

## Técnicas Digitales III

### Trabajo práctico: Señales

1. Escriba un programa que ejecute una espera activa con la función `sleep()`. Espere a que el programa finalice normalmente. Luego, observe el estado de salida (`exit status`) ejecutando

```
> echo $?
```

2. Luego, ejecute nuevamente el programa, pero finalice el mismo ejecutando en consola la combinación de teclas `CONTROL+C`. Vuelva a verificar el estado de salida. ¿Qué observa? ¿Varía este valor si el programa termina en forma normal o en forma abrupta?

3. Escriba un programa que usando la función `signal()` haga que el programa ignore la señal `SIGKILL`. Luego, imprima en pantalla el PID y ejecute un bucle infinito. Ejecute el programa. Lea el PID del proceso y desde otra consola ejecute

```
> kill -SIGKILL PID
```

¿Qué sucede? ¿Realmente se ignora la señal?

4. Escriba un programa que ejecute un bucle infinito. Luego, programe un manejador que al recibir la señal `SIGKILL` imprima por consola "No voy a terminar". Ejecute el programa y lea el PID del proceso y desde otra consola ejecute

```
> kill -SIGKILL PID
```

¿Qué sucede?

5. Escriba un programa que cree tres procesos hijos que ejecute cada uno un bucle infinito. Posteriormente, el proceso padre finaliza cada uno de los procesos enviando a cada uno de ellos la señal `SIGKILL` (para ello utilice la función `kill()`).

6. Escriba un programa que realice las siguientes tareas:

- Cree un proceso hijo que haga una espera activa de dos segundos y finalice.
- Luego de crear el proceso hijo, el padre debe ejecutar una espera activa por más de 10 segundos.
- Establezca en el proceso padre un manejador para la señal `SIGCHLD` que, cada vez que llegue esta, lea el estado de finalización del hijo.

7. Escriba un programa que ejecute un bucle infinito. Programe un manejador para la señal `SIGINT`, de manera que cada vez que se presiona la combinación de teclas `CTRL-C`, en lugar de terminar el proceso, se imprima el texto "me rehusó a terminar".