

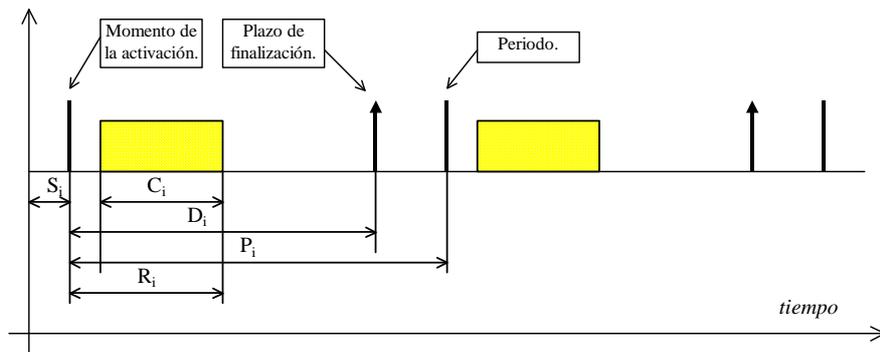
Introducción

- En un programa concurrente no se indica el orden en que se ejecutan las tareas, se utilizan primitivas de sincronización.
- Aunque los resultados siempre sean correctos, el orden en que se ejecutan las tareas influye en su instante de finalización.
- Solo unas pocas combinaciones del orden de ejecución de las tareas dará una respuesta temporal correcta.
- En los STR es necesario poder restringir el orden de ejecución, seleccionando sólo el orden en que se cumplen las restricciones temporales.

Tipos de tareas

- Según sus características temporales
 - Periódicas
 - Esporádicas
 - Aperiódicas
- En base a las características semánticas
 - Críticas
 - Opcionales
 - Con plazo
 - Sin plazo

Parámetros utilizados



Planificación - I

3

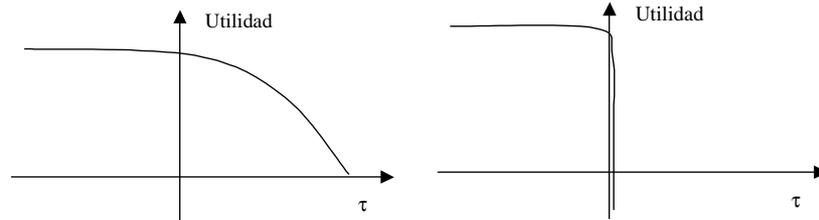
Tiempo real crítico y opcional

- Tardanza: $\tau_i = R_i - D_i$
- Una tarea se ejecuta de forma tardía si su tardanza se hace positiva.
- En los sistemas de tiempo real, un resultado puede seguir siendo útil aunque se obtenga de forma tardía (vencido su plazo).
- En los sistemas de tiempo real crítico, la utilidad descende bruscamente cuando la tardanza se hace positiva.

Planificación - I

4

Tiempo real crítico y opcional



Tiempo real opcional

Tiempo real crítico

Planificación - I

5

Tiempo real crítico y opcional

- La utilidad es una magnitud difícil de medir y de validar
- En su lugar se puede utilizar la probabilidad de obtener una respuesta, y representarlo en función de la tardanza (campana de Gauss alrededor del valor medio)
- La utilidad se expresará como un valor de la probabilidad de que el tiempo de respuesta sea menor o igual a una cota temporal (área de la campana de Gauss)
- Un sistema de tiempo real crítico será aquel en que la probabilidad que se cumplan las cotas temporales es 1

Planificación - I

6

Tiempo real crítico y opcional

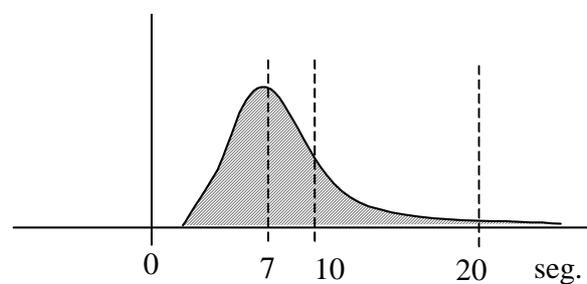
- Expresión de las restricciones temporales
 - Deterministas: Valor absoluto de los límites temporales
 - Probabilísticas: Probabilidad de obtener un resultado después de un límite
 - Función de utilidad: Como un valor de la utilidad
- En la práctica, las restricciones críticas se expresan de forma determinista mientras que las opciones se expresan en términos de probabilidad o utilidad

Planificación - I

7

Tiempo real crítico y opcional

- Ejemplo de sistema opcional: Tiempo de conexión de una llamada en un sistema telefónico:
 - Antes de 10 seg el 95% de los casos
 - Antes de 20 seg el 99.95% de los casos



Planificación - I

8

Modelo de referencias de los STR

- Modelo teórico sencillo
- Tiempo de ejecución de las tareas
- Modelo de tareas periódicas. Definiciones
- Dependencias entre tareas
- Parámetros funcionales
- Planificadores
- Restricciones del modelo simple (resumen)

Tiempo de ejecución de una tarea

- Tiempo variable entre un máximo y un mínimo
- Le afectan los recursos compartidos
- Se utiliza un modelo determinista
 - Muchos sistemas requieren un alto nivel de seguridad
 - Se evita el uso de los elementos no deterministas
- No todos los procesos son críticos
- Supondremos un sistema determinista en el que el tiempo de ejecución en el peor de los casos de todas las tareas críticas son conocidos
- El número de tareas críticas será fijo y conocido

Modelo de tareas periódicas

- Suponemos todas las tareas periódicas
- Necesario para limitar la carga del sistema
- Las tareas esporádicas se convierten en periódicas
- Definiciones:
 - Sistema síncrono
 - Factor de utilización
 - Trabajo o activación
 - Hiperperiodo

Dependencias entre tareas

- Tipos de dependencias temporales:
 - Restricciones de precedencia
 - Dependencia en los datos
 - Dependencia temporal relativa
 - Restricciones de precedencia AND / OR
 - Ejecución condicional
 - Relación de cauce (productor / consumidor)
- Consideraremos todas las tareas independientes
- Posteriormente consideraremos la dependencia en los datos

Parámetros funcionales

- Trabajos interrumpibles
 - Se puede suspender su ejecución en cualquier momento para dar paso a un trabajo más prioritario
 - Cambio de contexto y tiempo de cambio de contexto
 - Suponemos tareas interrumpibles y el tiempo de cambio de contexto despreciable
- Importancia de los trabajos
 - Se indica asignando una prioridad
 - Se ejecuta la de mayor prioridad entre las activas
- Ejecución opcional
 - Distinción entre trabajos opcionales y obligatorios
 - Suponemos todos los trabajos obligatorios

Planificadores

- Planificador: el módulo que decide que trabajo se ejecuta en cada procesador y en que orden
- Planificación: Una asignación concreta de trabajos
- Un planificador debe producir planificaciones válidas
- Una planificación válida es posible si cada trabajo finaliza antes de su límite temporal
- Un conjunto de tareas es planificable bajo un planificador si este siempre produce planificaciones posibles
- Un planificador es óptimo si, cuando existe, es capaz de encontrar una planificación posible
- Cuando un planificador falla, puede:
 - Incumplir los límites temporales de algunas tareas
 - No ejecutar algunas tareas opcionales

Restricciones del modelo simple

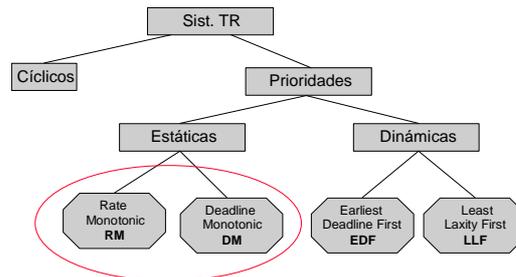
1. Sistema formado por un único procesador
2. El conjunto de tareas es estático
3. Se conoce el WCET de todos los trabajos
4. Todos los trabajos son interrumpibles
5. El tiempo de cambio de contexto es nulo
6. Todas las tareas son periódicas
7. Los plazos de finalización de todas las tareas son iguales a sus periodos
8. Las tareas son independientes

Políticas de planificación

- Un conjunto de aspectos que engloban el diseño e implementación de un sistema de tiempo real y que está formado por:
 1. Test de garantía off-line
 2. Un **algoritmo de planificación**
 3. Los protocolos de acceso a recursos compartidos
 4. La forma de servir las tareas aperiódicas
- En el modelo simple no se incluyen 3 y 4, pero se introducirán más adelante

Políticas de planificación (grupos)

- Cíclicos: Se coloca “a mano” el orden de ejecución de los trabajos durante un hiperperiodo
- Por prioridades: Se asigna una prioridad a las tareas en base a su importancia:
 - Prioridades estáticas
 - Prioridades dinámicas

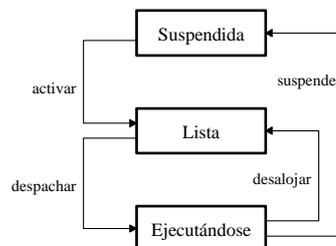


Planificación - I

17

Planificación por prioridades

- Asigna un valor de prioridad a cada tarea en cada momento (0 mayor prioridad)
- Se ejecuta la tarea activa más prioritaria
- Si la prioridad es estática una tareas no varia su prioridad. Si es dinámica la modifica el planificador
- Estados importantes:
- Test de garantía



Planificación - I

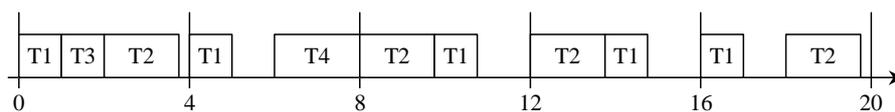
18

Planificadores cíclicos

- Sólo son aplicables cuando existe un grado alto de determinismo. Asumimos el modelo restringido de tareas periódicas:
 - El número de tareas N es constante
 - Se conocen los parámetros de todas las tareas
 - Cada trabajo $J_{i,k}$ está listo para ser ejecutado a partir de su tiempo de activación $r_{i,k}$
 - Existe un único procesador en el sistema
- Los parámetros de una tarea se definen por la 4-tupla (S_i, P_i, C_i, D_i)
- Por defecto el desfase se toma cero, y el plazo de finalización el periodo. Entonces se abrevia con (P_i, C_i, D_i) o (P_i, C_i) respectivamente

Planificaciones cíclicas

- Se define de antemano cuando se ejecuta cada trabajo
- El tiempo de CPU para cada trabajo será su tiempo máximo de ejecución
- El planificador activa los trabajos según esta planificación. Salvo errores, todas las tareas deben cumplir sus plazos
- Ejemplo: $T_1 = (4, 1)$, $T_2 = (5, 1.8)$, $T_3 = (20, 1)$ y $T_4 = (20, 2)$



- Los intervalos vacíos se pueden usar por las tareas aperiódicas

Planificaciones cíclicas (implementación)

- Se guarda en una tabla de entradas $(t_k, T(t_k))$
- Funcionamiento:
 - El planificador utilizará un timer para activar cada una de las entradas de la tabla
 - En la inicialización, el sistema debe crear todas las tareas que se van a ejecutar, reservándoles todos los recursos que vayan a necesitar
 - Tras inicializar las tareas, se programará el timer para que active al planificador en el instante t_1 y lanzará la ejecución de la tarea $T(t_0)$
 - En cada interrupción del timer en el instante t_k se lanzará la tarea $T(t_k)$ y se programará el timer para que despierte al planificador en el instante t_{k+1}
 - Cuando se alcance el final de la tabla al final del hiperperiodo se volverá a leer la tabla desde el principio otra vez
- Las planificaciones cíclicas también se llamas estáticas periódicas

Planificadores cíclicos con tramas

- Las activaciones en instantes arbitrarios puede ser un inconveniente
- Es más conveniente realizar las activaciones con cierta periodicidad. Se divide el tiempo de ejecución en tramas
- La duración de la trama se llama tamaño de trama: f
- Con las activaciones, al principio de cada trama se pueden realizar otras acciones. Por ejemplo, comprobar:
 - Que cada trabajo a ejecutar ha sido activado
 - Si hay overrun en la trama anterior
- Unos valores de tramas serán mas adecuados que otros

Condiciones sobre el tamaño de trama

1. Es deseable que cada trabajo comience y finalice en la misma trama:

$$f \geq \max(C_i), 1 \leq i \leq n$$

2. Que f sea divisor de H , para reducir el ciclo de ejecución ($H=F*f$):

$$P_i / f - \lfloor P_i / f \rfloor = 0$$

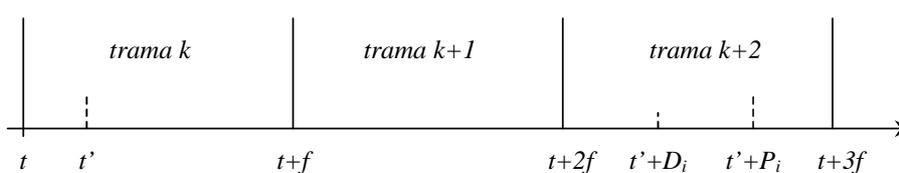
3. Para que se pueda determinar cuando un trabajo finaliza antes de su plazo, entre el instante de activación y el plazo de cualquier trabajo debe existir una trama completa

$$2f - \text{mcd}(P_i, f) \leq D_i$$

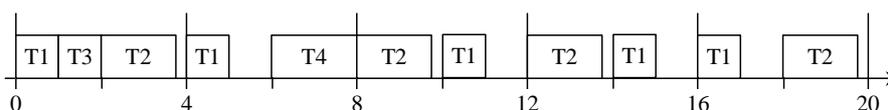
Planificación - I

23

Demostración de la condición 3



- Si $t = t'$ basta con que f sea menor que D_i
- Si $t' > t$, entonces debe ocurrir que $t+2f \leq t'+D_i$
- Como $(t' - t) \geq \text{mcd}(P_i, f)$, se cumple la desigualdad
- El ejemplo anterior, con $f = 2$, se cumplen las tres condiciones:

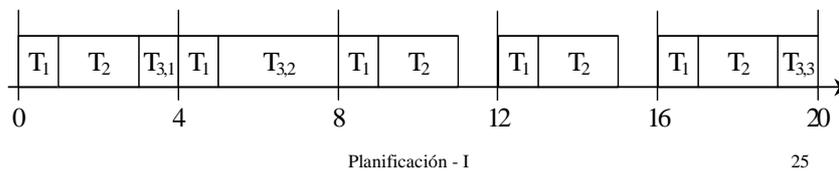


Planificación - I

24

Planificadores cíclicos con tramas

- En ocasiones no se pueden cumplir las tres condiciones. Por ejemplo en el sistema $T = \{(4, 1), (5, 2, 7), (20, 5)\}$
 - Para (1) $f \geq 5$, pero según (3) $f \leq 4$
- Hay que dividir cada trabajo de las tareas con C_i mayor que f en porciones que sean menores
- Podemos dividir T_3 en las tres subtareas $(20, 1)$, $(20, 3)$ y $(20, 1)$, y ya se puede tomar $f = 4$



Diseño de un planificador cíclico

- Decisiones
 - Elegir el tamaño de la trama
 - Dividir los trabajos en rodajas
 - Situar las rodajas en las tramas
- Consideraciones
 - Dividir en las menos rodajas posibles para no penalizar el rendimiento
 - Pero si las rodajas son grandes es posible que no se pueda construir un planificador válido
 - Si los trabajos no son interrumpibles, con rodajas pequeñas será más fácil encontrar problemas al partir los trabajos

Planificación - I

26

Algoritmo para construir el planificador

- La elección de la longitud de trama y planificar los trabajos es un problema NP-duro
- Considerando el caso de **trabajos interrumpibles**, veremos un algoritmo donde el tiempo para alcanzar la solución es una función polinómica
- Algoritmo INF: *iterative network-flow algorithm*
- Primero se consideran todos los tamaños de trama que cumplan las condiciones 2) y 3)
- INF trata de encontrar una planificación válida para cada tamaño de trama, comenzando por la mayor

Planificación - I

27

Gráfico de flujo de red

- Se tienen en cuenta los trabajos a planificar en F tramas (hiperperiodo)
 - El gráfico contiene vértices y flechas. Cada flecha tendrá asociada una capacidad que será un número no negativo
1. Existe un *vértice de trabajo* J_i que representa a cada trabajo J_i , para $i = 1, 2, \dots, N$.
 2. Existe un *vértice de trama* llamado j que representa a cada trama del hiperperiodo, para $j = 1, 2, \dots, F$.
 3. Existen dos vértices especiales llamados *fuelle* y *sumidero*.
 4. Existe una flecha (J_i, j) desde un vértice de trabajo J_i a un vértice de trama j , si el trabajo J_i puede ser planificado en la trama j . La capacidad de la flecha es el tamaño de trama f .
 5. Existe una flecha desde el vértice fuente a cada uno de los vértices de trabajo J_i , y su capacidad es el tiempo de ejecución C_i del trabajo.
 6. Existe una flecha desde cada vértice de trama j hacia el vértice sumidero, y la capacidad de esta flecha es f .

Planificación - I

28

Gráfico de flujo de red (cont.)

- El flujo de una flecha es un número no negativo que satisface las siguientes restricciones
 - No puede ser mayor que la capacidad del eje
 - Con la excepción de los vértices fuente y sumidero, la suma de los flujos de las flechas que entran en cada vértice debe ser igual a la suma de los flujos en las flechas que salen del mismo vértice
 - El flujo de los ejes (J_i, j) será la cantidad de tiempo que el trabajo J_i se ejecuta en la trama j . La suma de los flujos de todas las flechas que salen del vértice J_i debe C_i si el trabajo se puede planificar por completo. Por otro lado, la suma del flujo de todas las flechas que llegan a la trama j será como máximo f , que es la cantidad máximo de trabajo que se puede ejecutar en una trama
 - En las flechas que parten del vértice fuente y llegan a un vértice de trabajo, el flujo es la porción del trabajo planificado
 - En las flechas que parten de un vértice de trama y llegan al vértice sumidero, el flujo es la porción de trama rellena por trabajos

Gráfico de flujo de red (cont.)

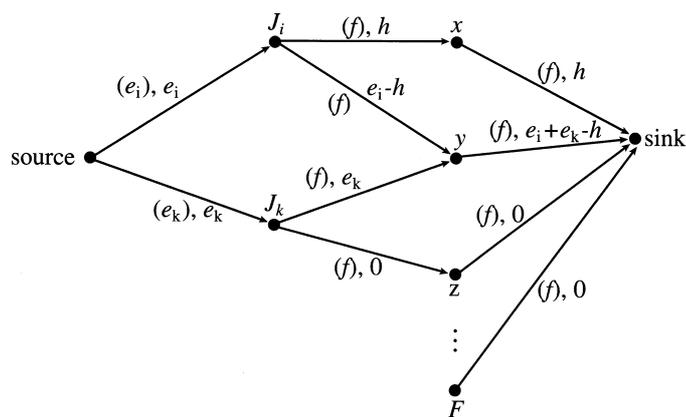


Gráfico de flujo de red (cont.)

- El **flujo total** del gráfico se define como la suma de los flujos que parten del vértice fuente
- Será igual a la suma de los flujos que llegan al vértice sumidero
- El **flujo máximo** será la suma de los tiempos de ejecución de todos los trabajos y se alcanzará cuando todos los trabajos puedan ser planificados en las tramas
- Al planificar los trabajos se tendrán en cuenta las siguientes restricciones
 - Un trabajo no se puede planificar en una trama que comienza antes de su instante de activación
 - La finalización de un trabajo en una trama no debe superar su límite de ejecución

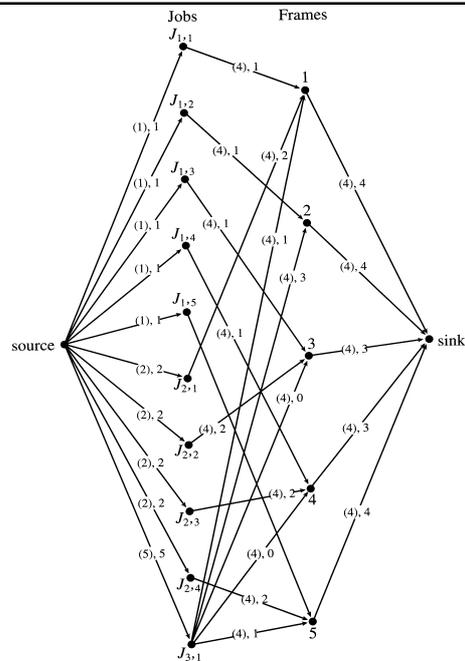
Gráfico de red de flujo

$$T_1 = (4, 1)$$

$$T_2 = (5, 2, 7)$$

$$T_3 = (20, 5)$$

$$f=4$$



Propiedades de los ejecutivos cíclicos

- Son conceptualmente sencillos y fáciles de implementar
- Son altamente deterministas
- Se pueden tener en cuenta las restricciones de precedencia entre tareas y las secciones no interrumpibles
- Los sistemas son relativamente fáciles de validar, verificar y certificar

Inconvenientes de los ejecutivos cíclicos

- Son poco flexibles y difíciles de mantener
- Las tareas esporádicas son difíciles de tratar
- El plan cíclico puede ser difícil de construir
 - Si los periodos son de distintos órdenes de magnitud
 - Puede ser necesario dividir una tarea en varios procedimientos
 - En el caso más general es NP-duro
- Se usan sobre todo en sistemas sencillos. En los sistemas complejos los inconvenientes superan a las ventajas

ERROR: syntaxerror
OFFENDING COMMAND: --nostringval--

STACK:

()
/Title
()
/Subject
(05.04.2005 23:25:41)
/ModDate
()
/Keywords
(PDFCreator Version 0.8.1)
/Creator
(05.04.2005 23:25:41)
/CreationDate
(Gomis)
/Author
-mark-