Julia López Augusto 23/10/2022

Práctica 2: Cálculo de latencias.

1. **Análisis con cyclictest**

Calcula las latencias máximas y medias en los 3 escenarios descritos arriba (S1,S2,S2) para las 3 configuraciones de hardware y kernel disponibles. Muestra los resultados obtenidos en la siguiente tabla (si lo prefieres puedes usar gráficos o plots de barras mientras la información quede correctamente mostrada).

cyclictestURJC						
	Laboratorios		RaspberryPi			
	Kernel NO RT		Kernel NO RT		Kernel RT	
	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)
S1						
S2						
S3						

CYCLICTEST:

Laboratorios

- Kernel NO RT
  - Latencia media (ns)
    - s1 -> 5095,83 nsecs
    - s2 -> 7134,33 nsecs
    - s3 -> 4531,83 nsecs
  - Latencia max (ns)
    - s1 -> 303776 nsecs
    - s2 -> 440624 nsecs
    - s3 -> 2645451 nsecs

RaspberryPi

- Kernel NO RT
  - Latencia media (ns)
    - s1 -> 22766,25 nsecs
    - s2 -> 24729,75 nsecs
    - s3 -> 20386,75 nsecs
  - Latencia max (ns)
    - s1 -> 155303 nsecs
    - s2 -> 563246 nsecs

        - s3 -> 157389 nsecs

- Kernel RT
    - Latencia media (ns)
        - s1 -> 22168,75 nsecs
        - s2 -> 14799,5 nsecs
        - s3 -> 18348,5 nsecs
    - Latencia max (ns)
        - s1 -> 63178 nsecs
        - s2 -> 78357 nsecs
        - s3 -> 82492 nsecs

Estos han sido los valores que he obtenido tras probar cyclictest en los distintos escenarios que se pedía.
En los un sistema RT podemos ver que el resultado deseado no es de un valor único. En el resto de casos también se cumple.

2. Desarrollo de cyclictestURJC

Utilizando la herramienta cyclictestURJC que has desarrollado, rellena la siguiente tabla del mismo modo que hiciste en el apartado 1

cyclictestURJC						
	Laboratorios		RaspberryPi			
	Kernel NO RT		Kernel NO RT		Kernel RT	
	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)
S1						
S2						
S3						

CYCLICTESTURJC:

Las esperas en este caso son del orden de 1 milisegundo

Laboratorios

- Kernel NO RT
  - Latencia media (ns)
    - s1 -> 2021423 nsecs
    - s2 -> 2016246 nsecs
    - s3 -> 2018320nsecs
  - Latencia max (ns)
    - s1 -> 998997587 nsecs
    - s2 -> 998996579 nsecs
    - s3 -> 998997627 nsecs

RaspberryPi

- Kernel NO RT
  - Latencia media (ns)
    - s1 -> -8044 ; intuyo que es porque ha pasado 1 segundo y 8044 ns
    - s2 -> -6613; intuyo que es porque ha pasado 1 segundo y 8044 ns
    - s3 -> -24030; intuyo que es porque ha pasado 1 segundo y 24030 ns
  - Latencia max (ns)
    - s1 -> -74767468 ; intuyo que es porque ha pasado 1 o más segundos y 74767468 ns
    - s2 -> -74759387 ; intuyo que es porque ha pasado 1 o más segundos y 74759387 ns
    - s3 -> -74761721 ; intuyo que es porque ha pasado 1 o más segundos y 74761721 ns

- Kernel RT
    - Latencia media (ns)
        - s1 -> -8077; intuyo que es porque ha pasado 1 o más segundos y 8077 ns
        - s2 -> -18005; intuyo que es porque ha pasado 1 o más segundos y 18005 nsecs
        - s3 -> - 23269; intuyo que es porque ha pasado 1 o más segundos y 23269 nsecs
    - Latencia max (ns)
        - s1 -> -74762542; intuyo que es porque ha pasado 1 o más segundos y 74761721 ns
        - s2 -> -74753931; intuyo que es porque ha pasado 1 o más segundos y 74753931 nsecs
        - s3 -> -747663404; intuyo que es porque ha pasado 1 o más segundos y 747663404 nsecs

Las latencias que se han podido obtener podemos ver que las máximas son muy grandes e intuyo que al ser un sistema generalista las llamadas que he usado en el programa no son expulsables de la CPU en cualquier momento.