

# Signals

## Práctica 1

1) Crear el archivo loop.py. Escribir en su interior:

```
import time
while True:
    time.sleep(1)
    print("tick")
```

2) Ejecutar el script.

> **python loop.py**

Se deberá observar en la consola los mensajes “tick” cada un segundo.

3) Ejecutar la lista de procesos activos para ver el estado del mismo y obtener el pid

> **ps -elf**

Aparecerá el proceso con una S (Interruptable) debido a la función sleep(). Obtener el pid

3) Presionar Ctrl+Z para enviar la signal SIGSTP al proceso

4) Ejecutar la lista de procesos activos para ver el estado del mismo

> **ps -elf**

Aparecerá el proceso con una T (stopped).

5) Reanudar el proceso enviando la signal de continuar

>**kill -SIGCONT pid**

6) Se deberá ver el tick nuevamente

**Nota 1:** Puede utilizarse el comando watch para ejecutar periódicamente el comando ps. Ejemplo:

**watch -n 1 "ps -elf | grep [proceso]"**

**Nota 2:** Usar el nombre del proceso “proceso” y no “./proceso”. Ya que al ser zombie el comando ps lo muestra sin el “./”

## Práctica 2

1) Escribir un programa que cree una instancia child con la syscall fork(). El proceso padre deberá registrar el handler para SIGCHLD, y dentro del mismo ejecutar la función wait(). El proceso hijo deberá ejecutar un sleep de 5 segundos y luego terminar. Asegurarse mediante el comando PS que el proceso hijo no quede en estado “zombie”.

# Pipes

## ***Práctica 1***

1) Tomar el ejercicio de la practica 2 de signals y comunicar el proceso hijo con el padre mediante pipes. El proceso hijo deberá enviar un mensaje luego de esperar 5 segundos, luego esperará 10 segundos y terminará. El proceso padre deberá imprimir el mensaje por pantalla.