

Entrada, salida y error estándar.

El sistema Operativo asigna tres descriptores de archivos:

```
0 Entrada estandar (stdin) -> teclado
1 Salida estandar (stdout)-> pantalla
2 Error estandar (stderr) -> pantalla
```

Redireccionamiento de salida y error estándar.

Salida Estandar

```
ls -la > docs
ls -la 1>docs1
```

Error Estandar

```
ls -la papa 2> error
```

Bash

Ejecución de comandos consecutivamente

','

Se ejecuta un programa despues de otro consecutivamente sin importar el resultado.

Ejemplos

1. `Date ; cal`
2. `Mkdir dir ; cd dir ; ls`
3. `sleep 5 ; echo hola`

Valores de retorno (\$?)

1. Cuando el comando se ejecuta correctamente retorna un valor 0.
ls
2. Cuando el comando no se ejecuta correctamente retorna un valor diferente de 0
ls okio

'&&'

Ejecuta un comando sólo si el anterior se ejecuta '&&'

Ejemplos

1. > ls /carpeta/archivo && cp /carpeta/archivo .
2. > ls /carpeta/archivo && cp /carpeta/archivo .

'||'

Utilizamos '||' cuando un comando se ejecuta debido a que el anterior no.

```
cp dir/ . || echo "error"
```

Procesos en segundo plano

Ejecutar tareas en segundo plano para poder continuar trabajando

'&'

```
sleep 100 &
```

Background 'bg'

Muestra cuantos procesos fueron mandados a segundo plano

Foreground 'fg'

Trae tareas de segundo plano a consola

pipe '|'

1.

```
cat nombre | nl
```
2.

```
cat nombre | sort
```
3.

```
cat nombre | shuf
```
4.

```
cat nombre | sort | uniq
```
5.

```
cat nombre | sort | uniq | shuff | head -n 1
```

Scripts

Es un programa simple que se ejecuta en la consola

Ejemplo

1.

```
#!/bin/bash  
echo hola mundo
```

2. `#!/bin/bash`
`ls -la`

3. `#!/bin/bash`
`for i in {1..23};`
`do`
`echo $i`
`done`