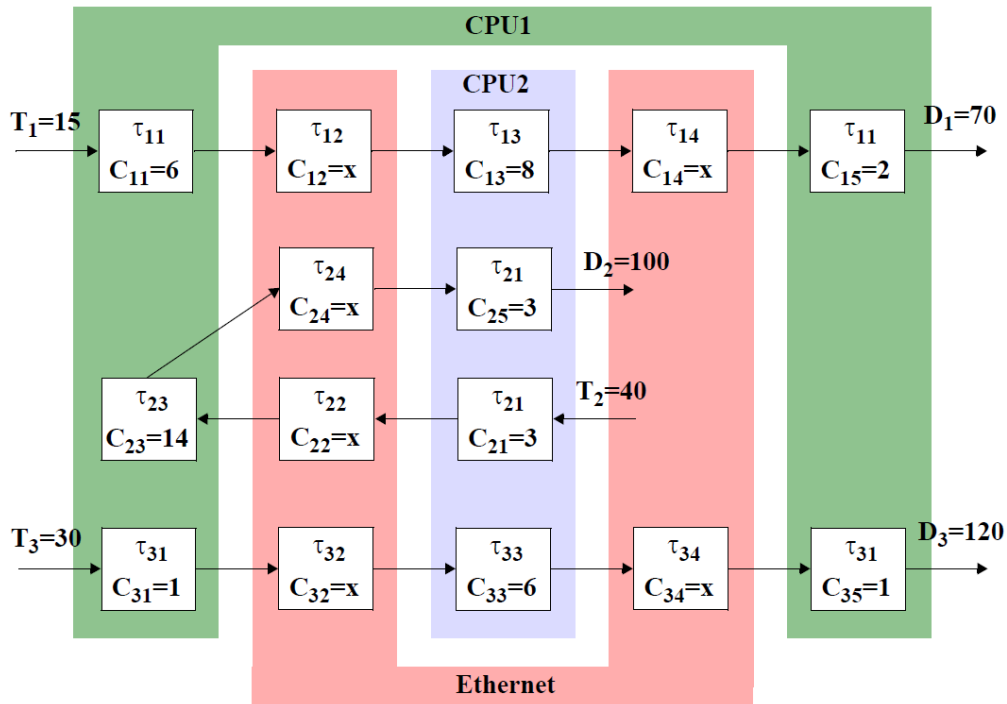


PRÁCTICA 5

Autor: Julio Escudero Cuesta

En la práctica 5 se proporciona el siguiente modelo de aplicación distribuida.



Entre la información a destacar sobre la aplicación se tiene que:

- Los enlaces por los que se envían los mensajes en la aplicación distribuida tienen una velocidad de **100 Mb/s**
- Los tiempos de ejecución señalados en el dibujo del modelo son en **décimas de segundo**

Actividad 1

Para la implementación de la aplicación que se suministra (FP), se pide:

- modelar el sistema con MAST (estimar aproximadamente los tiempos de transmisión de los mensajes en la red)
- asignar prioridades usando las técnicas de asignación y análisis que se consideren más adecuadas
 - anotar los resultados del análisis (tiempos de respuesta de las actividades con requisitos temporales)
- modificar el programa para ajustar las prioridades obtenidas
- medir los tiempos de respuesta en los PCs industriales del laboratorio (anotarlos).

Sacar conclusiones sobre el comportamiento observado en la ejecución de la aplicación, en particular en comparación con los valores obtenidos del análisis de planificabilidad.

Para la realización de esta práctica se suponen mensajes de 1518 bytes.

Como la red se va a modelar como un procesador más, hay que calcular los tiempos de ejecución de las tareas que se ejecutan.

La red tiene una velocidad de 100 Mb/s, que son 12.500.000 bytes/s. Para obtener cuánto tiempo tarda un mensaje de 1518 bytes en atravesar la red hay que dividir el tamaño del mensaje entre la velocidad de la red. $1518 / 12.500.000 = 0.00012144$ segundos.

Por tanto, un mensaje tarda **0.00012144 segundos** en enviarse por esa red.

Para el análisis con MAST se ha utilizado la técnica de análisis basada en offset con la variante de slanted y para la asignación de prioridades se ha usado el algoritmo HOSPA. En el sistema modelado con MAST se han puesto prioridades asignadas a la red, para que no entren dentro del análisis. El tiempo de respuesta asignado a la red es de 0.0012, para redondear y porque el resto de los tiempos estaban en décimas de segundos.

Los tiempos de respuesta obtenidos son:

- Tiempo de respuesta flujo 1: 29.013
- Tiempo de respuesta flujo 2: 54.013
- Tiempo de respuesta flujo 3: 42.013

Las prioridades obtenidas son:

- Operación t11: 6554
- Operación t13: 8192
- Operación t21: 16384
- Operación t23: 1
- Operación t31: 1307
- Operación t33: 1

El resto de las operaciones son de red y tenían prioridades preasignadas.

Viendo los resultados del análisis de planificabilidad, parece lógico pensar que la tarea t11 sea la que obtenga el menor tiempo de respuesta, ya que tiene el menor de los deadlines. No obstante, la siguiente tarea con menor tiempo de ejecución es la tercera, cuyo deadline es superior a la segunda.

Esto se puede explicar viendo la asignación de prioridades realizada por MAST. En la CPU1 la tarea t23 del flujo 2 tiene menos prioridad que la tarea t31 del flujo 3 y que la tarea t11 del flujo 1 y además su periodo es el mayor de todos. En el peor caso, la tarea t23 se encuentra bloqueada por las otras dos tareas del mismo procesador y por activaciones de estas dos tareas de nuevo siguiendo su periodo, podría volver a bloquearse. A esto hay que sumarle que en total es el flujo que tiene la mayor suma de tiempos de ejecución de tareas.

Para modificar el programa y asignar las prioridades obtenidas en el análisis, basta con modificar el valor asignado en el atributo `sched_priority` de la variable `sch_param`.

A continuación, un ejemplo con el primer hilo del flujo 1:

```
// Set the priority of thread 1 to 6554
sch_param.sched_priority = 6544;
if (pthread_attr_setschedparam
    (&attr,&sch_param) != 0)
{
    printf("Error en atributo schedparam\n");
    exit(1);
}
```

El número de prioridad asignado no es relevante siempre y cuando se mantenga el orden de prioridad entre todas las tareas en el sistema, pero he preferido asignar exactamente el número obtenido en el análisis para no complicar el problema.

Una vez modificados ambos programas, hay que lanzar cada 1 en un ordenador distinto dentro del laboratorio y anotar los tiempos de respuesta.

Los tiempos de respuesta obtenidos han sido:

- Tiempo de respuesta flujo 1: Infinito
- Tiempo de respuesta flujo 2: 31.25
- Tiempo de respuesta flujo 3: 19.05

Tras dejar el programa en ejecución en ambos ordenadores, los tiempos de respuesta de los flujos 2 y 3 se estancan en los valores proporcionados. Sin embargo, el tiempo de respuesta del flujo 1 crece sin parar.

En cuanto al porqué de este comportamiento, lo único que se me ocurre es que el análisis de planificabilidad que se ha utilizado en MAST, usando el algoritmo de planificación PD, obtenga unas prioridades que, a la hora de ejecutar el programa en Marte, no sean las óptimas.

Actividad 2

Trabajar con la aplicación de la Actividad 1 basada en FP para adaptarla al cumplimiento de unos plazos más estrictos.

Repetir el experimento de la Actividad 1 teniendo en cuenta las siguientes consideraciones:

- se deben modificar los plazos de principio a fin para que sean iguales a los periodos
- asignar de nuevo prioridades con MAST y analizar el sistema calculando los slacks
- modificar los tiempos de ejecución de la aplicación de acuerdo con el slack del sistema

- realizar el análisis de planificabilidad y anotar los tiempos de respuesta de las actividades con requisitos temporales
- medir esos tiempos de respuesta ejecutando la aplicación (anotarlos).

Comentar los resultados que se han obtenido, y en particular:

- explicar los resultados en comparación con los de la Actividad 1
- observar los slacks (del sistema, de los procesadores y de los flujos e2e) obtenidos antes de modificar los tiempos de ejecución de las actividades y comentarlos en el informe

Primero de todo se piden igualar los plazos de ejecución de cada tarea a su periodo.

flujo1:TRANSACTION	flujo2:TRANSACTION	flujo3:TRANSACTION
Periodic: T=15.000 D=15.000	Periodic: T=40.000 D=40.000	Periodic: T=30.000 D=30.000

A continuación, hay que analizar el sistema asignado prioridades y calculando los slacks. Para el análisis del sistema se utiliza la técnica de asignación de prioridades NP.

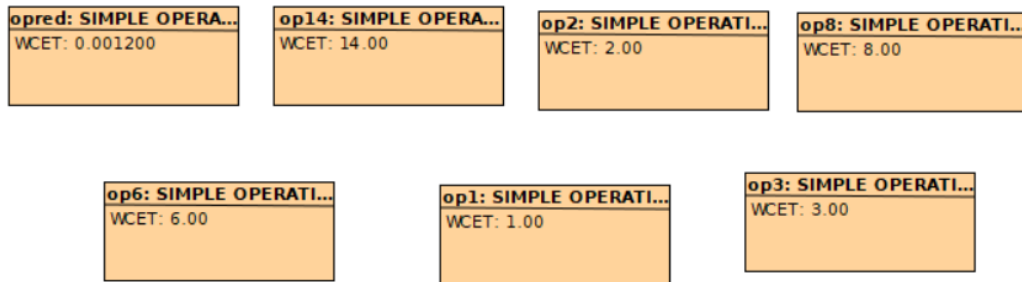
Las prioridades obtenidas son:

- Operación t11: 6554
- Operación t13: 8192
- Operación t21: 16384
- Operación t23: 1
- Operación t31: 1307
- Operación t33: 1

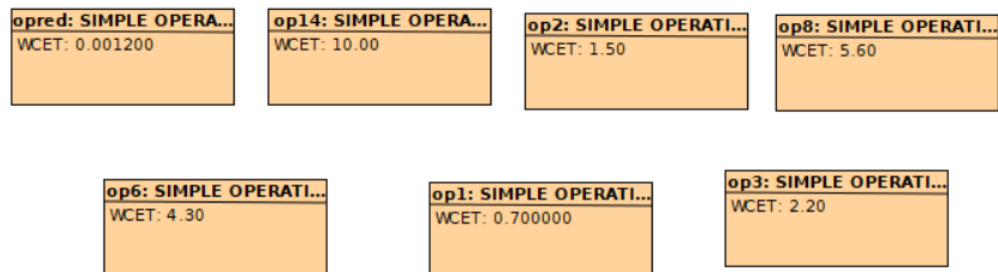
El slack obtenido del sistema es de **28,91%**.

El siguiente paso es modificar los tiempos de ejecución del sistema para ajustarse al slack obtenido.

Los tiempos de ejecución de las tareas del sistema para el cuál se obtenía un slack del 28.91% eran:



Los tiempos de ejecución de las tareas del sistema tras restarles el slack obtenido son:



Es importante destacar que se han hecho redondeos para simplificar el problema.

Con estos tiempos el slack obtenido al analizar el sistema es de **-0.390625%**

Los tiempos de respuesta obtenidos tras hacer el análisis son:

- Tiempo de respuesta flujo 1: 22.200
- Tiempo de respuesta flujo 2: 28.107
- Tiempo de respuesta flujo 3: 21.307

El último paso es medir estos tiempos de ejecución con MARTE. Para ello hay que modificar los tiempos de ejecución de los hilos definidos en los programas *periodic_FP_1* y *periodic_FP_2*

Ejemplo modificando los tiempos del hilo t11:

Antes de la modificación:

```

data1.period.tv_sec =1;
data1.period.tv_nsec=500000000;
data1.wcet1.tv_sec =0;
data1.wcet1.tv_nsec=600000000;
data1.wcet2.tv_sec =0;
data1.wcet2.tv_nsec=200000000;
data1.phase.tv_sec =3;
data1.phase.tv_nsec=0;
data1.sem=&sem1;
data1.id=1;
message1[0]='1';
data1.message=message1;

```

Después de la modificación:

```

data1.period.tv_sec =1;
data1.period.tv_nsec=500000000;
data1.wcet1.tv_sec =0;
data1.wcet1.tv_nsec=430000000;
data1.wcet2.tv_sec =0;
data1.wcet2.tv_nsec=150000000;
data1.phase.tv_sec =3;
data1.phase.tv_nsec=0;
data1.sem=&sem1;
data1.id=1;
message1[0]='1';
data1.message=message1;

```

Por falta de tiempo, no han podido obtenerse las medidas reales en los ordenadores industriales. No obstante, mi compañera Susana ha obtenido el mismo slack en el sistema con los plazos iguales a los periodos y también los mismos tiempos de respuesta de los flujos tras haber ajustado los tiempos de ejecución de las tareas de acuerdo al slack obtenido, por lo que los tiempos de respuesta obtenidos en MARTE hubieran sido los mismos.

Los tiempos de respuesta obtenidos han sido:

- Tiempo de respuesta flujo 1: 16.38
- Tiempo de respuesta flujo 2: 18.66
- Tiempo de respuesta flujo 3: 9.96

Comparando estos resultados con los de la actividad 1, se ve como son mucho menores. Además, se aprecia como coinciden bastante con la suma de cada una de las tareas de cada flujo por separado. Se me ocurren dos posibles explicaciones para esto. La primera es que o bien no se ha dejado suficiente tiempo de ejecución para obtener tiempos de peor caso, o bien el entorno de ejecución considerado por MAST tiene en cuenta más

tareas de ejecución en segundo plano (como tareas de kernel) que tienen más prioridad y por ello se retrasan las tareas del análisis. Esto último lo considero poco improbable dado que creo con seguridad que MAST no trabaja así, con lo que me inclino por la primera opción.

Por último, se pide obtener los slacks del sistema, de los procesadores y de los flujos e2e **antes** de modificar los tiempos de ejecución de cada actividad, es decir, los de la actividad 1.

En mi caso los slacks son:

- Sistema: -28.91%
- Cpu1: -90.07%
- Cpu2: -98.44%
- Red: -98.44%
- Flujo 1: -37.89%
- Flujo 2: -100.00%
- Flujo 3: -78.13%