

# Medidas de performance

Clase de Práctico N°4  
Procesamiento Digital de Señales



INGENIERÍA  
BIOLÓGICA

# Contenidos

01

## Implementación FIR

Implementación del FIR causal en tiempo real

02

## Tutorial

Valgrind y memcheck

03

## Ejemplos

Uso de memoria

04

## Chequeo del FIR

Chequear el uso de memoria en el FIR implementado

**01**

**Implementación  
del FIR**

# Recordar

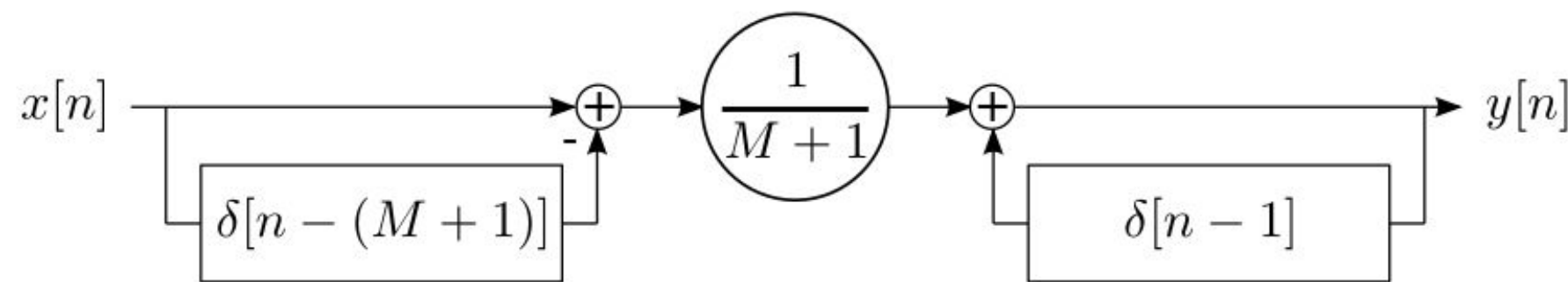
- ▶ El sistema de **Media Móvil Causal** se define por la ecuación

$$y[n] = \frac{1}{M+1} \sum_{k=0}^M x[n-k]$$

- ▶ **Implementación como sistema recursivo:** La ecuación genérica del sistema de media móvil causal es

$$y[n] = y[n-1] + \frac{x[n] - x[n-(M+1)]}{M+1}$$

- ▶ Se necesitan solo tres operaciones para calcular la salida.
- ▶ Se necesita almacenar solo dos valores,  $y[n-1]$  y  $x[n-(M+1)]$



M = 4

n	Array[0]	Array[1]	Array[2]	Array[3]
4	X_0	X_1	X_2	X_3
5	X_1	X_2	X_3	X_4
6	X_2	X_3	X_4	X_5



# 02

# Tutorial

Valgrind y  
memcheck

# Valgrind

- Es un **conjunto de herramientas** libres que ayuda en la depuración de problemas de memoria y rendimiento de programas.
- Debugger y Profiler (simulado).
- Utiliza el programa compilado:
  - `gcc -g programa.c -o programa`
- **Instalación**
  - `sudo apt-get install valgrind`

# Memcheck

- **Memcheck:**

- `valgind -v --tool=memcheck <programa>`
- Es un detector de errores en el uso de la memoria.
- Permite
  - Realizar un seguimiento del uso de memoria por parte del programa
  - Realizar control de acceso
  - Hallar fugas de memoria



03

# Ejemplos

Uso de memoria



# Ejemplo 1

- **Memcheck:**

- `gcc -g primero.c -o primero`

- `valgrind -v --tool=memcheck./primero`

# Ejemplo 1

- **Memcheck:**

- `Invalid write of size 4` marca el error (se intentó escribir un `int` fuera del lugar que asignó `malloc`).
- `at 0x8048434: main (primero.c:12)` marca que el error se cometió en la línea 12 de código.
- También puede verse que se hizo un `malloc` y un `free`, con lo que no hay *leaks*.

## Ejemplo 2 (inicialización)

```
❏ gcc -g segundo.c -o segundo
```

```
❏ valgrind -v --tool=memcheck --track-origins=yes ./segundo
```

## Ejemplo 3 (fuga de memoria)

```
❏ gcc -g tercero.c -o tercero
```

```
❏ valgrind -v --tool=memcheck --leak-check=full ./tercero
```

## Ejemplo 4 (fuga de memoria)

```
❏ gcc -g cuarto.c -o cuarto
```

```
❏ valgrind -v --tool=memcheck --leak-check=full ./cuarto
```

## Ejemplo 5 (limitaciones de memoria estática)

```
❏ gcc -g quinto.c -o quinto
```

```
❏ valgrind -v --tool=memcheck --leak-check=full ./quinto
```



04

# Chequeo del FIR

Chequeo de memoria sobre el FIR  
implementado

# Gracias

¿Preguntas?

Renato Sosa Machado



renato.sosast@gmail.com

Lucía Lemes



llemes@cup.edu.uy

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**