



# LABORATORIO DE SISTEMAS OPERATIVOS I

TRABAJO PRÁCTICO N° 7

REDIRECCIONAMIENTO, FILTROS Y TUBERÍAS



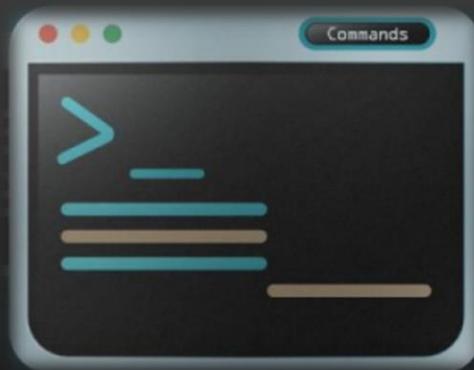


# ENTRADAS Y SALIDAS DE COMANDOS

- ¿Cómo operan los comandos?
  - Los comandos como cualquier otro programa utilizan datos de entradas que son procesados y transformados en información de salida.

**? ENTRADA** →

Parámetros del comandos  
Dispositivos  
Archivos



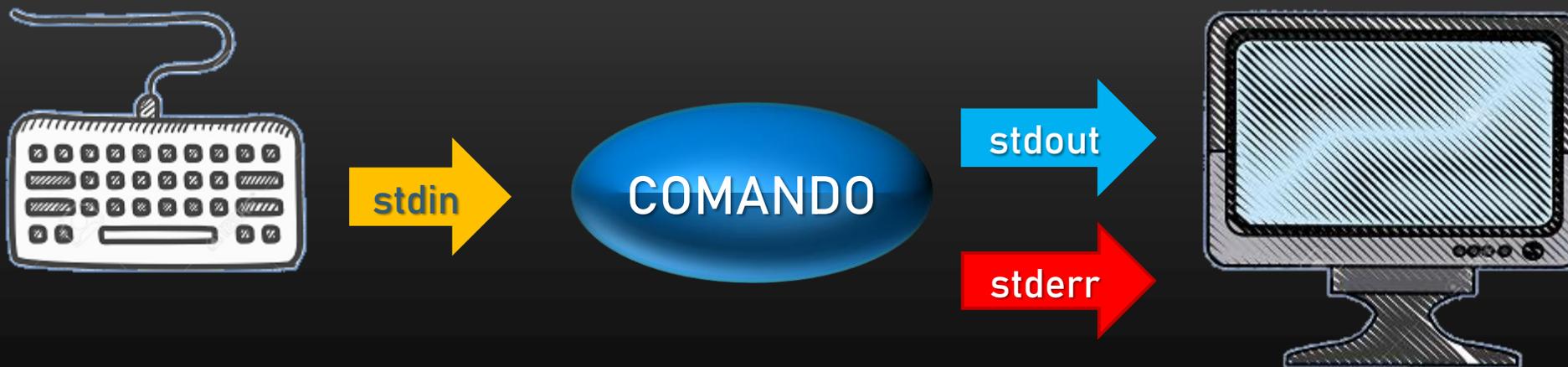
Dispositivos  
Archivos

Normal  
Error



# ARCHIVOS DE E/S ESTÁNDAR

- Cada vez que se ejecuta un comando el Kernel abre automáticamente los siguientes archivos
  - Entrada Estándar (stdin) **Descriptor 0**
  - Salida Estándar (stdout) **Descriptor 1**
  - Salida de Error Estándar (stderr) **Descriptor 2**





# REDIRECCIÓN DE ENTRADA (1)

- Se reemplaza la entrada estándar (teclado) por otro dispositivo (por ejemplo, un archivo)

**comando** < **archivo\_entrada**





## REDIRECCIÓN DE ENTRADA (2)

- **Ejemplo:** El comando `tee` permite que la entrada de datos realizada por teclado se muestre en pantalla y se almacene en los archivos especificados.

```
tee arch_salida_1 arch_salida_2 ... arch_salida_n
```

- Utilizando redireccionamiento de entrada y el comando `tee`, genere los archivos `prueba1.txt`, `prueba2.txt` y `prueba3.txt` con el contenido del archivo `passwd` del directorio `etc`.

```
tee prueba1.txt prueba2.txt prueba3.txt < /etc/passwd
```



# REDIRECCIÓN DE SALIDA (1)

- Se reemplaza la salida estándar (pantalla) por otro dispositivo (por ejemplo, un archivo)

**comando** > **archivo\_salida**

**comando** >> **archivo\_salida**





## REDIRECCIÓN DE SALIDA (2)

- **Ejemplo:** El comando `head` permite mostrar las `n` primeras líneas de uno o varios archivos de texto simple.

```
head -n cant_lineas archivo
```

- Utilizando redireccionamiento de salida y el comando `head`, genere en `~` el archivo `usuarios.txt` con los 5 primeros usuarios del archivo `passwd` del directorio `etc`.

```
head -n 5 /etc/passwd > ~/usuarios.txt
```

 Crea usuarios.txt

```
head -n 5 /etc/passwd >> ~/usuarios.txt
```

 Agrega datos a usuarios.txt



# REDIRECCIÓN DE SALIDA DE ERROR (1)

- Se reemplaza la salida de error estándar (pantalla) por otro dispositivo (por ejemplo, un archivo)

```
comando 2> archivo_salida_error
```

```
comando 2>> archivo_salida_error
```





## REDIRECCIÓN DE SALIDA DE ERROR (2)

- **Ejemplo:** El comando `find` permite buscar archivos en el sistemas de archivos devolviendo las ubicaciones (rutas) correspondientes.

```
find directorio_busq [-opciones] patrón_busqueda
```

- Utilizando redireccionamiento de salida de error y el comando `find`, guarde los mensajes de error del comando en el archivos `~/errores.dat`. Considere que se buscará en todo el sistema el archivo `fstab`.

```
find / -name fstab 2> ~/errores.dat Crea errores.dat
```

```
find / -name fstab 2>> ~/errores.dat Agrega datos a errores.dat
```

¿Cómo eliminar la salida de error?



# REDIRECCIÓN DE ENTRADAS Y SALIDAS

- Es posible combinar las distintas formas de redireccionamiento al ejecutar un comando.

```
comando < archivo_entrada > archivo_salida
```

```
comando < archivo_entrada 2> archivo_salida_error
```

```
comando > archivo_salida 2> archivo_salida_error
```

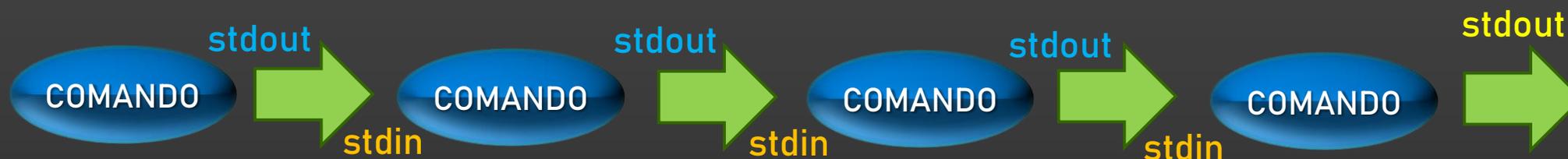
```
comando > archivo_salida 2>&1
```

```
comando 2> archivo_salida_error >&2
```



# TUBERÍAS (1)

- Es posible encadenar comandos para realizar tareas complejas que un único comando no puede resolver.



- Formato

```
Comando_1 | comando_2 | comando_3 | ... | comando_n
```



## TUBERÍAS (2)

- Los comandos que pueden conectarse mediante tuberías reciben en nombre de filtros y se caracterizan por trabajar con la entrada (stdin) y salida (stdout) estándar.

- `cat`
- `more`
- `grep`, `egrep`, `fgrep`
- `wc`
- `head`
- `tail`
- `cut`
- `paste`
- `sort`
- `uniq`
- `tee`



# COMANDO GREP

- Permite buscar una cadena (patrón regular) en uno o más archivos y muestra las líneas que la contienen.

**grep** [-opciones] "cadena" archivo

## Algunas opciones

- i ignora diferencias entre mayúsculas y minúsculas
- n muestra los números de líneas donde aparece la cadena buscada
- w busca la cadena como una palabra
- c cuenta las líneas que contienen la cadena buscada
- v muestra las líneas que no coincidan con la cadena buscada
- e permite indicar 2 o más cadenas a buscar
- ^ indica que sólo se muestren las líneas que empiecen con la cadena
- \$ indica que sólo se muestren las líneas que terminen con la cadena



# COMANDO CUT

- Permite mostrar secciones de uno o más archivos.

```
cut [-opciones] archivo
```

## Algunas opciones

- f especifica las columnas que serán "extraídas"
- d especifica el carácter separador de columnas a considerar



# COMANDO SORT

- Ordenar el contenido de un archivo de acuerdo a un campo específico. Por defecto ordena por el primer campo.

```
sort [-opciones] archivo
```

## Algunas opciones

- k especifica la columna por la se realizará la ordenación
- t especifica el carácter separador de columnas a considerar
- n ordena considerando un campo numérico (por defecto se toma ASCII)
- u elimina las líneas repetidas
- r orden inverso



## EJEMPLOS (1)

- ¿Cuántos archivos ordinarios contiene el directorio personal del usuario?

```
alumno@debian:~$ ls -lRa ~ | grep ^"- " | wc -l
```

- Considerando las líneas del archivo /etc/group que contengan la palabra "alumno", ordénelas de forma creciente.

```
alumno@debian:~$ grep "alumno" /etc/group | sort
```

- ¿En cuántos directorios se encuentra el archivo hostname?

```
alumno@debian:~$ find / -name "hostname" 2> /dev/null | wc -l
```



## EJEMPLOS (2)

- Muestre la cantidad de caracteres de las primeras 5 líneas del archivo `passwd` del directorio `etc`. ¿Cómo podría guardar el resultado obtenido en el archivo `~/salida.txt`?

```
alumno@debian:~$ head -5 /etc/passwd | wc -c > ~/salida.txt
```

- Muestre los últimos 10 usuarios (nombre e id) del archivo `passwd` del directorio `/etc`.

```
alumno@debian:~$ tail -10 /etc/passwd | cut -f1,f3 -d ":"
```

- Modifique el comando anterior de modo que los usuarios se ordenen por id.

```
$ tail -10 /etc/passwd | cut -f1,f3 -d ":" | sort -k2 -t ":" -n
```

