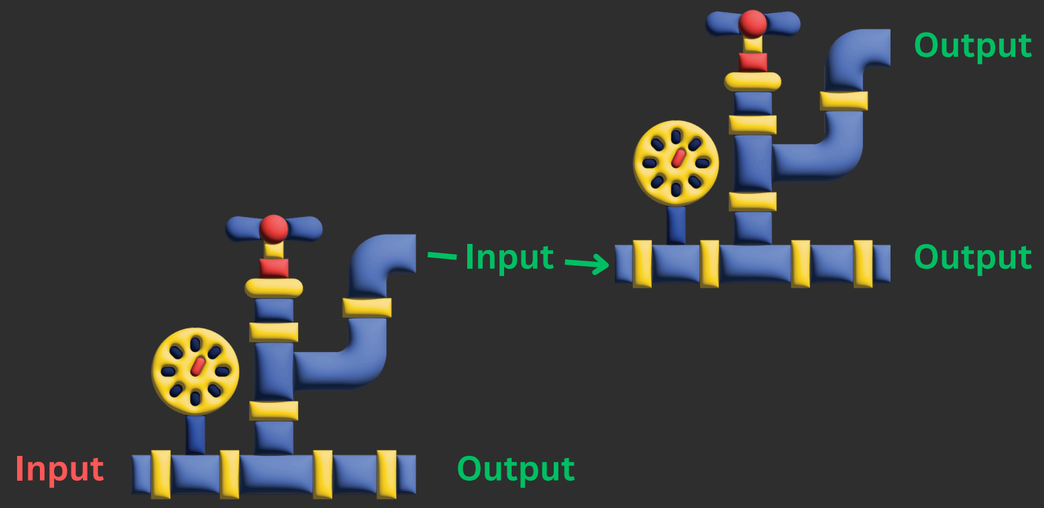
Pipeline utility in Linux



What is a Pipeline?

A Pipeline (|) is a powerful feature that allows you to combine multiple commands to perform complex tasks by passing the output of one command as the input to another. This way, you can create streamlined workflows, efficiently manipulating data without needing intermediate files or steps

Example 1

Imagine you have a Python script that generates output, and you need to automatically encrypt this output using Base64 encoding. By setting up a **pipeline**, you can automate this entire process, seamlessly handling the execution of the Python script and subsequent Base64 encryption in one streamlined workflow.

INPUT:

```
$ python3 printOut.py | base64
```

Python command — File to encrypt — Pipeline command — Encryption command

OUTPUT:

```
S2VlcCBnb2luZywgYmUgYmV0dGVyLCBi
ZSBzdHJvbmdlciwgYmUgdGhlIGJlYXN0C
k9yIGRpZSEhIQpPciBkaWUhISEKT3IgZGlI
SEhCk9yIGRpZSEhIQpPciBkaWUhISEK
```

Example 2

This **pipeline** finds a specific process and displays the process ID along with CPU and memory usage.

INPUT:

```
$ ps aux | grep "process_name" | awk '{print $2, $3, $4}'
```

OUTPUT:

```
4928. 0. 0. 0
```

Pipelines in Linux are a fundamental tool for anyone working in a command-line environment. They enhance productivity by simplifying complex tasks, conserving resources, reducing the need for temporary files, and allowing for real-time data processing. This makes them invaluable for system administrators, developers, data analysts, and any professional who needs to handle data efficiently.

Thank You
For watching