

Diagramabilidad de Sistemas de Tiempo-Real Multitarea-Monoprocesador con Relaciones de Precedencia

Ricardo Cayssials, Javier Orozco y Edgardo Ferro

Universidad Nacional del Sur, Departamento de Ing. Eléctrica
Instituto de Ciencias e Ingeniería de la Computación - Bahía Blanca - Argentina
e-mail: {iecayss, ieorozco, ieferro}@criba.edu.ar

Resumen

En este trabajo se muestran algunos resultados obtenidos a partir del análisis de sistemas de tiempo-real diagramados por un diagramador por Períodos Monotónicos Crecientes (PMC). Los mismos muestran que la diagramabilidad del sistema aumenta si se establecen relaciones de precedencia entre las tareas. Luego, una adecuada partición de tareas en subtareas con relaciones de precedencia de menor duración, facilita la diagramabilidad del sistema.

Palabras Clave: Tiempo-Real, Diagramabilidad, Multitarea-monoprocesador, Precedencia

1. Introducción

El problema de diagramar un conjunto de tareas en sistemas multitarea-monoprocesador es de complejidad polinomial y desde el trabajo de Liu y Layland [1], diversos métodos han sido propuestos para resolverlo con eficacia creciente tanto desde el punto de vista de la utilización del procesador [2, 3, 4, 5] como el de compartir otros recursos [6, 7] operando bajo distintas disciplinas de prioridades. De ellas, la más importante, por ser un estándar *de facto* impuesto por el Departamento de Defensa de EEUU, es la denominada de Períodos Monotónicos Crecientes, PMC (*Rate Monotonic Scheduling*, RMS).

Un problema adicional, no tratado en [1] es que, entre pares de tareas, puede haber una relación de precedencia. Se entiende por ello que, para comenzar su ejecución, una de ellas (*sucesora*) necesita datos producidos por otra (*predecesora*). La llegada de dichos datos desde todas las predecesoras al procesador que aloja a la tarea sucesora define (en el contexto de este trabajo) el instante de arribo de la misma. Por otro lado, si las tareas no poseen relaciones de precedencia entre ellas se dice que las tareas son independientes.

En este trabajo se mostrará que, cuando existen relaciones de precedencia entre tareas, aumentan las posibilidades de diagramar el sistema para igual factor de utilización del procesador.

El método de las ranuras vacías será utilizado para la determinación de la diagramabilidad.

2. Relaciones de Precedencia

Es usual en Sistemas de Tiempo Real (STR) que las tareas no sean independientes y cooperen entre sí para lograr un determinado objetivo. Cuando una tarea τ_j necesita total o parcialmente los resultados producidos por otra tarea τ_i para poder ser ejecutada, se

establece una relación de precedencia, notada $\tau_i \prec \tau_j$ (τ_i precede a τ_j), la cual establece un orden parcial sobre las tareas. Si $\tau_i \prec \tau_j$ y no existe una $\tau_l | \tau_i \prec \tau_l$ y $\tau_l \prec \tau_j$ entonces τ_i y τ_j son denominadas predecesora y sucesora respectivamente. Este modelo de computación puede ser asociado a un grafo \mathbf{G} [11]. Cada nodo del mismo simboliza una tarea y existirá un arco dirigido desde el nodo que representa τ_i al nodo que representa τ_j si y sólo si ellas son predecesora y sucesora respectivamente. $\mathbf{G} = \{\Gamma, \mathbf{P}\}$, donde Γ es el conjunto de tareas que forman el trabajo y \mathbf{P} es el conjunto de sus relaciones de precedencia.

Una relación de precedencia agrega complejidad a la diagramación. En [13] se ha resuelto en forma óptima, con complejidad $\mathbf{O}(n^2)$, un problema simple de un conjunto de tareas no apropiativas, idéntico tiempo de arribo y con condiciones de precedencia. Este método, lamentablemente, no es suficiente para la mayoría de los problemas de interés en STR, en los cuales las tareas no tienen un instante conocido de arribo. Los mejores resultados sobre precedencia fueron obtenidos a partir del trabajo sobre subclases de la relación de precedencia, alcanzándose algoritmos polinómicos para relaciones *intrees* (un solo sucesor) y *outtrees* (un solo predecesor). Estos casos son incluidos en un grupo más interesante denominado serie-paralelo que es definido en forma recursiva y sobre el cual existen soluciones eficientes, aunque, lamentablemente, los problemas no son comunes en la práctica. Si el sistema es apropiativo, se reduce la complejidad del diagramador y es solucionable en $\mathbf{O}(n^2)$ aplicando el algoritmo de Baker [15].

En Chetto se obtienen condiciones de diagramabilidad para tareas con relaciones de precedencia en sistemas multitarea-monoprocesador bajo la disciplina de menor tiempo a crisis. Sin embargo, los resultados son escasos aplicando una diagramación con la disciplina PMC.

3. Diagramación de tareas con relaciones de precedencia en sistemas monoprocesador.

En esta sección se demostrará que la disciplina PMC en sistemas multitarea-monoprocesador, con relaciones de precedencia entre las tareas no es la óptima entre las disciplinas de prioridades fijas. En un sistema multitarea-monoprocesador con tareas independientes el peor caso de carga es la generación simultánea de todas las tareas [1]. Al existir relaciones de precedencia, no todas las tareas podrán generarse simultáneamente. Al necesitar las tareas sucesoras la completa ejecución de sus respectivas predecesoras no podrá darse el arribo simultáneo de la sucesora y sus correspondientes predecesoras. Esta consideración permite prever que en sistemas con tareas con una adecuada relación de precedencia, el peor caso de carga puede ser más fácilmente diagramable que en sistemas con tareas independientes, mejorando así la utilización del procesador. A continuación se ilustrará que el aprovechamiento de las relaciones de precedencia depende del ordenamiento que posean las tareas en la pila de prioridades.

3.1. Resultado 1: alternativas en la asignación de prioridades.

Consideraremos un sistema: $\mathbf{S}(5)=\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, $\tau_4 \prec \tau_5$, mientras que τ_1, τ_2 y τ_3 son tareas independientes Fig. 1. Consideremos también las distintas alternativas que pueden darse en el ordenamiento de las prioridades de las tareas.

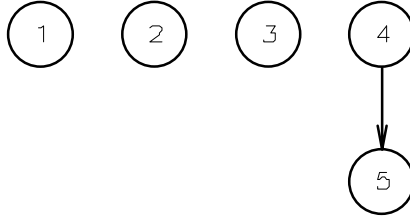


Figura 1

3.1.1. Alternativa 1

Consideremos el sistema con la siguiente pila de prioridades $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$ siendo τ_1 la tarea de mayor prioridad. Notaremos que esta pila de prioridades cumple con la disciplina de prioridades de períodos monotónicos crecientes. De [[5], para que un sistema PMC resulte diagramable debe cumplirse que:

$$\forall i, \quad T_i \geq D_i \geq \min t | t = C_i + \sum_{h=1}^{i-1} C_h \cdot \left\lceil \frac{t}{T_h} \right\rceil = e_{C_i(S(i-1))} \quad (1)$$

En [[11] se muestra que un sistema con inversión de prioridades será diagramable si:

$$\forall i, \quad T_i \geq e_{(C_i+k)(S(i-1))} \quad (2)$$

donde k es el máximo número de inversiones a la que puede verse sometida τ_i .

Luego, para que τ_3 y el trabajo formado por las tareas τ_4 y τ_5 resulten diagramables, debe satisfacerse:

$$\begin{aligned} D_3 &\geq \min t | t = C_3 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil \\ D_5 &\geq \min t | t = C_5 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil + C_3 \cdot \left\lceil \frac{t}{T_3} \right\rceil + C_4 \cdot \left\lceil \frac{t}{T_4} \right\rceil = \\ &= C_4 + C_5 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil + C_3 \cdot \left\lceil \frac{t}{T_3} \right\rceil \end{aligned} \quad (3)$$

Nótese que en la expresión anterior $\left\lceil \frac{t}{T_4} \right\rceil = 1$ ya que las tareas τ_4 y τ_5 al estar vinculadas por una relación de precedencia tienen el mismo período y, ser contiguas en la pila de prioridades pueden considerarse como una tarea cuyo tiempo de ejecución de $C_4 + C_5$. Un razonamiento análogo es aplicado en [[5] para extender el método de las ranuras vacías a sistemas con tareas con tiempos de ejecución no unitarios.

3.1.2. Alternativa 2

Consideremos el sistema con la siguiente pila de prioridades $\{\tau_1, \tau_2, \tau_3, \tau_5, \tau_4\}$, siendo τ_1 la tarea de mayor prioridad. Este caso de ordenamiento provoca un comportamiento del sistema idéntico al de la alternativa anterior. La tarea τ_3 debe satisfacer la misma

inecuación que en la alternativa 1, mientras que el trabajo compuesto por las tareas τ_4 y τ_5 será diagramable si la (C_4+C_5) -ésima ranura libre que deja el sistema $\mathbf{S}(3)$, $e_{(C_4+C_5)(\mathbf{S}(3))}$, es menor que el vencimiento de dicho trabajo.

3.1.3. Alternativa 3

Consideremos el sistema con la siguiente pila de prioridades $\{\tau_1, \tau_2, \tau_4, \tau_3, \tau_5\}$, con τ_1 tarea de mayor prioridad. Para que el sistema sea diagramable, deben satisfacerse las siguientes inecuaciones:

$$D_3 \geq \min t \mid t = C_3 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil + C_4 \cdot \left\lceil \frac{t}{T_4} \right\rceil$$

$$D_5 \geq \min t \mid t = C_4 + C_5 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil + C_3 \cdot \left\lceil \frac{t}{T_3} \right\rceil$$

El vencimiento de la tarea τ_3 debe ser mayor que en la alternativa 1 debido a que en el peor caso de generación debe tolerar la ejecución de τ_4 por ser de mayor prioridad. Sin embargo, para la tarea τ_5 el vencimiento permanece invariante debido a que sólo se modifica el sumando de la tarea τ_4 . Como t debe ser menor o igual a T_4 para que el trabajo compuesto por las tareas τ_4 y τ_5 sea diagramable, el factor que tiene C_4 , por la ecuación de las ranuras vacías es igual a uno. De esta manera podemos afirmar que si la tarea predecesora posee mayor prioridad que la sucesora, ambas tareas pueden considerarse, para el estudio de la diagramabilidad de la sucesora, como una tarea con tiempo de ejecución igual a la suma de los tiempo de ejecución de ambas tareas, y con prioridad igual a la prioridad de la sucesora.

3.1.4. Alternativa 4

Consideremos el sistema con la siguiente pila de prioridades $\{\tau_1, \tau_2, \tau_5, \tau_3, \tau_4\}$, siendo τ_1 la tarea de mayor prioridad. Para que el sistema sea diagramable, deben satisfacerse las siguientes inecuaciones:

$$D_3 \geq \min t \mid t = C_3 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil + C_5 \cdot \left\lceil \frac{t}{T_5} \right\rceil$$

τ_5 sólo podrá ejecutarse luego de τ_4 y ambas deberán soportar la carga producida por τ_3 hasta la completa ejecución de τ_4 . Esta finalizará su ejecución en $e_{C_4((\tau_1, \tau_2, \tau_3))}$, ya que τ_5 no puede interferir su ejecución. Luego, si en (2), $k = \left(C_3 \cdot \left\lceil \frac{e_{C_4((\tau_1, \tau_2, \tau_3, \tau_4))}}{T_3} \right\rceil \right)$ y

$$D_5 \geq \min t \mid t = \left(C_3 \cdot \left\lceil \frac{e_{C_4((\tau_1, \tau_2, \tau_3))}}{T_3} \right\rceil \right) + C_4 + C_5 + C_1 \cdot \left\lceil \frac{t}{T_1} \right\rceil + C_2 \cdot \left\lceil \frac{t}{T_2} \right\rceil \quad (4)$$

Si tomamos t_1 y t_4 como las soluciones de las expresiones (3) y (4) de las alternativas 1 y 4 respectivamente vemos que t_1 es la primer ranura libre que deja el sistema $\{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$ y $e_{C_4((\tau_1, \tau_2, \tau_3))}$ es la primer ranura libre que deja el sistema $\{\tau_1, \tau_2, \tau_3\}$ con lo que

$$t_1 > e_{C_4((\tau_1, \tau_2, \tau_3))}, \text{ luego } C_3 \left[\frac{t_1}{T_3} \right] \geq C_3 \left[\frac{e_{C_4((\tau_1, \tau_2, \tau_3))}}{T_3} \right]$$

De esta manera, la inecuación que debe satisfacer la tarea τ_3 es menos restrictiva respecto a las alternativas anteriores.

Del estudio de los diferentes ordenamientos que pueden tener las tareas con relación de precedencia, se hace notoria la conveniencia de otorgar a las tareas sucesoras mayor prioridad que a las predecesoras.

3.2. Resultado 2: prioridades y precedencias

Otro resultado interesante muestra la conveniencia de particionar las tareas en subtareas con una relación de precedencia entre ellas que ya que así el sistema presenta más posibilidades de resultar diagramable. Por ejemplo: es preferible tener dos tareas predecesora y sucesora con tiempos de ejecución 1 en lugar de una tarea con tiempo de ejecución 2. Esto se debe a que las relaciones de precedencia tienen un mecanismo implícito de diagramación que, utilizado correctamente, puede mejorar la condiciones de diagramabilidad del sistema. Para ilustrar el efecto que presentan las relaciones de precedencia en la diagramación, consideremos el siguiente ejemplo:

Sea el sistema $S(3) = \{\tau_1, \tau_2, \tau_3\}$, con los siguientes parámetros:

i	C_i	T_i
1	3	6
2	3	8
3	1	8

Tabla I

La figura 2 muestra la evolución del sistema en el peor caso de carga. Como puede observarse, la tarea τ_3 alcanza su vencimiento en la ranura 7 al no poder ejecutarse.

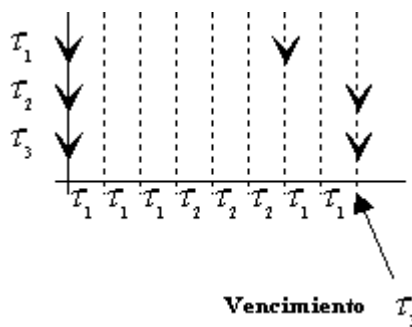


Figura 2

Estudiemos el caso en que las tareas τ_1 y τ_2 están compuestas, cada una, por un par de tareas sucesora y predecesora con tiempo de ejecución unitario. τ_1 se descompondrá en τ_{1a} y τ_{1b} predecesora y sucesora respectivamente. τ_2 se descompondrá en τ_{2a} y τ_{2b} predecesora y sucesora respectivamente. El sistema queda conformado por $S(5) = \{\tau_{1a}, \tau_{1b}, \tau_{2a}, \tau_{2b}, \tau_3\}$, con los parámetros mostrados en la Tabla II:

i	C_i	T_i
τ_{1a}	2	6
τ_{1b}	1	6
τ_{2a}	1	8
τ_{2b}	2	8
τ_3	1	8

Tabla II

Consideremos que el diagramador del sistema tiene la siguiente pila de prioridades: τ_{1b} , τ_{2b} , τ_3 , τ_{1a} , τ_{2a} siendo τ_{1b} la tarea de mayor prioridad. Las figuras 3.a, 3.b y 3.c muestran los peores casos de generación para los trabajos τ_1 , τ_2 y τ_3 respectivamente. Las tareas sucesoras son generadas al término de ejecución de su correspondiente predecesora.

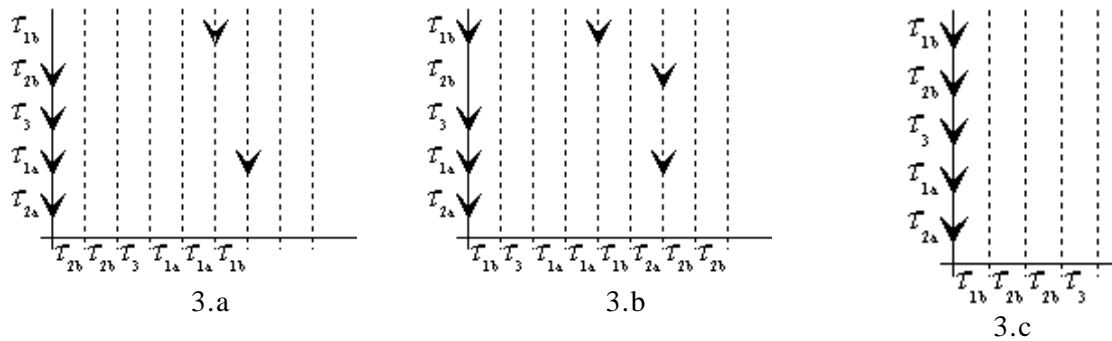


Figura 3

Puede observarse que ninguna tarea alcanza su vencimiento. También puede notarse que las tareas no están ordenadas por períodos monotónicos crecientes, y que el sistema es diagramable mediante prioridades fijas debido a la diagramación implícita que tienen las relaciones de precedencia.

Nuevamente es posible observar que, para obtener relaciones de diagramación menos rigurosas, es conveniente asignar mayores prioridades a las tareas sucesoras que a sus respectivas predecesoras. De esta manera es aprovechada la diagramación interna que presentan las relaciones de precedencia entre las tareas.

4. Conclusiones

En este trabajo se muestran algunos resultados que revelan la conveniencia de establecer relaciones de precedencia en sistemas multitarea-monoprocesador que son diagramados utilizando una disciplina de prioridades fijas. Por otro lado, una adecuada partición de tareas en subtareas con relaciones de precedencia de menor duración, facilita la diagramabilidad del sistema. Por supuesto que existe una relación de compromiso entre la eficiencia del diagramador y el aumento del número de tareas. Esta debe ser tomada en cuenta por la sobrecarga que produce, en determinadas implementaciones, al sistema operativo. Por otro lado es posible, en términos prácticos, que la partición de una tarea de tiempo de ejecución C_i en n subtareas conduzca a un conjunto, cuyo tiempo de ejecución, total sea mayor que el de la tarea original ($\sum_{h=1}^n C_{i,h} \geq C_i$). Esto se debe a que resulta necesario implementar mecanismos de comunicación entre las tareas que previamente no eran necesarios. A pesar de la complejidad del problema su tratamiento es importante ya

que, como se ha ejemplificado, existen problemas intratables para un diagramador PMC estándar que, con un adecuado tratamiento de las relaciones de precedencia entre subtareas, puede tornarse factible.

Actualmente se encuentra en desarrollo la formalización de estos resultados y una metodología que permita sistematizar la partición y asignación de prioridades a las subtareas.

Referencias

- [1] Liu, C.L. and Layland, J.W., "Scheduling algorithms for multiprograming in hard real time environments", *J. ACM*, Vol. 20, No 1, pág. 46-61, 1973.
- [2] Joseph, M. and Pandya, P., "Finding Response Times in a Real-Time System", *The Computer Journal*, Vol. 29, N° 5:390-395, 1986.
- [3] Leung J., Whitehead J., "On the complexity of Fixed Priority Sheduling of periodic, Real - Time tasks", *Perfomance Evaluation*, Vol. 2, N° 4, pág. 237-250, 1982.
- [4] Lehozcky, J.P., Sha, L. and Ding, Y., "The Rate Monotonic Scheduling Algorithm: Exact characterization and average case behavior", In *Proc. Real Time Symp. IEEE CS*, Los Alamitos, CA. 1989.
- [5] Santos, J. and Orozco, J., "Rate Monotonic Scheduling in Hard Real-Time Systems", *Information Processing Letters*, N° 48, pág. 39-45, 1993.
- [6] Sha L., Rajkumar R., Lehozcky J.P., "Priority Inheritance Protocols: An aproach to Real-Time Synchronization", *IEEE Trans. Computers*, Vol. 39, N° 9,pág 1175-1185, 1990.
- [7] Zhao W., Ramamritham K., Stankovic J., "Preemptive Scheduling under Time and Resource Constraints", special issue on *Real-Time Systems*, *IEEE Trans. Computers*, Vol. 36, N° 8, pág. 949-960, 1987.
- [8] Tindell K., Burns A., Wellings A., "Allocating hard Real-Time tasks: An NP-hard problem made easy", *Real Time Systems*, pág. 145-166, 1992.
- [9] Lawler, "Optimal Sequencing of a Single Machine Subject to Precedence Constraints", *Management Science*, N° 19, 1973.
- [10] Ramamritham, K., "Allocation and scheduling of complex periodic task", In *Proc. 10th International Conference on Distributed Computing Systems*, pp. 108-115, 1990.
- [11] Orozco and J. Santos. 1994. "Diagramación de tareas en sistemas operativos de tiempo real PMC con inversión de prioridades". *23 Jornadas Argentinas de Informática e Investigación Operativa*, pp. 1.25-1.35.