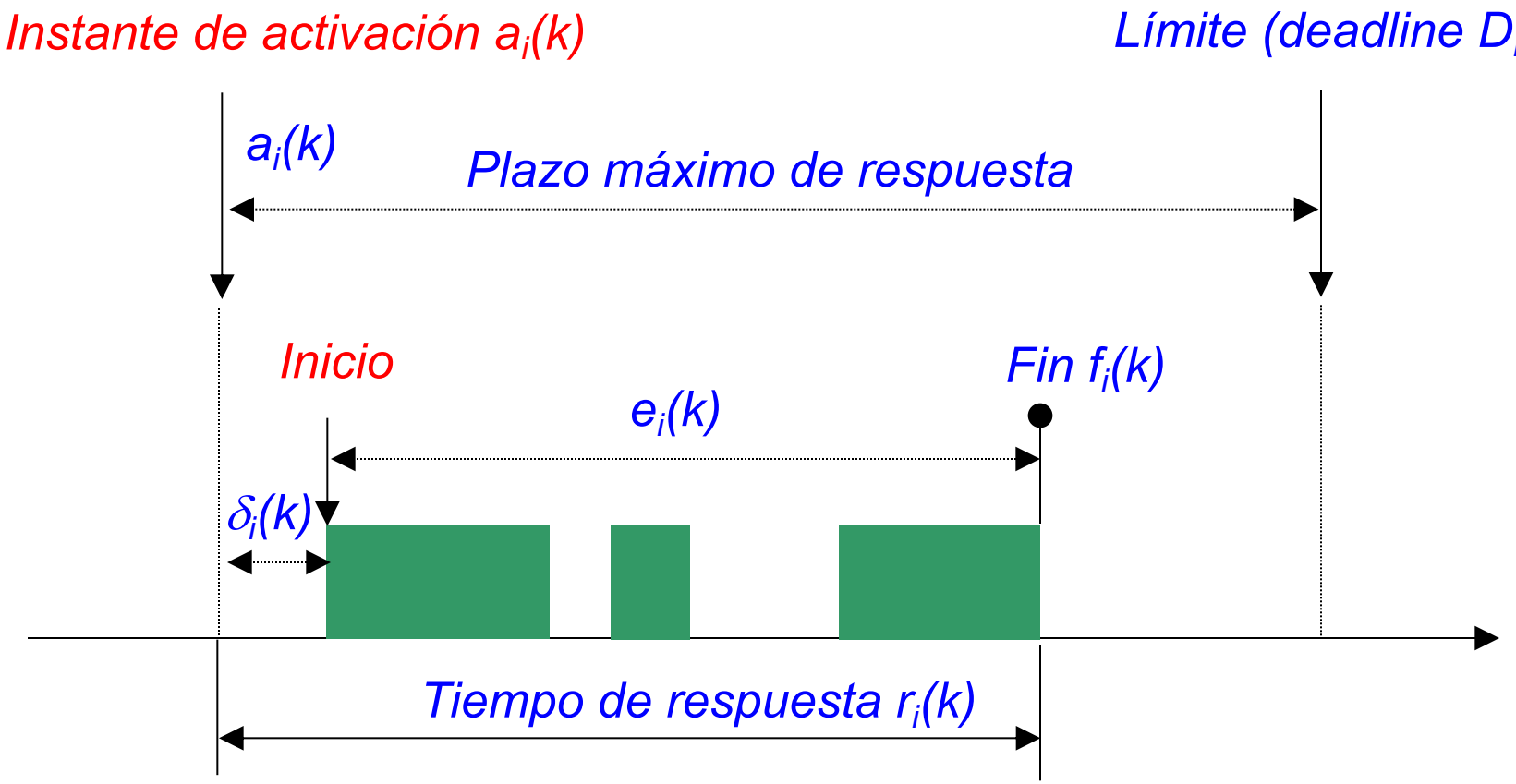
Planificación dirigida por tiempo

Dr. Óscar Rodríguez Polo
Dr. Sebastián Sánchez Prieto
Dr. Pablo Parra Espada

Parámetros temporales



Planificación

- Vamos a considerar una planificación *off-line* en sistemas con un único procesador
- Existen básicamente dos alternativas:
  - Planificación dirigida por tiempo: la decisión de qué trabajo se ejecuta en cada instante se lleva a cabo en instantes de tiempo concretos
  - Planificación dirigida por eventos o por prioridades: en cada momento se ejecuta la acción que tenga la prioridad más alta
- En sistemas de tiempo real estrictos se suele emplear una planificación dirigida por tiempo

Tipos de tareas

- Atendiendo a la criticidad de las tareas podemos hablar de:
  - Tareas de tiempo real críticas: nunca se puede superar su plazo máximo de respuesta
  - Tareas de tiempo real acríticas: se pueden superar ocasionalmente los plazos máximos de respuesta
  - Tareas interactivas: no tienen unos tiempos de respuesta estrictos, pero estos deben ser lo más pequeños posible
  - Tareas de fondo: sin restricciones
- Se podrían llevar a cabo otras clasificaciones atendiendo a otros criterios

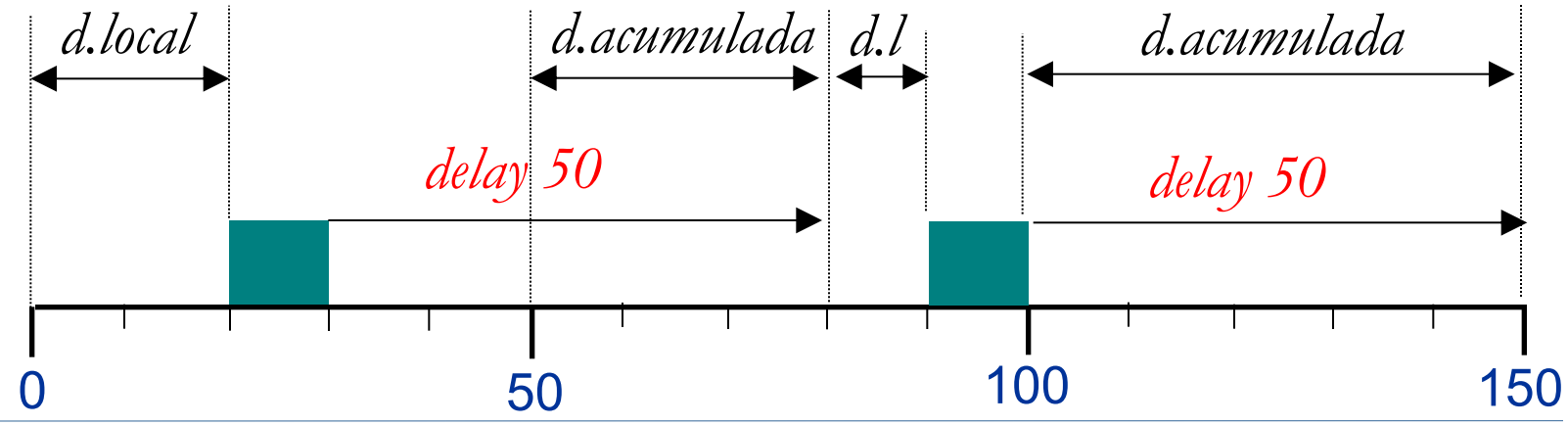
Planificación dirigida por tiempo

- Existe una tabla en la que se especifica qué acción debe llevarse a cabo en cada instante
- Esta tabla podría establecerse de un modo estático o dinámico
- El método más común empleado en planificación dirigida por tiempo son los ejecutivos cíclicos basados en retardos
- Estos retardos sirven para controlar el tiempo de activación de las tareas
- Normalmente se realizan por medio de una función del tipo **`delay()`**
- Los retardos se pueden emplear para realizar acciones periódicas:

```
do  {
      delay (TIEMPO);
      FuncionPeriodica();
} while (TRUE);
```

Problemas del esquema anterior

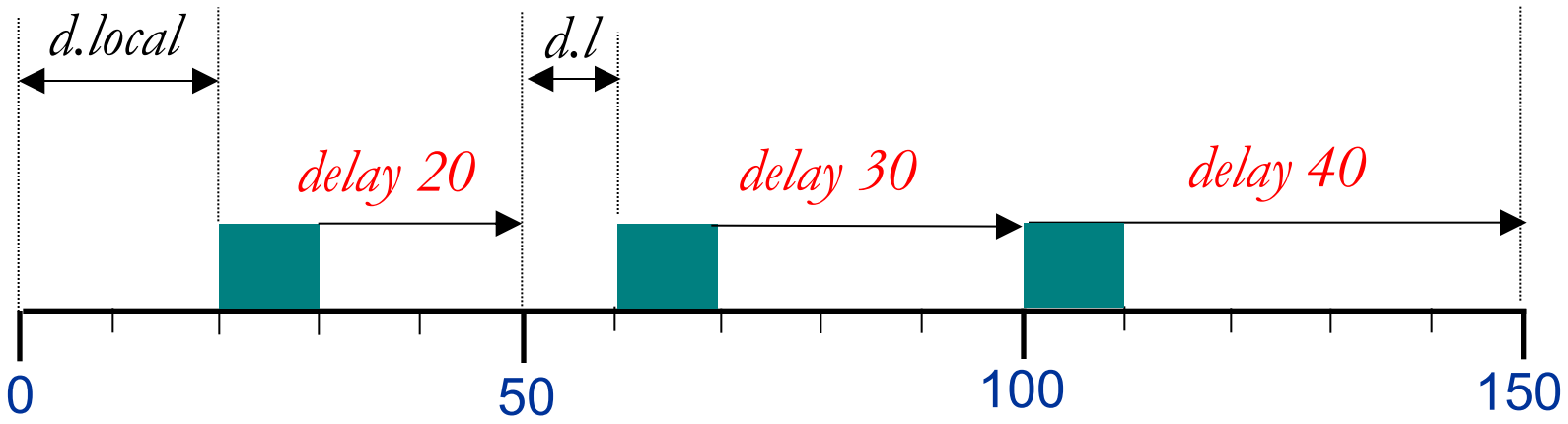
- No se tiene en cuenta la duración de la tarea periódica
- La ejecución de la acción puede retrasarse por existir otras tareas
- El resultado es una deriva acumulativa
- En el caso del ejemplo la tarea en el intervalo especificado debería haberse ejecutado tres veces y sólo se ejecuta dos:



Solución

- La deriva local no se puede eliminar, pero sí la acumulada:

```
Periodo = 0,050;
Siguiente = Time();
do {
    delay (Siguiente - Time());
    FuncionPeriódica();
    Siguiente = Siguiente + Periodo;
while (TRUE);
```



Planificación con ejecutivos cíclicos

- Es aplicable si todas las tareas son periódicas
- Tenemos un único programa secuencial
- Este método se ha venido utilizando mucho hasta ahora
- Un plan cíclico se divide en:
  - Ciclo principal: se repite con un periodo T_M
  - Ciclos secundarios: divisiones del ciclo principal sincronizadas con un periodo T_S

Ejemplo

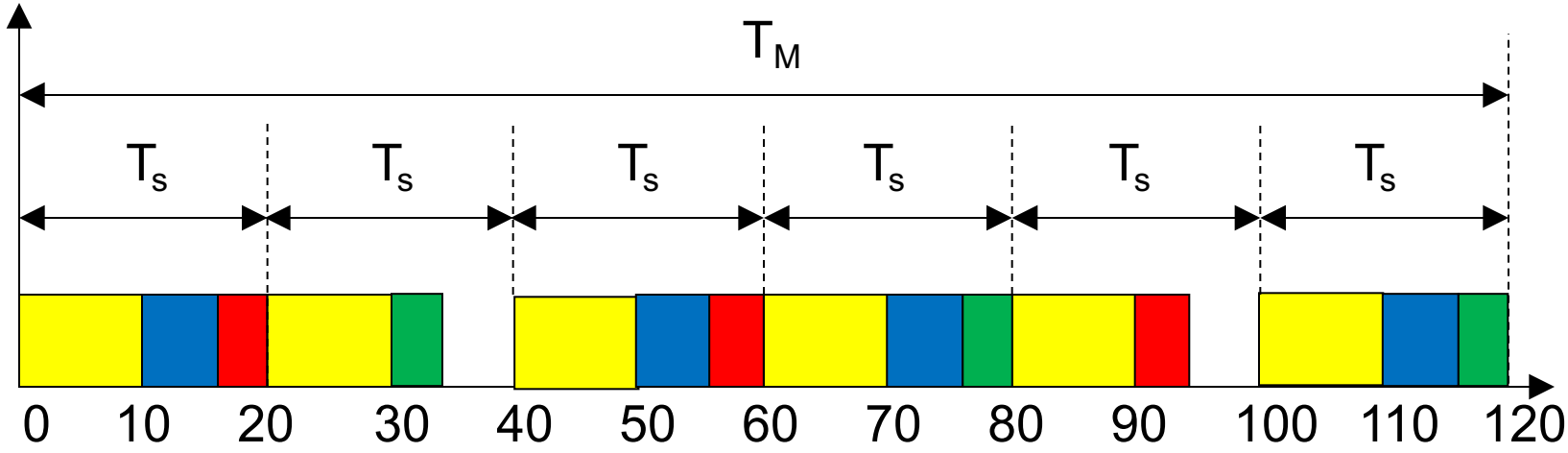
τ_1

τ_2

τ_{31}

τ_{32}

	C	D	T	U
t_1	10	20	20	0,5
t_2	6	30	30	0,2
t_3	8	40	40	0,2
				0,9



Programa cíclico

```
int Marco = 0;
Periodo = 20;
Siguiente = Time();
do {
      delay (Siguiente - Time());
      switch (marco) {
          case 0: T1; T2; T31;
          break;
          case 1: T1; T32;
          break;
          case 2: T1; T2; T31;
          break;
          case 3: T1; T2; T32;
          break;
          case 4: T1; T31;
          break;
          case 5: T1; T2; T32;
          break;
      }
      Marco = (Marco + 1) % 6;
      Siguiente = Siguiente + Periodo;
} while (TRUE);
```

Propiedades

- Ventajas
  - Eficiencia: no hay ni cambios de contexto ni gestión de procesos
  - Determinismo en los instantes de activación
  - No hace falta emplear mecanismos para evitar condiciones de carrera
- Inconvenientes
  - Rigidez y dificultad de mantenimiento
  - Bajo nivel de abstracción
  - Puede ocurrir que el número de subperiodos sea excesivamente grande