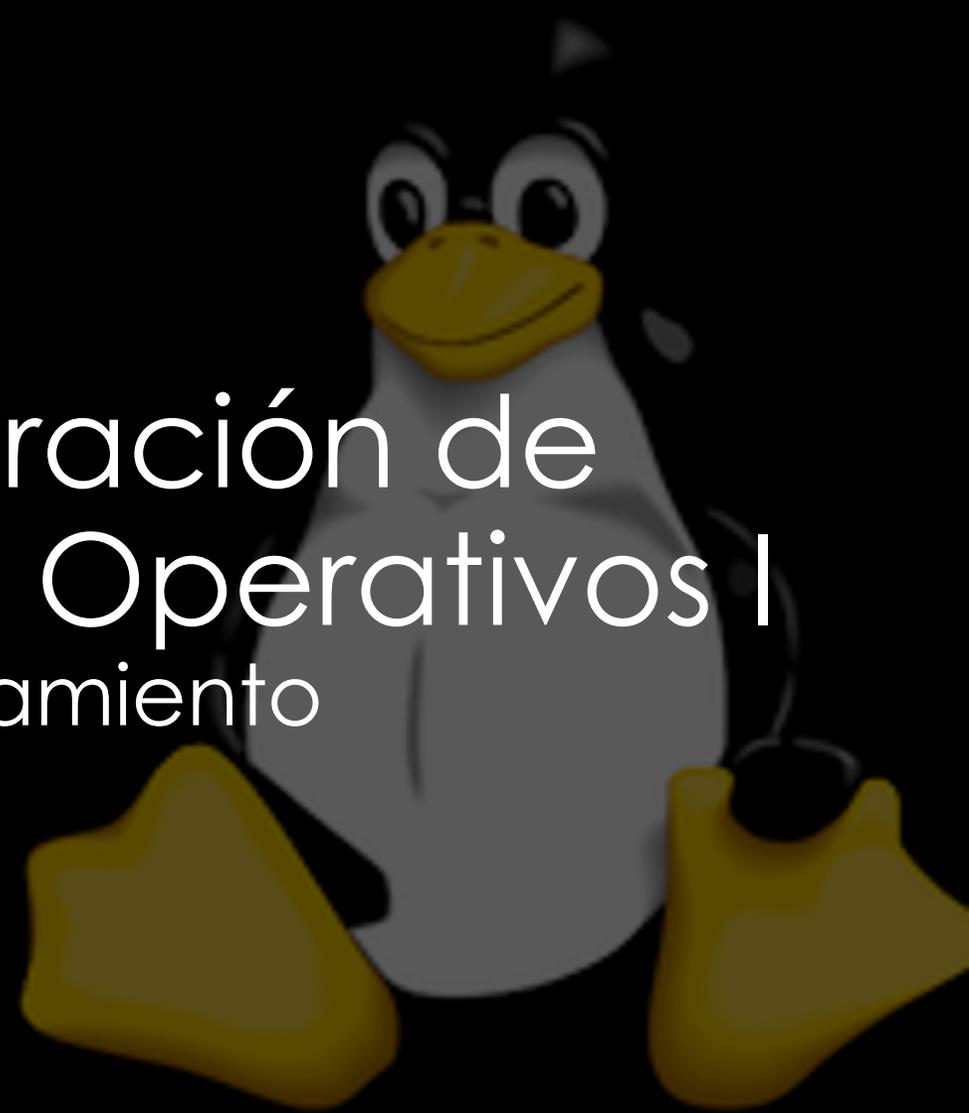


The Windows logo is partially visible on the left side of the slide, consisting of four colored panes (red, green, blue, yellow) arranged in a cross pattern. A vertical white line is positioned to the right of the logo.

Alejandra Pozo Jara

Administración de Sistemas Operativos I

Redireccionamiento



Pipes (Tuberías)

- Los pipes pueden ser utilizados para tomar la salida de un comando y que se pueda usar como entrada para un segundo comando

Pipes (Tuberías)

- Head y tail se pueden usar también no solo con archivos sino por ejemplo cuando se desea listar un lista larga de directorios, por ejemplo si se quiere la primeras 10 líneas quedaría:

```
ls /etc | head
```

- La salida será los primeros 10 directorios, es decir lo que sucede es que la salida de ls /etc se envía por el pipe como datos de entrada al comando head y la salida de head se muestra en pantalla.

Pipes (Tuberías)

- Los pipes se pueden usar uno a continuación de otro teniendo en cuenta que la salida de uno se utiliza como entrada del siguiente. Por ejemplo si colocamos:

```
ls -l /etc/ppp | nl
```

En este ejemplo **nl** se usa para numerar el listado de directorios.

- Si usamos:

```
ls -l /etc/ppp | head | nl
```

En este caso se numerarán los primeros 10 directorios.

- Si usamos:

```
ls -l /etc/ppp | nl | tail -5
```

Primero se numeran y luego se muestran los 5 últimos directorios numerados



Redireccionamiento I/O

Existen 3 tipos de flujos estándar:

- **STDIN (<)**

Se considera como estándar de entrada, que por lo general es proporcionada por el usuario desde el teclado.
- **STDOUT (>)**

Es el estándar de salida que por lo general (cuando no hay errores) aparecen en la pantalla
- **STDERR**

Son mensajes de error que se producen cuando se ejecutan los comandos

Redireccionamiento I/O

- `< /path/to/file` (Redirect STDIN from file)
- `> /path/to/file` (Redirect STDOUT overwriting file)
- `>> /path/to/file` (Redirect STDOUT appending file)
- `2> /path/to/file` (Redirect STDERR overwriting file)
- `2>> /path/to/file` (Redirect STDERR appending file)
- `&> /path/to/file` (Redirect STDERR and STDOUT overwriting file)
- `&>> /path/to/file` (Redirect STDERR and STDOUT appending file)



Redireccionamiento STDIN

- Se conoce como también como estándar 0, los datos son proporcionados por el teclado.
- Se puede usar el comando tr:

```
tr 'a-z' 'A-Z'
```

Lo que hace es convertir todo lo que se escriba en minúsculas a mayúsculas y cuando se desea finalizar se ejecuta Ctrl+D.

- Para utilizarlo con STDIN

```
tr 'a-z' 'A-Z' < /etc/yum.conf
```

se está diciendo que el contenido de ese directorio se convierta a mayúsculas.

Redireccionamiento STDOUT

- Es la salida de un comando cuando se ha ejecutado correctamente .
- El comando `echo` se puede usar para imprimir los mensajes a STDOUT.
- Si se coloca **echo Linux 1** se visualizará el texto Linux 1 en el STDOUT, pero si colocamos lo siguiente:

```
echo Linux 1 > ejemplo.txt
```

La salida STDOUT se redirecciona al archivo `ejemplo.txt`, cabe mencionar que si el archivo no existe este se creará. Si colocamos `cat ejemplo.txt` colocaremos la salida en el STDOUT.



Redireccionamiento STDOUT

- Con estos elementos, si creamos un archivo ejemplo.txt y colocamos:

```
echo "línea 1" >ejemplo.txt  
cat ejemplo.txt
```

- Se mostrará como contenido la línea que mandamos a imprimir con echo. Si volvemos a realizar nuevamente los mismo con un nuevo texto:

```
echo "segunda línea" >ejemplo.txt  
cat ejemplo.txt
```

- Se mostrará no dos líneas, sino una sola ya que se sobrescribe, para evitar sobrescribir se debe colocar >>, que significa (append).

Redireccionamiento STDERR

- Es la salida de un comando cuando se ha producido un error, que normalmente se envía a la consola donde se ejecuta.
- Por ejemplo si colocamos:

```
ls /fake
```

La salida será un error porque el directorio no existe

- Se puede redireccionar el mensaje de error: para ello se puede colocar:

```
ls /fake 2> msgerror.txt
```

```
ls /fake 2> /dev/null    (el mensaje se descarta.)
```

Redireccionamiento múltiple

- STDERR y STDOUT se pueden redirigir al mismo tiempo:

```
ls /fake /etc/ppp >ejemplo.txt
```

En este caso la salida de error se visualizará en pantalla y el contenido se irá al archivo.

```
ls /fake /etc/ppp 2>ejemplo.txt
```

Aquí se guardará el mensaje de error y se visualizará el listado de directorios.

- Si queremos que los dos resultados se vayan a un solo archivo se utilizará `&>`, o a su vez en una sola línea se puede colocar

```
ls /fake /etc/ppp >ejemplo.txt 2>error.txt
```

se puede usar `&>>` para no sobrescribir.

Comando find

- Permite buscar archivos:

```
find [starting directory] [search option] [ search  
criteria] [result option]
```

- Por ejemplo:

```
find /etc -name "ssh*"
```

- En este caso se está mandando a buscar bajo etc todas las coincidencias con archivos que inicien con ssh, si no se especifica el directorio donde se desea buscar se asume el directorio actual
- Si al final de la línea escribimos la opción `-ls` nos mostrará en detalle los resultados similar cuando colocamos `ls -l`:

```
find /etc -name "ssh*" -ls
```

Comando find

- Si ejecutamos:

```
find /etc -size +5M
```

Significa que se liste archivos de más de 5 Megabytes. Se puede usar K, M, G dependiendo las unidades, si ponemos en el ejemplo -5M sería archivos de -5 megabytes.

Option	Example
<code>-maxdepth</code>	<code>-maxdepth 1</code>
<code>-group</code>	<code>-group payroll</code>
<code>-iname</code>	<code>-iname hosts</code>
<code>-mmin</code>	<code>-mmin -10</code>
<code>-type</code>	<code>-type f</code>
<code>-user</code>	<code>-user bob</code>

Comando less

Es un comando pager, este comando permite visualizar el contenido en una sola página, así por ejemplo si ejecutamos:

```
ls /etc
```

vemos que el contenido es muy grande pero si utilizamos una tubería:

```
ls /etc | less,
```

lo que hacemos es enviar el contenido a less y se podrá desplazar en el resultado visualizado. Dentro de la ejecución del comando less podemos utilizar:

- / para buscar un elemento desde donde está el cursor hasta el final de archivo
- ? para buscar un elemento desde donde está el cursor hasta el inicio de archivo
- N nos envía hasta la coincidencia previa
- n nos envía hasta la próxima coincidencia

Comando egrep

El comando grep se utiliza para buscar una palabra dentro de un archivo. Se puede utilizar diferentes opciones:

```
egrep -ni bash /etc/passwd
```

En este caso buscamos bash en este archivo se resalta con color la coincidencia, se numera la salida y es indistinto de mayúsculas y minúsculas