

# Guía de Trabajos Prácticos – Sistemas Operativos

---

Esta guía contiene 10 ejercicios prácticos para trabajar con argumentos por línea de comandos utilizando `getopt`, creación de procesos mediante `fork`, y manejo de señales con `signal`, todo en Python. Los ejercicios están diseñados para afianzar los conceptos de programación a bajo nivel en Sistemas Operativos.

## Ejercicio 1 – Getopt: Calculadora

Crear una calculadora, donde se pase como argumentos luego de la opción -o el operador que se va a ejecutar (+, -, \*, /), luego de -n el primer número de la operación, y de -m el segundo número. El programa deberá aceptar los argumentos en cualquier orden y verificar que no estén repetidos, que los números sean enteros y las operaciones válidas.

### Ejemplo de uso:

```
$ python3 calculadora.py -n 10 -m 5 -o +
```

Resultado: 15

## Ejercicio 2 – Getopt: Copiar Archivos

Escribir un programa que reciba dos nombres de archivos por línea de comandos con los parámetros -i y -o procesados con getopt(). El programa debe verificar que el archivo pasado a -i exista, abrirlo en modo lectura, copiar su contenido a un archivo con el nombre pasado en -o, y sobrescribirlo si ya existe.

### Ejemplo de uso:

```
$ python3 copiar.py -i archivo_entrada.txt -o copia.txt
```

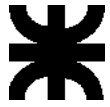
Archivo copiado correctamente.

## Ejercicio 3 – fork: Suma de Pares por Proceso

Escribir un programa en Python que reciba -n <numero> para generar esa cantidad de procesos hijos. Cada hijo calculará la suma de números pares entre 0 y su PID. El padre debe esperar a todos los hijos. Formato de salida: PID – PPID : <suma\_pares>

```
32803 – 4658: 269009202
```

```
32800 – 4658: 268943600
```



#### **Ejercicio 4 – fork + getopt: Modo Verboso**

Extender el ejercicio anterior agregando las opciones -v (verboso) y -h (ayuda). El modo verboso debe mostrar cuándo empieza y termina cada hijo, además de su resultado.

##### **Ejemplo de salida:**

Starting process 32800

Starting process 32803

Ending process 32803

32803 – 4658: 269009202

Ending process 32800

32800 – 4658: 268943600

#### **Ejercicio 5 – signal: Comunicación Ping-Pong**

Escribir un programa donde un hijo1 envíe SIGUSR1 al padre cada 5 segundos. El padre, al recibirlo, debe reenviar SIGUSR1 a hijo2. Hijo2 imprime 'pong' cada vez que recibe la señal. Finaliza luego de 10 señales.

##### **Ejemplo de salida:**

Soy el hijo1 con PID=1545: ping

Soy el hijo2 con PID=1547: pong

#### **Ejercicio 6 – signal: Múltiples Hijos con Señales**

El programa debe recibir -p <N> y crear N hijos. El padre enviará SIGUSR2 a cada uno. Cada hijo mostrará que recibió la señal, su PID, el número de señal y el PID del padre.

##### **Ejemplo de salida:**

Creando proceso: 1545

Creando proceso: 1547

Soy el PID 1545, recibí la señal 12 de mi padre PID 1540

Soy el PID 1547, recibí la señal 12 de mi padre PID 1540



### Ejercicio 7 – signal: Finalización por Señal

Un proceso hijo entra en bucle mostrando 'Esperando señal...' cada 3 segundos. El padre, tras 10 segundos, le envía SIGUSR1. El hijo debe manejar la señal, salir del bucle y terminar.

#### Ejemplo de salida del hijo:

Esperando señal...

Esperando señal...

Señal recibida, finalizando proceso.

### Ejercicio 8 – fork: Hijos con Tiempos Aleatorios

El programa recibe -n <cantidad> y genera esa cantidad de hijos. Cada hijo duerme entre 1 y 5 segundos y luego informa su finalización. El padre debe esperar a todos los hijos.

#### Ejemplo de salida:

Soy el hijo PID 3251 y terminé luego de 3 segundos

Soy el hijo PID 3253 y terminé luego de 5 segundos

### Ejercicio 9 – signal: Confirmar Salida con SIGINT

El programa muestra 'Trabajando...' indefinidamente. Al recibir Ctrl+C (SIGINT), pide confirmación para salir. Si se recibe un segundo SIGINT dentro de 5 segundos, termina. Si no, sigue ejecutando.

#### Ejemplo de salida:

Trabajando...

^C ¿Seguro que querés salir? Presioná Ctrl+C otra vez para confirmar

Trabajando...

^C Saliendo...

### Ejercicio 10 – getopt: Validación de Argumentos

Escribir un programa que acepte -u <usuario>, -p <puerto> y -h para ayuda. Debe validar que usuario no tenga espacios y que puerto esté entre 1024 y 65535.

#### Ejemplo de uso:

```
$ python3 servidor.py -u admin -p 8080
```

Usuario: admin | Puerto: 8080