INFORME PRÁCTICA 2

Comando cyclictest

sudo cyclictest -p99 -N --smp -D 60

-p → Establece la **prioridad** para el primer thread. Cada hilo adicional obtiene una prioridad menor que equivale al max(valor anterior - 1, 0).

-N → Muestra el resultado en **nanosegundos**.

--smp → Establece opciones para testeos en **sistemas SMP** (multiprocesamiento).

-t, **establece** el número de **threads** para testear (default = todas las CPUs disponibles).

-a, **ejecuta hilos** en los procesadores especificados siguiendo round-robin fashion (default = todos).

Misma prioridad para todos los hilos.

-D → Especifica el **tiempo** que durará **ejecutando** el test.

Tablas

Cyclictest

No RT	*Latencia media (ns)*	*Latencia máxima (ns)*
S1	1737	359975
S2	5070	112616
S3	2275	4653213

RT	*Latencia media (ns)*	*Latencia máxima (ns)*
S1	1911	67689
S2	4046	23727
S3	2178	16737

CyclictestURJC

No RT	*Latencia media (ns)*	*Latencia máxima (ns)*
S1	2679	134404
S2	6688	48804
S3	2704	5323918

No RT	*Latencia media (ns)*	*Latencia máxima (ns)*
S1	2729	58308
S2	5191	64802
S3	2708	60405

Se observan varias cuestiones:

1. Al estresar el planificador aumenta significativamente la latencia media en todos los casos.

2. Los test ejecutados en kernel RT de linux, tardan más, esto es porque garantizan los plazos de los procesos que se ejecutan en los mismos, aumentando en ciertos casos la latencia para poder cumplir la temporalidad.

   Esto se ve reflejado en el S3, donde se mantienen unas latencias más estables que en los kernel no RT.

3. Los resultados son similares, y el funcionamiento es prácticamente idéntico. La varianza en las latencias se puede deber a que las medidas se tomaron en momentos distintos, donde el uso compartido del ordenador ha podido influir.

   No obstante, el comportamiento frente a los test de estrés es idéntico.