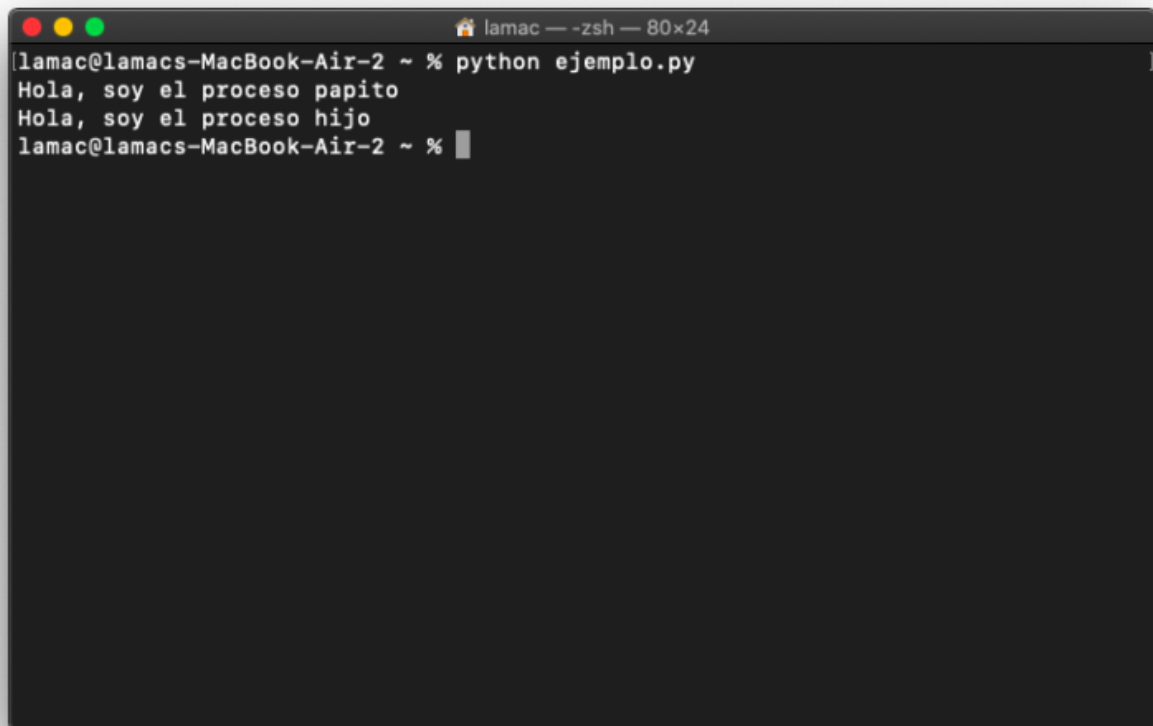


Crear un proceso hijo que comunica con el proceso padre mediante pipes en Python

[crayon-6a9d5804c3580247570466/]

En este ejemplo, se crea un pipe utilizando la función `os.pipe()`. Esto devuelve dos descriptores de archivo: uno para el extremo de escritura (`pipe_out`) y otro para el extremo de lectura (`pipe_in`). Luego, se crea un proceso hijo utilizando la función `os.fork()`. En el proceso hijo, se cierra el extremo de entrada del pipe (`pipe_in`), se escribe un mensaje en el extremo de salida (`pipe_out`) utilizando la función `os.write()`, y se cierra el extremo de salida (`pipe_out`). En el proceso padre, se cierra el extremo de salida del pipe (`pipe_out`), se lee el mensaje del extremo de entrada (`pipe_in`) utilizando la función `os.read()`, se imprime el mensaje, y se cierra el extremo de entrada (`pipe_in`).

Ten en cuenta que los procesos se ejecutan en paralelo y pueden no terminar en el orden esperado. También es importante asegurarse de cerrar todos los descriptores de archivo.



A terminal window titled "lamac — zsh — 80x24" with standard macOS window controls (red, yellow, green buttons). The terminal shows the execution of a Python script named "ejemplo.py". The output of the script consists of two lines: "Hola, soy el proceso papito" and "Hola, soy el proceso hijo". The prompt "lamac@lamacs-MacBook-Air-2 ~ %" is visible at the end of the output, followed by a cursor.

```
lamac@lamacs-MacBook-Air-2 ~ % python ejemplo.py  
Hola, soy el proceso papito  
Hola, soy el proceso hijo  
lamac@lamacs-MacBook-Air-2 ~ %
```