

1. Ejecutar los siguientes casos y justificar su comportamiento:

a) ./practica1 (multicore)

Se cumplen las restricciones temporales ya que el sistema es multicore y puede hacer más de una tarea a la vez.

```
julioa@f-l3210-pc01:~/Escritorio/TERCERO/PRIMER_CUATRI/SISTEMAS_EMPOTRADOS/practica1/src$ ./practica1
[1664567191.529982936] Thread 2 - Iteración 1 : Coste=0.65 s
[1664567191.53090338] Thread 4 - Iteración 1 : Coste=0.67 s
[1664567191.53072897] Thread 3 - Iteración 1 : Coste=0.69 s
[1664567191.529822936] Thread 1 - Iteración 1 : Coste=0.71 s
[1664567191.553183114] Thread 4 - Iteración 2 : Coste=0.66 s
[1664567191.553718936] Thread 2 - Iteración 2 : Coste=0.66 s
[1664567191.553217035] Thread 3 - Iteración 2 : Coste=0.68 s
[1664567191.553104089] Thread 1 - Iteración 2 : Coste=0.68 s
[1664567192.853324534] Thread 1 - Iteración 3 : Coste=0.33 s
[1664567192.853342026] Thread 3 - Iteración 3 : Coste=0.32 s
[1664567192.853321575] Thread 4 - Iteración 3 : Coste=0.31 s
[1664567192.853766868] Thread 2 - Iteración 3 : Coste=0.29 s
[1664567194.120273438] Thread 3 - Iteración 4 : Coste=0.57 s
[1664567194.123672938] Thread 4 - Iteración 4 : Coste=0.57 s
[1664567194.93177713] Thread 1 - Iteración 4 : Coste=0.61 s
[1664567194.183786507] Thread 2 - Iteración 4 : Coste=0.60 s
[1664567194.993323059] Thread 1 - Iteración 5 : Coste=0.60 s
[1664567195.20401792] Thread 3 - Iteración 5 : Coste=0.64 s
[1664567195.23788224] Thread 4 - Iteración 5 : Coste=0.65 s
[1664567195.83912290] Thread 2 - Iteración 5 : Coste=0.61 s
```

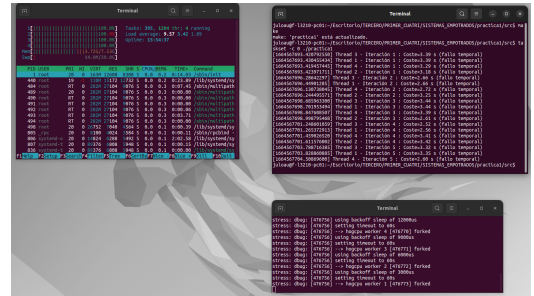
b) ./practica1 (monocore)

No se cumplen las restricciones temporales ya que el sistema es monocore y solo puede hacer una tarea a la vez.

```
julioa@f-l3210-pc01:~/Escritorio/TERCERO/PRIMER_CUATRI/SISTEMAS_EMPOTRADOS/practica1/src$ taskset -c 0 ./practica1
[1664567330.948734358] Thread 3 - Iteración 1 : Coste=3.79 s (fallo temporal)
[1664567338.942831588] Thread 4 - Iteración 1 : Coste=3.77 s (fallo temporal)
[1664567338.956732961] Thread 2 - Iteración 1 : Coste=2.22 s (fallo temporal)
[1664567338.964756697] Thread 1 - Iteración 1 : Coste=2.22 s (fallo temporal)
[1664567341.162661246] Thread 3 - Iteración 2 : Coste=2.21 s (fallo temporal)
[1664567341.170582553] Thread 4 - Iteración 2 : Coste=2.21 s (fallo temporal)
[1664567341.175436496] Thread 2 - Iteración 2 : Coste=2.21 s (fallo temporal)
[1664567341.183806033] Thread 1 - Iteración 2 : Coste=2.21 s (fallo temporal)
[1664567343.388867693] Thread 1 - Iteración 3 : Coste=2.18 s (fallo temporal)
[1664567343.388879982] Thread 3 - Iteración 3 : Coste=2.21 s (fallo temporal)
[1664567343.376395795] Thread 4 - Iteración 3 : Coste=2.21 s (fallo temporal)
[1664567343.382694209] Thread 2 - Iteración 3 : Coste=2.21 s (fallo temporal)
[1664567345.574468086] Thread 3 - Iteración 4 : Coste=2.18 s (fallo temporal)
[1664567345.582929095] Thread 4 - Iteración 4 : Coste=2.18 s (fallo temporal)
[1664567345.588747325] Thread 2 - Iteración 4 : Coste=2.18 s (fallo temporal)
[1664567345.571820803] Thread 1 - Iteración 4 : Coste=2.21 s (fallo temporal)
[1664567347.702898569] Thread 4 - Iteración 5 : Coste=3.71 s (fallo temporal)
[1664567347.77165782] Thread 2 - Iteración 5 : Coste=3.71 s (fallo temporal)
[1664567347.756502090] Thread 3 - Iteración 5 : Coste=3.70 s (fallo temporal)
[1664567347.777833051] Thread 1 - Iteración 5 : Coste=3.71 s (fallo temporal)
```

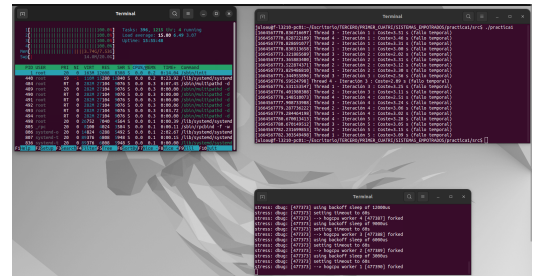
c) ./practica1 (monocore) + stress

No se cumplen las restricciones temporales ya que la única cpu que está ejecutando está saturada al 100% y por eso el coste de las ejecuciones de los threads es mayor.



d) ./practica1 (multicore) + stress

No se cumplen las restricciones temporales ya que todas las cpus están saturadas al 100% y por eso el coste de las ejecuciones de los threads es mayor.



2. ¿En qué casos de ejecución (nombrados anteriormente) el sistema es capaz de cumplir las restricciones temporales (tanto tiempo de cómputo como periodicidad)?

El único caso que ocurre es en el caso multicore sin stress.

3. ¿Qué número mínimo de cpus se necesitan para que tu programa ejecute correctamente sin fallos de restricciones temporales? Usa el comando taskset para comprobarlo.

Se necesitan 4 cpus.

4. ¿Qué solución se podría proponer para cumplir plazos estrictos temporales de periodicidad en la ejecución de los threads utilizando la configuración actual que tienen los ordenadores del laboratorio?

El uso de algún mecanismo de sincronización como mutex que no hacen espera activa; así no consume cpu y los plazos temporales se podrían cumplir.