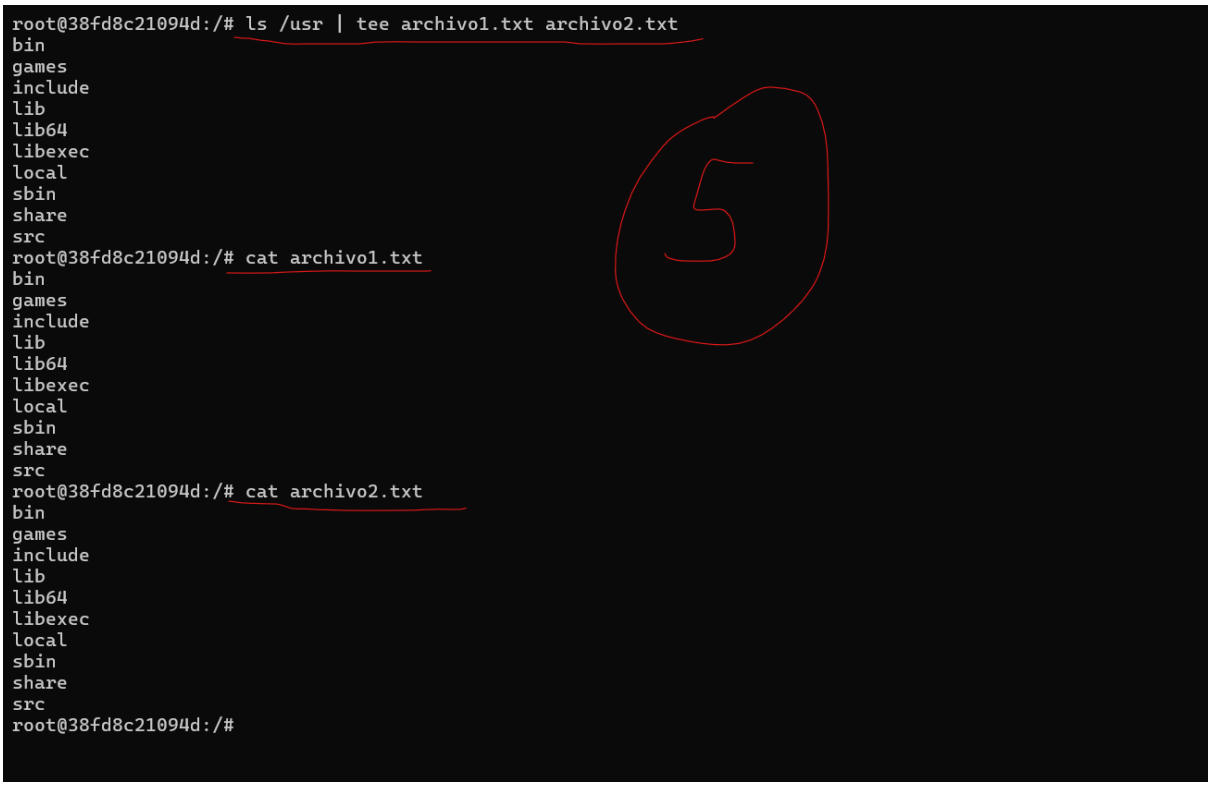
```
echo "Comando 1: Listado de directorios" > complejo.txt
ls /usr >> complejo.txt
echo "Comando 2: Fecha actual" >> complejo.txt
date >> complejo.txt
```



5. Divide la salida estándar en dos archivos diferentes simultáneamente:

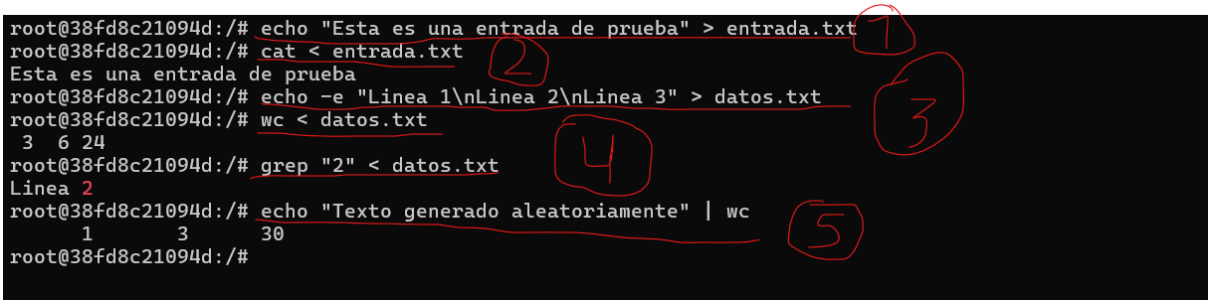
- Usa tee para enviar la salida a dos archivos:

```
ls /usr | tee archivo1.txt archivo2.txt
```



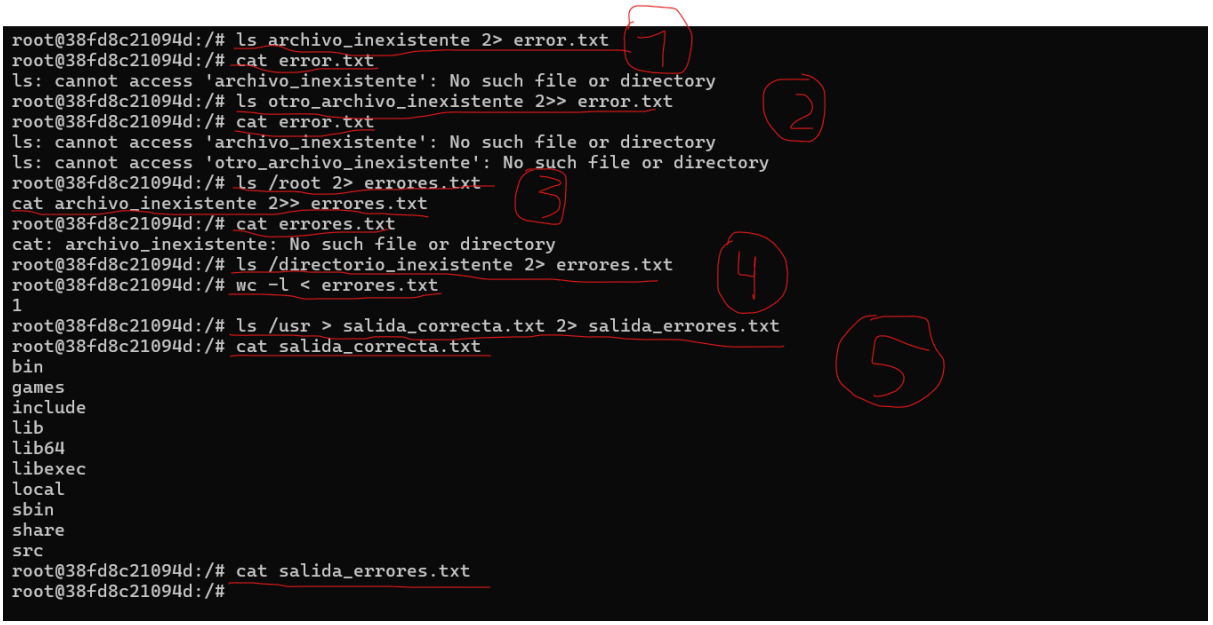
Ejercicio 2: Redireccionamiento de Entrada (<)

1. **Crea un archivo de texto para la práctica:**
   - Crea un archivo llamado entrada.txt:
     `echo "Esta es una entrada de prueba" > entrada.txt`
2. **Redirige el contenido como entrada para otro comando:**
   - Usa cat para mostrar el contenido de entrada.txt como entrada estándar:
     `cat < entrada.txt`
3. **Usa un archivo como entrada para un comando que procese datos:**
   - Crea un archivo de prueba y pásalo como entrada a un comando como wc (que cuenta líneas, palabras y caracteres):
     `echo -e "Linea 1\nLinea 2\nLinea 3" > datos.txt`
     `wc < datos.txt`
4. **Redirecciona la entrada estándar a otro comando:**
   - Usa el contenido de un archivo como entrada para otro comando que filtre los datos. Por ejemplo, busca una palabra específica en el archivo:
     `grep "2" < datos.txt`
5. **Simula un archivo como entrada con datos generados aleatorios:**
   - Usa echo o un comando para redirigir una entrada simulada a otro comando:
     `echo "Texto generado aleatoriamente" | wc`



Ejercicio 3: Redireccionamiento de Errores (2> y 2>>)

1. **Genera un error y redirígelo a un archivo:**
   - Ejecuta un comando que genere un error (por ejemplo, intentar listar un archivo inexistente):
     `ls archivo_inexistente 2> error.txt`
   - Verifica el contenido del archivo `error.txt`:
     `cat error.txt`
2. **Añade otro error al archivo existente:**
   - Genera otro error y usa `2>>` para agregarlo al archivo `error.txt`:
     `ls otro_archivo_inexistente 2>> error.txt`
   - Verifica nuevamente el contenido del archivo:
     `cat error.txt`
3. **Captura errores de múltiples comandos:**
   - Ejecuta una serie de comandos, algunos válidos y otros con errores, y redirige los errores a un archivo:
     `ls /root 2> errores.txt`
     `cat archivo_inexistente 2>> errores.txt`
4. **Genera un error y analiza cuántas líneas ocupa en el archivo de errores:**
   - Usa `wc` para contar las líneas del archivo generado:
     `ls /directorio_inexistente 2> errores.txt`
     `wc -l < errores.txt`
5. **Combina salida estándar y errores en diferentes archivos:**
   - Redirige la salida estándar a un archivo y los errores a otro archivo:
     `ls /usr > salida_correcta.txt 2> salida_errores.txt`



Ejercicio 4: Redireccionamiento Combinado (>&, 2>&1, &>>)

1. **Combina salida estándar y de errores:**
   - Redirige tanto la salida estándar como los errores al mismo archivo:
     `ls archivo_inexistente archivo_valido > combinado.txt 2>&1`
   - Verifica el contenido de `combinado.txt`:

```
root@38fd8c21094d:/# ls archivo_inexistente /etc > combinado.txt 2>&1
root@38fd8c21094d:/# cat combinado.txt
ls: cannot access 'archivo_inexistente': No such file or directory
/etc:
alternatives
apt
bash.bashrc
bindresvport.blacklist
cloud
cron.d
cron.daily
debconf.conf
debian_version
default
dpkg
e2scrub.conf
environment
fstab
gai.conf
gnutls
group
group-
```

`cat combinado.txt`

2. **Añade más salidas estándar y errores combinados al archivo:**
   - Usa `&>>` para agregar tanto la salida estándar como los errores:
     `ls otro_archivo archivo_inexistente &>> combinado.txt`
   - Verifica nuevamente el contenido del archivo:
     `cat combinado.txt`

```
root@38fd8c21094d:/# ls otro_archivo archivo_inexistente &>> combinado.txt
root@38fd8c21094d:/# cat combinado.txt
ls: cannot access 'otro_archivo': No such file or directory
ls: cannot access 'archivo_inexistente': No such file or directory
```

3. **Redirige salida estándar y errores combinados a diferentes archivos:**
   - Guarda la salida estándar en un archivo y redirige errores a otro archivo:
     `ls /etc /directorio_inexistente > salida_estandar.txt 2> salida_errores.txt`

```
root@38fd8c21094d:/# ls /etc /directorio_inexistente > salida_estandar.txt 2> salida_errores.txt
root@38fd8c21094d:/# cat salida_estandar.txt
/etc:
alternatives
apt
bash.bashrc
bindresvport.blacklist
cloud
cron.d
cron.daily
debconf.conf
debian_version
default
dpkg
```

4. **Redirige todo al mismo archivo pero con apéndice:**
   - Usa `&>>` para guardar tanto la salida estándar como los errores combinados en el mismo archivo:
     `ls /etc /directorio_inexistente &>> combinados.txt`

```
root@38fd8c21094d:/# ls /etc /directorio_inexistente &>> combinados.txt
root@38fd8c21094d:/# cat combinados.txt
ls: cannot access '/directorio_inexistente': No such file or directory
/etc:
alternatives
apt
bash.bashrc
bindresvport.blacklist
cloud
cron.d
cron.daily
debconf.conf
debian_version
default
dpkg
e2scrub.conf
environment
fstab
gai.conf
gnutls
```

5. **Filtra errores y salidas con comandos encadenados:**
   - Usa `grep` para buscar errores específicos y redirigirlos a un archivo, mientras capturas las salidas estándar:
     `ls /etc /directorio_inexistente > salidas.txt 2>&1 | grep "No such file" > errores_filtrados.txt`

```
root@38fd8c21094d:/# ls /etc /directorio_inexistente > salidas.txt 2>&1 | grep "No such file" > errores_filtrados.txt
root@38fd8c21094d:/# cat salidas.txt
ls: cannot access '/directorio_inexistente': No such file or directory
/etc:
alternatives
apt
bash.bashrc
bindresvport.blacklist
cloud
cron.d
cron.daily
debconf.conf
debian_version
default
dpkg
e2scrub.conf
```

```
root@38fd8c21094d:/# cat errores_filtrados.txt
root@38fd8c21094d:/#
```

```
root@38fd8c21094d:/# cat salida_errores.txt
ls: cannot access '/directorio_inexistente': No such file or directory
root@38fd8c21094d:/#
```

Ejercicio 5: Pipe (|) para Redirigir Salida entre Comandos

1. **Encadena comandos con un pipe:**
   - Redirige la salida de un comando como entrada para otro:
     `cat salida.txt | grep "ejemplo"`
   - Observa el resultado del comando.
2. **Combina múltiples pipes con comandos avanzados:**
   - Encadena comandos para filtrar y procesar datos:
     `ls /bin | grep "bash" | sort > resultado.txt`
3. **Procesa archivos con salida redirigida y filtrada:**
   - Usa `cat` y `wc` para procesar datos de un archivo redirigido y mostrar solo las líneas que cumplen con una condición:
     `cat salida.txt | grep "ejemplo" | wc -l`
4. **Usa pipes para filtrar errores y enviarlos a otro comando:**
   - Captura errores y pásalos a otro comando que los procese:
     `ls /directorio_inexistente 2>&1 | grep "No such file" > errores.txt`