



INSTITUTO POLITÉCNICO NACIONAL



**CENTRO DE INVESTIGACIÓN EN CIENCIA
APLICADA Y TECNOLOGÍA AVANZADA**

**ANÁLISIS DE PLANIFICABILIDAD DE LOS ALGORITMO EDF
Y FIFO USANDO LA DISPERSIÓN DE LOS TIEMPOS DE
ARRIBO EN TAREAS DE TIEMPO REAL ESPORÁDICAS**

**TESIS QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN TECNOLOGÍA AVANZADA**

PRESENTA:

JOSÉ ALFREDO JIMÉNEZ BENÍTEZ

DIRECTOR DE TESIS: *DR. JOSÉ DE JESÚS MEDEL JUÁREZ*
CO- DIRECTOR DE TESIS: *DR. PEDRO GUEVARA LÓPEZ*

MÉXICO D. F., JULIO DE 2007



INSTITUTO POLITECNICO NACIONAL
SECRETARIA DE INVESTIGACION Y POSGRADO

ACTA DE REVISION DE TESIS

En la Ciudad de México, D.F. siendo las 11:00 horas del día 21 del mes de junio del 2007 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de CICATA - IPN para examinar la tesis de grado titulada:
ANALISIS DE PLANIFICABILIDAD DE LOS ALGORITMOS EDF Y FIFO USANDO LOS TIEMPOS DE ARRIBO EN TAREAS DE TIEMPO REAL ESPORADICAS.

Presentada por el alumno:

Jiménez
Apellido paterno

Benítez
materno

José Alfredo
nombre(s)

Con registro:

B	0	5	1	5	3	9
---	---	---	---	---	---	---

aspirante al grado de: Maestría en Tecnología Avanzada

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISION REVISORA

Director de tesis

Dr. José de Jesús Medel Juárez

Co-Director de tesis

Dr. Pedro Guevara López

Dr. José Antonio Calderón Arenas

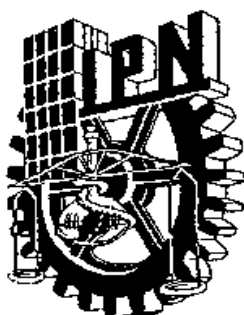
Dr. José Luis Fernández Muñoz

Dr. Ernesto Marín Moares

EL PRESIDENTE DEL COLEGIO

**CENTRO DE INVESTIGACION EN CIENCIA
APLICADA Y TECNOLOGIA AVANZADA
DEL IPN**

Dr. José Antonio Irán Díaz Góngora

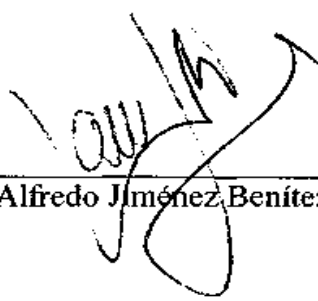


INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESION DE DERECHOS

En la Ciudad de México, D.F. el día 27 del mes de Junio del año 2007, el que suscribe José Alfredo Jiménez Benítez alumno del Programa de Maestría en Tecnología Avanzada con número de registro B051539, adscrito al CICATA LEGARIA, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del Dr. José de Jesús Medel Juárez y del Dr. Pedro Guevara López y cede los derechos del trabajo intitulado Análisis de Planificabilidad de los Algoritmos EDF y FIFO usando los Tiempos de Arribo en Tareas de Tiempo Real Esporádicas, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección alfredojimben@prodigy.net.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.



José Alfredo Jiménez Benítez

Este trabajo se lo dedico a Dios, a Eva y a mis Padres.

A Dios que me ha iluminado y guiado en este camino y me ha dado la fortaleza necesaria para seguir adelante a pesar de las dificultades y los sueños fallidos.

A mi esposa Eva que ha sido el amor, el cariño, el impulso, el soporte, la paciencia y la tolerancia en todo momento.

A mis Padres que siempre han estado presentes en mi mente y en mi corazón.

Agradecimientos.

A mi esposa Eva por su amor, su apoyo incondicional y su constancia en nuestro matrimonio ya que sin ella como pilar, la vida carecería de sentido.

Al Dr. José de Jesús Medel Juárez y al Dr. Pedro Guevara López por aceptarme en su vida y trabajo. Con sus opiniones y críticas me ha ayudado ha crecer como individuo y profesionista.

A mi familia por ser parte esencial del accionar diario y que sin cuestionar se han involucrado en mis problemas y tribulaciones.

A mis sinodales, el Dr. José Antonio Díaz Góngora, el Dr. José Antonio Calderón Arenas, el Dr. José Luis Fernández Muñoz, el Dr. Ernesto Marín Moares y al Dr. Miguel Ángel Aguilar Frutis, por tomarse el tiempo para revisar esta tesis y participar en mi crecimiento personal con sus aportes hacia este trabajo.

A Leticia Cardona y Laura Vega por su paciencia, su compromiso y su dedicación con nosotros los estudiantes.

A los compañeros de clase Daniel, Juan Carlos, Adolfo, Arturo, Armando, de quienes aprendí muchas cosas que ayudaron a ir formando criterios y perspectivas.

A Todas aquellas personas que pasaron por mi vida y contribuyeron con un granito de arena en mi pequeña historia que hasta este momento se ha construido.

ÍNDICE

RESUMEN	I
ABSTRACT	II
GLOSARIO	III
RELACIÓN DE FIGURAS	IV
RELACIÓN DE TABLAS	V

CAPÍTULO 1- INTRODUCCIÓN

1.1 Sistemas en Tiempo Real.	1
1.2 Clasificación de los Sistemas en Tiempo Real.	4
1.3 Sistemas Digitales como Sistemas en Tiempo Real.	4
1.4 Sistemas Operativos en Tiempo Real y Software de Simulación.	6
1.5 Tareas en Tiempo Real.	9
1.6 Planificadores en Tiempo Real.	13
1.7 Conclusiones del Capítulo 1.	16

CAPÍTULO 2- PLANTEAMIENTO DEL PROBLEMA

2.1 Definición del problema.	18
2.2 Objetivo general.	19
2.3 Objetivos específicos.	19
2.4 Justificación del proyecto.	19
2.5 Límites.	20
2.6 Alcances.	20

CAPÍTULO 3- CARACTERIZACIÓN Y REPRESENTACIÓN DE TAREAS EN TIEMPO REAL ESPORADICAS.

3.1 Introducción.	21
3.2 Caracterización de Tareas en Tiempo Real Esporádicas.	21
3.3 Características para el Simulador de Tareas en Tiempo Real Esporádicas.	22
3.4 Representación de los Tiempos de Arribo de las Tareas en Tiempo Real.	24
3.5 Resultados de la Simulación del Generador de Tareas.	25
3.6 Conclusiones del Capítulo 3.	29

CAPÍTULO 4- PLANIFICACIÓN EN TIEMPO REAL USANDO EDF PARA TTR ESPORÁDICAS.

4.1 Introducción.	30
4.2 Definiciones.	30
4.3 Las Tareas en Tiempo Real y la Planificación en Tiempo Real.	32
4.4 Las Tareas en Tiempo Real y el Planificador en Tiempo Real EDF.	36
4.5 Conclusiones del Capítulo 4	43

CAPÍTULO 5- SIMULACIÓN DEL ALGORITMO EDF CON TTR ESPORADICAS.

5.1 Introducción.	44
5.2 Algoritmo de Planificación EDF.	44
5.3 De la Teoría ala Práctica: El simulador del Algoritmo EDF.	45
5.4 Conclusiones del Capítulo 5	52

CAPÍTULO 6- PLANIFICACIÓN EN TIEMPO REAL USANDO EL ALGORITMO FIFO PARA TTR PERIÓDICAS Y ESPORADICAS.

6.1 Introducción.	53
6.2 Algoritmo de Planificación FIFO.	53

6.3 Conclusiones del Capítulo 6.	58
----------------------------------	----

CAPÍTULO 7- SIMULACIÓN DEL ALGORITMO FIFO CON TTR PERIÓDICAS Y ESPORÁDICAS.

7.1 Introducción.	59
7.2 De la Teoría a la Práctica.	59
7.3 Conclusiones del Capítulo 7.	66

CAPÍTULO 8- ANÁLISIS DE PLANIFICABILIDAD DE LOS ALGORITMOS EDF Y FIFO.

8.1 Introducción.	67
8.2 Factor de Utilización y Planificabilidad.	67
8.3 Resultados del Factor de Utilización y Planificabilidad con los Algoritmos Usados en los Capítulos 4, 5, 6 y 7.	68
8.4 Conclusiones del Capítulo 7.	72

CAPÍTULO 9- CONCLUSIONES.

9.1 Conclusiones.	73
-------------------	----

REFERENCIAS.	76
---------------------	----

ANEXOS.	83
----------------	----

RESUMEN.

Este trabajo inicia dando la introducción a los Sistemas en Tiempo Real (STR) para después introducir a los Sistemas Digitales (SD) como una generalización de los anteriores. Desde el punto de vista discreto se explican temas cruciales como las Tareas en Tiempo Real (TTR), su caracterización y su clasificación así como los parámetros que se utilizan en esta tesis. En el capítulo 4 llegamos al estudio de los Algoritmos de Planificación en Tiempo Real (APTR), los tipos que hay de estos y donde se ubican tanto el EDF como el FIFO. En los capítulos que siguen se definen características para poder llegar a los simuladores de los algoritmos EDF y FIFO. Por último, en base a la simulación de los algoritmos mencionados, se hacen cálculos para poder concluir con el Factor de Utilización y el nivel de Planificabilidad.

Palabras Clave: Sistemas en Tiempo Real, Tareas en Tiempo Real Periódicas, Tareas en Tiempo Real Esporádicas, Algoritmos de Planificación, EDF, FIFO, Simuladores, Factor de Utilización, Planificabilidad.

ABSTRACT.

This work initiates giving the introduction to the Real Time Systems later to introduce to the Digital Systems as a generalization of the previous ones. From the discreet point of view crucial topics are explained as the Tasks in Real Time, his characterization and his classification as well as the parameters that are in use in this thesis. In the chapter 4 it come to the study of the Algorithms of Planning in Real Time, the types that exist of these and where both the EDF and the FIFO are located. In the chapters that continue characteristics are defined to be able to come to the simulators of the algorithms EDF and FIFO. Finally, on the basis of the simulation of the mentioned algorithms, calculations are done to be able to conclude with the Factor of Utilization and Scheduling level.

Key words: Real Time Systems, sporadic Tasks in Real Time, Periodic Task in Real Time, Scheduling Algorithms, EDF, FIFO, Simulators, Factor of Utilization.

III

GLOSARIO.

Algoritmo: Nombre de cualquier procedimiento sistemático de cálculo con el que se halla el resultado deseado.

Banco de tareas: Conjunto de tareas que están en estado de listo y que se pueden ejecutar en el momento en el que se active su tiempo de arribo.

Correlación: Mide la relación que existe entre dos variables aleatorias, es decir, cómo afecta a una de las variables un cambio en la otra. La correlación es tanto mayor cuanto más estrecha es la relación entre las dos variables. Se habla de correlación positiva cuando al cambiar una de las variables en un sentido, y es negativa cuando las variables cambian en sentido opuestos, frecuentemente se

utiliza el llamado coeficiente de correlación que se define como $\rho := \frac{\text{Cov}(A,B)}{\sqrt{\text{Var}(A)\text{Var}(B)}}$, se cumple $-1 \leq \rho \leq 1$.

MATLAB™: Software usado para las simulaciones de las Tareas en Tiempo Real (TTR) y para la simulación de los algoritmos de planificación EDF y FIFO.

Planificador: Es un criterio predefinido que asigna procesos a los recursos en forma ordenada de tal forma que cada tarea es ejecutada hasta completarse. Un planificador puede hacer que un proceso pase del estado listo al estado ejecutándose y viceversa, dándole un lugar específico dentro de la cola de listos.

Proceso Estocástico: Es una familia de variables aleatorias indexadas $\{x(\omega_i)\}$ donde el conjunto $\{\omega_i\}$ pertenece al un conjunto universo Ω i.e.: $\{x(\omega), \omega \in \Omega\}$

Sistema Digital: Es cualquier dispositivo destinado a la generación, transmisión, procesamiento o almacenamiento de señales digitales.

Sistema Operativo: software de computadoras que administra los recursos del sistema de cómputo y ofrece una interfaz de programación al usuario para poder comunicarse con el hardware.

Sistema en Tiempo Real (STR): Es aquel sistema digital que obedece tres condiciones: a) Interacción con el mundo físico, b) Emisión de una respuesta

correcta de acuerdo a algún criterio preestablecido, c) Cumplimiento de las restricciones de tiempo del proceso físico con que interactúa.

Sistema Lineal (SL): Es aquel sistema que tiene una relación de entrada/salida y cumple las condiciones de causalidad y homogeneidad.

Soft: Se refiere al sistema de Tiempo Real en el que sus procesos pueden satisfacer sus restricciones de tiempo sólo en un cierto porcentaje de veces.

SOTR: Sistemas Operativos en Tiempo Real.

Tareas Concurrentes (TC): Son un conjunto de tareas que se ejecutan en un plazo máximo y en forma no secuencial en una computadora tomando recursos en instantes diferentes. No pueden existir tareas concurrentes sin un planificador predefinido.

Tarea en Tiempo Real (TTR): Es una entidad ejecutable de trabajo J_i que al menos es caracterizada por un tiempo de arribo y una restricción temporal. Está formada por un conjunto de instancias $j(k)$, tal que $J_i = \{j(k)\}$ con $i, k \in \mathbb{Z}^+$.

Tarea en Tiempo Real Aperiódica (TTRA): Es aquella tarea donde todas sus instancias tienen tiempos de interarribo activados en forma irregular, tiene un tiempo mínimo de interarribo y los tiempos de arribo absolutos crecen en forma aleatoria respecto a la aparición de las instancias.

Tarea en Tiempo Real Esporádica (TTRE): Es aquella tarea donde todas sus instancias tienen tiempos de interarribo activados en forma irregular relacionados a una función de repetición y los tiempos de arribo absolutos crecen en forma oscilante y acotada respecto a la aparición de las instancias.

Tarea en Tiempo Real Periódica (TTRP): Es aquella tarea donde todas sus instancias tienen tiempos de arribo relativo próximos a un período T y los tiempos de arribo absolutos crecen en forma monótonica respecto a la aparición de las instancias.

Tiempo de arribo o llegada: También llamado tiempo de arribo absoluto, es el tiempo con respecto al inicio del proceso en que la instancia de una tarea pasa a formar parte de la cola de listos. Es decir, es un tiempo absoluto en que la instancia queda formada hasta ser atendida.

Tiempo de cálculo: (Ver tiempo de ejecución).

Tiempo de ejecución: Es el tiempo que el procesador se encarga de ejecutar una instancia de una tarea.

Tiempo de ejecución o tiempo de cálculo: Es el tiempo en que el procesador se encarga de ejecutar una instancia de una tarea (o conjunto de instancias sin interrupción).

Tiempo de inicio o tiempo de liberación: Es el tiempo en que la instancia de una tarea inicia su ejecución.

Tiempo de interarribo de una TTR: El tiempo de interarribo o tiempo de arribo relativo de una instancia k está definido como el tiempo en que la instancia pide atención al procesador en relación con arribo absoluto de la instancia con índice $k-1$.

Tiempo Real: Conjunto de conceptos y definiciones que permiten establecer los parámetros donde puede actuar un sistema común y cumplir con restricciones temporales. Véase también: Sistemas en Tiempo Real.

Variable: Es un símbolo, tal como $X, Y, H, x \circ B$, que puede tomar un conjunto de valores en forma determinística o no determinística. Si la variable puede tomar un solo valor, se llama constante. Una variable que puede tomar cualquier valor entre dos valores dados, se dice que es una variable continua; en caso contrario se dice que la variable es discreta.

IV

RELACIÓN DE FIGURAS.

Figura 1.1 Representación de un Sistema en Tiempo Real.	3
Figura 1.2 Estados de una Tarea en Tiempo Real.	10
Figura 1.3 Parámetros de una Tarea en Tiempo Real.	12
Figura 1.4 Instancias de una Tarea en Tiempo Real.	12
Figura 2.1 Comportamiento interno de un planificador.	18
Figura 3.1 Parámetros de una Tarea en Tiempo Real.	22
Figura 3.2 Características Usadas en un Simulador de TTR.	23
Figura 3.3 Tarea en Tiempo Real que no Cumple con el Plazo Mínimo.	23
Figura 3.4 Conjunto de Operaciones Discretas de una Tarea en Tiempo Real.	24
Figura 3.5. Banco de Tareas en Tiempo Real con sus Parámetros.	25
Figura 3.6 Tiempos de arribo a lo largo un intervalo 0-1000 unidades de ejecución. Simulación 1.	26
Figura 3.7 Tiempos de arribo a lo largo un intervalo 0-1000 unidades de ejecución. Simulación 2.	26
Figura 3.8 Muestra de 10 simulaciones o de 10 intervalos.	27
Figura 3.9 Media de cada una de los intervalos en una grafica de barras.	27
Figura 3.10 100 intervalos de simulación en grafica de barras.	28
Figura 3.11 Simulación de 100 arribos con 100 repeticiones.	28
Figura 3.12. Simulación de 100 arribos con 100 repeticiones. Histograma.	29
Figura 4.1 Representación de un Banco de Tareas.	32
Figura 4.2 Características de Tareas en Tiempo Real en el Banco de Tareas.	33
Figura 4.3 Tiempo de ejecución C dividido en n operaciones.	33
Figura 4.4 Ejemplo de Parámetros de una Tarea en Tiempo Real.	34
Figura 4.5 Ejemplo 1 de Atención a una Tarea con Parámetros en Tiempo Real.	35
Figura 4.6 Ejemplo 2 de Atención a una Tarea con Parámetros en	35

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

Tiempo Real.	
Figura 4.7 Ejemplo 3 de Atención a una Tarea con Parámetros en Tiempo Real.	36
Figura 4.8 Ejemplo 4 de Atención a una Tarea con Parámetros en Tiempo Real.	36
Figura 4.9 Ejemplo 5 de Atención a una Tarea con Parámetros en Tiempo Real.	36
Figura 4.10. Tarea Ejecutada Hasta Antes del Tiempo 11.	37
Figura 4.11 Se resaltan las Condiciones que Rondan a la Segunda Tarea.	37
Figura 5.1 Diagrama de Flujo para el Algoritmo EDF Usado en la Simulación en MATLAB™.	46
Figura 5.2 Progreso de Prioridades Dinámicas de la Simulación de Dos Tareas.	47
Figura 5.3 Progreso de los Tiempos de Ejecución de la Simulación de Dos tareas.	47
Figura 6.1 La Tarea 1 Empieza a Ejecutarse Desde su Tiempo de Arribo.	55
Figura 6.2 La Tarea 2 Empieza a Ejecutarse Desde su Tiempo de Arribo.	55
Figura 6.3 Ejemplo 1 de Ejecución de las Tareas 1 y 2.	55
Figura 6.4 Desenvolvimiento de ambas tareas de acuerdo a sus plazos.	55
Figura 6.5 Evolución del Ejemplo 3.	57
Figura 6.6 Evolución del Ejemplo 4.	58
Figura 7.1 Diagrama de Flujo Para el Simulador del Algoritmo FIFO.	60
Figura 7.1 Cumplimiento de Plazos Para Varias Muestras en la Simulación.	61
Figura 8.1 Cálculos de los Parámetros de Muestras para las Simulaciones del Algoritmo EDF.	69
Figura 8.2 Resultados del Factor de Utilización en las Simulaciones para el Algoritmo EDF.	70
Figura 8.3 Cálculo de los Parámetros de Muestras para las Simulaciones del Algoritmo FIFO.	71
Figura 8.4 Resultados del Factor de Utilización en las Simulaciones para el Algoritmo FIFO	71

RELACIÓN DE TABLAS.

Tabla 1.1 Tipos de Sistemas Operativos.	7
Tabla 1.2 Sistemas Operativos en Tiempo Real.	8
Tabla 1.3 Tipos de Sistemas Operativos en Tiempo Real.	9
Tabla 1.4. Clasificación de los Sistemas Operativos.	15
Tabla 3.1 Descripción de los Parámetros de una Tarea en Tiempo Real.	25
Tabla 4.1 Cálculos y Secuencia de Tiempos que Presentan las Tareas de acuerdo a la Ecuación 4.6	40
Tabla 4.2. La Prioridad de la Tarea 1 no se Toma debido al Plazo Mínimo.	41
Tabla 4.3. Secuencia de Tiempos Tomando en Cuenta el Cambio de Contexto.	42
Tabla 4.4. Cálculos y Secuencia de Tiempos que Presentan las Tareas de acuerdo a la Ecuación 4.7.	43
Tabla 5.1 Conjunto de 34 Ejecuciones con Diferentes Datos Aleatorios.	49
Tabla 5.2 Pruebas con el Nuevo Plazo Máximo.	52
Tabla 5.3 Nuevos Parámetro para una Reciente Simulación.	52
Tabla 5.4 Prueba de Plazo Máximo Requerido.	52
Tabla 6.1 Parámetros para la Tarea 1 Usados en la Simulación FIFO.	54
Tabla 6.2 Parámetros para la Tarea 2 Usados en la Simulación FIFO.	54
Tabla 6.3 Parámetros de la Tarea 1 Para Ejemplo 2.	56
Tabla 6.4 Parámetros de la Tarea 2 Para Ejemplo 2.	56
Tabla 6.5 Parámetros de la Tarea 1 Para Ejemplo 3.	57
Tabla 6.6 Parámetros de la Tarea 2 Para Ejemplo 3.	57
Tabla 6.7 Parámetros de la Tarea 1 Para Ejemplo 4.	57
Tabla 6.8 Parámetros de la Tarea 2 Para Ejemplo 4.	57
Tabla 6.9 Modificación en la Tarea 2 Para Ejemplo 4.	58
Tabla 7.1 Parámetros para la Simulación del Algoritmo FIFO.	59

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

Tabla 7.1a Resultados de la Simulación FIFO con Diferentes Entradas.	62
Tabla 7.1b Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).	63
Tabla 7.1c Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).	64
Tabla 7.1d Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).	65
Tabla 7.1e Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).	66

CAPÍTULO 1

INTRODUCCIÓN

En este capítulo se da un panorama general de los temas que aborda este trabajo de tesis de manera que se puedan introducir sus características principales, así también se conforma un trabajo de investigación de los STR como también de algunos subtemas que de este se derivan así como de otros que son complementarios para la estructuración completa de este trabajo.

1.1 Sistemas en Tiempo Real.

Hoy en día muchos sistemas de control han tenido la necesidad de introducir nuevas y mejores tecnologías que puedan cumplir con restricciones temporales y del medio donde se encuentran. Aquí es donde entran los STR, ya que estos tienen la capacidad de ser la herramienta adecuada para resolver los problemas que se plantean en el mundo real. Como ejemplos de esto que se acaba de decir existen en México algunas empresas que están utilizando los STR para resolver sus problemas de control como son Compañía de Luz y Fuerza del Centro (CLYFC), Comisión Federal de Electricidad (CFE), Sistema de Transporte Colectivo Metro (STCM), este último en particular maneja en sus computadores el Sistema Operativo (SO) en TR [Lib1] llamado QNX [Int1] el cual lo utiliza para monitorear el flujo de los trenes así como para el control de arribo a cada una de las estaciones [Lib1].

En México son pocas las empresas que utilizan este tipo de tecnología, sin embargo en el mundo muchas son las empresas que la están implementando cada vez más en sus sistemas. Por mencionar algunas están las industrias automotrices como Acura, Audi, Chrysler, Hyundai y otras más [Art2] [Int1] [Lib3]. Estas empresas automotrices están usando los STR para resolver problemas de navegación, telemática, audio y entretenimiento. Otra aplicación que las industrias ven con buenos ojos es donde los STR trabajan 24 horas al día los 365 días del año [Int1] [Lib2] [Lib3]. Estas empresas son las que se dedican principalmente a la manufactura de diversos productos; por dar algún ejemplo se tienen las que fabrican concreto, pintura en aerosol, energía eléctrica derivada de reactores nucleares, fotografía, herramientas de ultra precisión en la realización de diamantes, etc. En la rama de la medicina [Int1] [Lib2] se tienen sistemas personales de hemodiálisis, equipos de monitoreo electrocardiográfico de estrés, equipo de monitoreo del sistema hemodinámica, láser para cirugía ocular, fluoroscopia digital, radiografía, monitoreo de anestesia, etc. En las telecomunicaciones y redes se tiene un avance cada vez más importante ya que

los fabricantes de estas tecnologías están utilizando los STR cada vez más y una de las razones es la capacidad de el multiprocesamiento que algunos SO pueden manejar como lo son el QNX [Int1] o como el RT-Linux [Int2]. En los que concierne a aquellos sistemas que se encargan de la defensa de las naciones, la tecnología de los STR llevan un camino muy largo en aplicaciones [Int1] [Lib2] [Lib3] como asistencia no personal en el control de aviones [Art2], sensores acústicos para el movimiento de tanques, vehículos autónomos que viajan dentro del agua, sistemas guía para la defensa antitanques, sistemas de aterrizaje, etc. Así como estas hay muchas más, con las cuales podemos ver las interminables y muy variadas aplicaciones que tiene las tecnologías en tiempo real, tecnologías que tienen ganadas muchas batallas pero faltan muchas más que ganar.

Dejando un poco de lado las aplicaciones de los STR y entrando más en materia sobre la teoría del TR, la primera pregunta que se aborda es ¿Qué es un Sistema en Tiempo Real? Empezando con algo sencillo como esto: son aquellos sistemas donde el tiempo es el parámetro que hace la diferencia con los demás sistemas [Lib1] [Lib2]. Sin embargo existen muchas más definiciones que dan un panorama más amplio y complejo de esta área.

En [Lib2] se dice que un sistema en tiempo real es aquel que debe reaccionar dentro de restricciones de tiempo preciso a eventos en el medio ambiente. Como consecuencia, el correcto desempeño de estos sistemas depende de dos cosas: primero del valor computado y segundo el tiempo en el cual los resultados son producidos [Lib2] [Lib5]. Es importante mencionar que un sistema en tiempo real no es aquel que actúa de manera rápida de acuerdo a un evento.

Se tiene en cuenta que los procesos en el mundo real no siempre son rápidos o lentos, por el contrario estos están sujetos a las restricciones que cada sistema en particular tiene. En el supuesto caso de un proceso del mundo real: El proceso de gestación, desarrollo y nacimiento de un ser humano, si estamos hablando del ser humano este proceso tarda aproximadamente 9 meses, claro esta que hay límites tanto inferiores como superiores que presentan los plazos mínimos como máximos, estos límites pueden ser desde los 6 o 7 meses hasta los 9 ½ meses. Este por ejemplo no es un sistema rápido, es un STR que cumple con restricciones temporales. Si se tuviera a otro ser del reino animal (Lobos, águilas, etc.) este tendría una diferentes restricciones temporales pero también cumple con ser de tiempo real y por supuesto que no un sistema rápido.

En [Lib4] se dice que un STR es aquel que interactúa con un entorno de dinámica conocida, es decir se debe conocer bien la variable fundamental del medio donde se esta desarrollando el sistema la cual esta en relación con sus entradas, salidas y restricciones temporales todo esto para darle un correcto funcionamiento controlable, estable y alcanzable. De nuevo en esta referencia se hace hincapié en la diferencia que hay en los sistemas rápidos y los STR además se hace también la diferencia con los sistemas en línea los cuales tienen como características principales de estar encendido siempre, disponible, generalmente

estar conectados con una red de computadoras y la capacidad de atención al usuario depende del hardware.

Es importante hacer notar que los STR, son sistemas que están en el mundo real, es decir todos los procesos que podamos encontrar en la naturaleza son en TR. El punto fundamental es de poder imitar ese comportamiento en sistemas que controlan procesos inventados, dirigidos, manipulados por el ser humano como la construcción de un automóvil, o el sonido que su reproductor de audio emite, o el monitoreo de los frenos o las carreteras donde circula este automóvil. Por eso mismo los STR deben de tener una plataforma donde puedan desempeñarse adecuadamente, esta plataforma son los sistemas digitales y en particular los sistemas digitales computacionales (aunque no exclusivamente).

En [Lib1], el cual es antecedente directo de [Lib4], se habla de los tiempos de respuesta que debe de dar el sistema. Estos tiempos se toman como significativos, claro esta de acuerdo al proceso al que se le entrega la respuesta es significativo este tiempo. Aquí se caracteriza con tres premisas que debe cumplir el STR: a) debe de estar en contacto con el mundo físico, b) emisión de respuestas correctas, c) obtención de respuestas dentro de los intervalos de tiempo establecidos. En la siguiente figura [Figura 1.1] vemos como se podría representar lo anteriormente dicho.

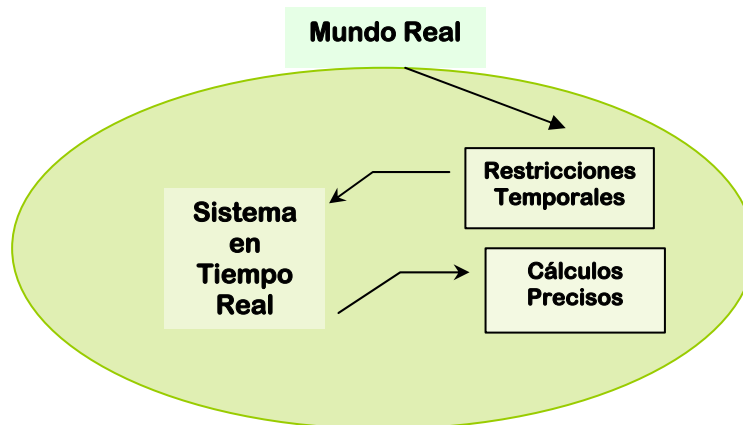


Figura 1.1 Representación de un Sistema en Tiempo Real.

En [Art2] se resalta como debe estar constituido un STR, el cual debe estar dividido en el sistema controlado y el sistema a controlar además como el medio ambiente donde estén ambos es fundamental para un buen control del sistema. El sistema controlador debe tener información disponible en cualquier momento, para esto debe haber un monitoreo necesario de las actividades que realiza el sistema controlado. Además de esto se menciona que los STR deben conservar las siguientes características: a) Granularidad del plazo máximo o deadline, donde si la diferencia entre el tiempo de ejecución completada y el tiempo de activación es corto entonces la granularidad de deadline es pequeña. Además se habla de un relajamiento de las tareas pequeño (relajamiento de las tareas = deadline – tiempo de procesamiento) cuando la granularidad es grande. b) Exactitud en el deadline,

donde en algunos casos por hacer que todos los sistemas cumplan con el plazo se sobre diseñan perdiendo claridad y simplicidad. c) Fiabilidad. d) Tamaño del sistema y grado de coordinación. e) y el tan renombrado medio ambiente donde se desarrolla todo lo anterior.

1.2 Clasificación de los Sistemas en Tiempo Real.

Una de las clasificaciones comunes de los STR es aquella donde en la atención a las tareas sucede una de dos situaciones [Art1]: donde se garantiza que en un intervalo de tiempo, se logrará realizar la tarea entonces se dice que el STR es “hard”, sin embargo si solo se toma la distribución estadística de los tiempos de respuesta como aceptable se dice que el STR es “soft”. También es importante decir que en los STR “soft” [Lib2] la perdida de un plazo para una tarea no causa gran daño en el sistema ya que este puede regularse en tiempos futuros.

Otra clasificación más amplia dice lo siguiente [Lib1] [Lib5] [Inf1]: los STR pueden ser de tres tipos:

- Críticos: aquellos donde se cumple la condición de ser “hard” como en [Art1]. El hecho de que no se llegara a cumplir la condición de los tiempos de terminación de cada tarea podría causar desastres al sistema completo [Lib2].
- No Críticos: estos permiten tener de manera ocasional la perdida de un plazo de alguna tarea. Es relevante hacer notar que debido a esta perdida el sistema se degrada y puede perder información o descalibrarse pero no será desastroso para el sistema en si.
- Flexibles: también llamados “firm”, este sistema tiene la particularidad de que si algún plazo se pierde el sistema ya no sirve o mejor dicho, deja de funcionar parcialmente o en su totalidad, aunque no causar gran daño al sistema en si.

1.3 Sistemas Digitales como STR.

Desde hace ya algunas décadas las tecnologías más avanzadas de cualquier rama de la ciencia han estado estrechamente ligadas con la electrónica. De la electrónica se divide dos rumbos, la analógica, de la que podemos decir que es el principio de la misma y de sus aplicaciones, y la digital que, en combinación con la analógica forman sistemas analógicas-digitales que dan como resultado los componentes que conforman las máquinas digitales que establecen la base de las múltiples aplicaciones que cotidianamente usamos hoy en día. Estos componentes digitales que van desde simples compuertas hasta los mas complejos dispositivos de procesamiento de bits conllevan una relación muy amplia y todos juntos pueden controlar muchos procesos que van desde los más sencillos hasta los que requieren de alto poder de procesamiento. A todo este conjunto de componentes electrónicos digitales les llamamos Sistemas Digitales.

Las aplicaciones de los sistemas digitales (SD) son tantas que aún tratando de describir un panorama de estas sería tedioso y más que esto no es el caso de estudio de este trabajo. Sin embargo, podemos decir algunas de estas solo para no dejar duda de lo dicho con prelación.

Una aplicación de los SD es la de cubrir la necesidad de la comunicación. Los sistemas digitales de comunicación [Lib6] [Lib11] abarcan bastantes de las aplicaciones que de comunicación de cualquier índole se refiere: sistemas de telecomunicación, sistemas telefónicos, sistemas satelitales, sistemas de televisión, redes de computadoras entre otros.

Otra aplicación de la electrónica que ha dado pauta a trabajos cada vez más elaborados por la necesidad de procesamiento de información y que antes hubiera sido muy complejo o casi imposible de realizar es la de la computación [Lib12]. Esta se puede ver desde dos puntos de vista [Lib10]: la de un programador de computadoras y la de un especialista en el hardware de computadoras como lo son los ingenieros en electrónica. Tomando el primer punto de vista, el de un programador, este tiene en la computadora (un SD), como una herramienta de trabajo para crear sistemas que procesan información ya sea de manera automática o con la interacción de un usuario, pero más allá de esto se tiene una herramienta que ayuda en la simulación de otros sistemas en donde sería complicado y costoso hacer directamente sobre ellos mismos las pruebas. Por lo mismo los sistemas digitales como las computadoras son un excelente medio para representar el comportamiento de distintos ambientes de trabajo, restricciones y variables que existen en el medio verdadero. Esto ayuda a guiar a los diseñadores de sistemas del mundo real sobre cual camino sería el mejor para el diseño.

Desde el punto de vista de un diseñador de hardware, una computadora es un conjunto de componentes que relacionados pueden hacer operaciones con bits y a su vez resolver problemas comunes de procesamiento de información que forman la vinculación de esos bits [Lib10]. También es importante decir bajo este punto de vista que una computadora puede estar conformada por uno de dos dispositivos fundamentales para el procesamiento de información; estos dos son los microprocesadores y los microcontroladores. En los microprocesadores se puede instalar un sistema operativo que comandará a los demás componentes de hardware para el manejo adecuado de la información, esta ventaja a su vez permite tener un SO dedicado a procesos en TR (el cual es nuestra área de interés) para primero poder conocer el comportamiento y los parámetros de un STR y segundo poder hacer simulaciones de dispositivos que se requiere que trabajen en TR.

En la actualidad existen computadoras con microprocesadores muy sofisticados y poderosos [Lib13] [Lib14], sin embargo el precio que se tiene que pagar en sistemas que no requieren esta capacidad de procesamiento es alto en costo, en espacio, en programación y en simulación de este tipo de sistemas. La solución

que se puede considerar como inmediata es la de usar como herramienta de procesamiento y de control un microcontrolador [Lib15] [Lib16]. Este es un descendiente directo de los microprocesadores (Sistema CISC – Complex Instruction Set Computer) con la diferencia de que su complejidad es mucho menor ya que cumple con las características de un sistema basado en RISC (Reduce Instruction Set Computer) lo que se traduce en menor tamaño de implementación, facilidad en la programación y fácil manejo de interfaces. Esto también implica que no requiere de un SO para funcionar.

Aunque se podría decir que en un microcontrolador se tiene la desventaja de no poder instalarse un sistema operativo tan sofisticado como en los microprocesadores y que por eso no se pueden hacer simulaciones y conocer los SO especializados en TR, no quiere decir que no se pueda implantar un STR en estos. Por el contrario para sistemas pequeños con la necesidad de ser controlados por un STR, los parámetros, las directivas y las restricciones de tiempo que se necesitan para este tipo de sistemas perfectamente se podrían acoplar a los microcontroladores por la facilidad que tienen estos de programarse de acuerdo a la necesidad para la cual son solicitados.

1.4 Sistemas Operativos en Tiempo Real y Software de Simulación.

Un sistema operativo es aquel que es capaz de administrar el hardware de un sistema digital de computo [Lib17], el cual esta constituido por uno o más procesadores, memoria principal, teclado, pantalla, interfaces de red y otros dispositivos de entrada y salida. Los SO han evolucionado a la par de las arquitecturas de los SD y las necesidades de las diferentes aplicaciones han generado que los desarrolladores de SO trabajen en diversos tipos para esas diferentes necesidades. Por lo mismo, hoy en día se tienen una gran variedad de SO como se muestra en la Tabla 1.1.

Tipo de SO	Aplicación
Mainframe	Se encuentran en centros corporativos. Cantidad de discos, dispositivos de E/S y gigabytes de datos muy grande.
Servidor	Se da servicio a usuarios en una red como el de impresión, Web, de archivos. Estos pueden estar instalados una computadora personal o en un mainframe.
Multiprocesador	Ayuda con grande capacidad de procesamiento al controlar y administrar computadoras conectadas en paralelo o tener multiprocesadores.
Computadora personal	Buena interfaz a un solo usuario. Procesamiento de texto, hojas de cálculo, acceso a Internet.
Integrados	Sistemas pequeños como PDA o Poquets,

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

	además de televisores, hornos de microondas.
Tarjeta Inteligente	Sistema primitivo usado en chips de tarjetas de pagos o credito.
Tiempo Real	Procesamiento de información donde se deben de considerar los plazos de tiempo de ejecución de cada proceso o de un conjunto de procesos.

Tabla 1.1 Tipos de Sistemas Operativos.

Un SO es indispensable para el manejo adecuado de un SD, en las computadoras es imprescindible tener uno y debido a la complejidad del sistema es muy complicado hacer uno especial para alguna solicitud particular. En los SD basados en microcontroladores la flexibilidad de crear un SO es mayor permitiendo de esta manera que los requerimientos de uso de hardware sean específicos a la aplicación. Se puede considerar como desventaja que para cada aplicación se tiene que estudiar una forma de SO diferente ya que el hardware a controlar cambia considerablemente para cada situación. Sin embargo el manejo sencillo y nada complicado de los microcontroladores permite con prontitud llegar a las metas que conciernen a control de hardware por parte del SO.

El rubro de SO que en este trabajo es de interés son aquellos que están dedicados al Tiempo Real. Para esta área existen a la fecha ya muchos SO que pueden ser considerados de TR ya que cumplen con las especificaciones de tiempo estipuladas en la teoría de TR. Estos pueden ser [Lib1] [Lib4] [Lib17] VxWorks, QNX, RTLinux, o más específicamente los que se muestran en la Tabla 1.2. La mayoría de los Sistemas Operativos en Tiempo Real SOTR que se muestra en la tabla están destinados a controlar procesos específicos de algunas empresas o en proyectos de investigación [Int5] [Int6] [Int7] [Int8] [Int9] [Int10] [Int11] [Int12] [Int13] [Int14] [Int15], por dar un ejemplo en los aviones nuevos de Boeing de la serie 787 se intentando usar el SOTR VxWorks.

Open source	Proprietario
Ecos	BeOS
Fiasco (L4 clone)	ChorusOS
Free RTOS	DMERT
Phoenix-RTOS	embOS (Segger)
Nut/OS	INTEGRITY
Prex	ITRON
RTAI	LynxOS
RTEMS	MERT
RT Linux	MicroC/OS-II
SHaRK	MQX RTOS
TRON Project	Nucleus RTOS Nucleus
Xenomai	OS-9
	OSE
	OSEK/VDX
	OSEKtime

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

	PDOS
	Phar Lap ETS
	PikeOS
	pSOS
	QNX
	RMX
	RSX-11
	RT-11
	RTOS-UH
	RTXC
	SINTRAN III
	Symbian OS
	ThreadX
	VRTX
	VxWorks
	Windows CE
	µnOS
	UNIX-RTR
	REX
	HP-1000/RTE

Tabla 1.2 Sistemas Operativos en Tiempo Real.

Los SOTR se pueden clasificar en propietarios, extensiones para SO comerciales y de investigación [Lib1] [Lib2] y sus características se muestran en la Tabla 1.3.

Tipo de SOTR	Características
Propietarios	• Aplicaciones industriales. • Cambio de contexto en las tareas. • Microkernel. • Respuesta rápida a interrupciones. • Memoria Real, no hay memoria virtual. • Archivos en disco en forma secuencial para búsquedas más rápidas. • Reloj en TR. • Asignación de prioridades a Tareas. • Manejo de tiempo en tareas.
Extensiones para SO comerciales	• Más lentos que los propietarios. • Optimización. • Asignación de recursos por demanda. • Planificación independiente del procesador que puede causar bloqueos.
Investigación	• Descubrir modelos de procesos de TR. • Nuevos métodos de sincronización. • Nuevas soluciones para facilitar el análisis del tiempo entre el sistema

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

	inicial y su modificación. • Desarrollar soporte para tolerancia a fallos. • Proporcionar soporte a sistemas distribuidos.
--	---

Tabla 1.3 Tipos de Sistemas Operativos en Tiempo Real.

Para las simulaciones con alto nivel de procesamiento y fácil acceso basta con el SO Windows XP Home donde se puede instalar de manera fácil paqueterías como MATLAB 7.0, en donde se pueden realizar operaciones de simulación y que además ha servido de ayuda para el procesamiento matemático y de programación de sistemas de muchas índoles [Lib18], la segunda es QNX 6.0 el cual es un SOTR y donde se más halla de hacer una simulación, se usara para ver el comportamiento en TR.

Los SO en general manejan conceptos en común [Lib10] [Lib17] como lo son procesos o tareas, planificación, administración de la memoria, archivos, llamadas al sistema, etc. Sin embargo cada sistema en particular maneja de manera particular algunos conceptos que otros SO no lo hacen, aunque no quiere decir que no los contengan como por ejemplo en UNIX se habla del manejo de señales y semáforos, así también en Linux junto con el de programación de multihilos, tuberías y memoria compartida [Lib20] [Man1] [Man2]. En Windows se maneja mucho el concepto de la API o Interfaz de Programación de Aplicaciones [Lib21].

En las siguientes secciones se detalla un poco sobre dos conceptos relevantes para el trabajo que en este escrito se presenta. Además en capítulos posteriores se seguirá haciendo hincapié en la diversidad del manejo de los mismos.

En lo que respecta a las herramientas de simulación, MATLAB es una de las tecnologías de software que permiten resolver problemas de ingeniería y ciencia en muchas ramas de las mismas [Lib18]. Por lo general se usa para resolver aquellos problemas que requieren de mucho cálculo matemático o para resolver modelos de sistemas físicos, ya sea que estén diseñados o que estén en proceso de investigación. Estos últimos son los que llamamos simulaciones de sistemas.

Debido al fácil manejo y la interfaz accesible al programador, MATLAB ha logrado que sus usos y aplicaciones se extiendan a muchas ramas de la tecnología actual como la electrónica, las comunicaciones, la informática, la biotecnología, la medicina y la farmacéutica entre otras [Int16].

1.5 Tareas en Tiempo Real.

Una parte de estudio de los SO son los procesos o tareas y básicamente se puede decir que son programas en ejecución [Lib17]. Cuando hablamos de un proceso

casi siempre tendremos relación con partes internas de hardware y software como son: el contador de programa, registros y variables.

Un proceso puede existir debido a diferentes circunstancias como son la iniciación de un SO, aquí se requiere que algunos procesos específicos se activen ya sea antes, durante o después del arranque de un SO, es decir son procesos necesarios para que el SO funcione correctamente. Otra instancia para crear procesos es cuando se hace una llamada al sistema como requerimiento de otro proceso en ejecución, esto es, cuando un proceso crea otro proceso. En UNIX, por ejemplo, solo se puede hacer este procedimiento con la llamada al sistema `fork()` [Lib17] [Lib20] [Lib22]. Cuando un usuario pide atención entonces esta creando un proceso que le atenderá de acuerdo a sus solicitudes. Por último cuando se inicia un trabajo por lotes se crea un proceso.

Así como esta la necesidad de crear procesos y que estos procesos ejecuten algo, ya sea una o más tareas, también existe la necesidad de que estos procesos terminen en algún momento. Las razones por las que puede terminar un proceso son: terminación normal voluntaria, terminación voluntaria por error, terminación fatal por un error fatal, terminado involuntario por otro proceso. Cuando un proceso realiza su trabajo de principio a fin se dice que tuvo una terminación normal, esta la razón por la cual la mayoría de los procesos terminan.

Existen ciertos estados que determinan la actividad de un proceso en cierto tiempo. Estos estados son: En estado de ejecución, que es cuando el proceso esta usando recursos del sistema o recursos de CPU. En estado de listo, cuando el proceso esta listo pero no se ejecuta porque el SO decidió mediante el planificador (El cual se verá en la siguiente sección) que otro proceso se ejecutara. El estado de bloqueado que es cuando el proceso no puede continuar ya que necesita información proveniente de otro proceso para poderse ejecutar. En estado de terminado, que más que un estado es un aviso de que el proceso termino de alguna manera. En la Figura 1.2 se muestra como los procesos pasan de un estado a otro de acuerdo al estado anterior.

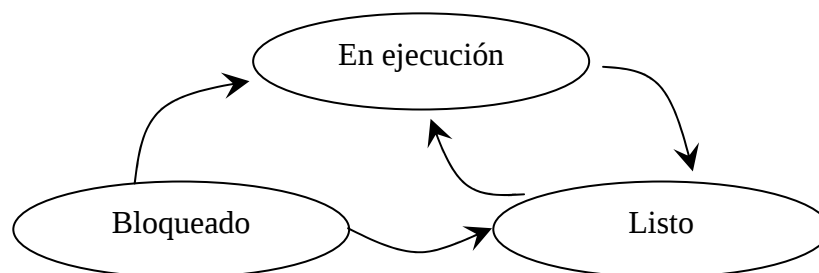


Figura 1.2 Estados de una Tarea en Tiempo Real.

Como ya se menciona, cuando se bloquea un proceso lo mas seguro es que se necesite información de entrada por parte de otro proceso. Para que el proceso

que manda la información y para el que la recibe se requiere de un método de comunicación donde ambos sepan que existe información común a los dos. Esta comunicación entre los procesos se forma con el IPC o comunicación entre procesos [Int3] [Lib17] [Lib1]. Esta comunicación se basa en distintos mecanismos como lo pueden ser las señales, las tuberías, la memoria compartida, semáforos.

En los STR y en los SOTR se les llama a los procesos como “tareas”. Una Tarea en Tiempo Real (TTR) se puede definir como un conjunto finito de actividades básicas que se ejecutan potencialmente en una computadora [Lib1] [Art7]. También podemos considerar que las TTR son aquellas que deben cumplirse en un plazo de tiempo [Tes1]. Las actividades que debe realizar una tarea en tiempo real se denominan instancias, entonces una TTR es un conjunto finito de instancias que se ejecutan hasta completar la tarea [Tes1]. Otra forma de ver una TTR es cuando esta caracterizada por su tiempo de arribo y la restricción temporal [Art6].

Las TTR se pueden caracterizar con los siguientes parámetros [Art11]:

- Tiempo de arribo o llegada $l(k)_i$
- Tiempo de ejecución $C(k)_i$
- Plazo relativo mínimo $D(k)_{imin}$
- Plazo relativo máximo $D(k)_{imax}$
- Tiempo de liberación o de Tiempo de inicio $s(k)_i$
- Tiempo de finalizado $f(k)_i$
- Tiempo de retardo $L(k)_i$
- Tiempo prematuro $P(k)_i$
- Plazo de convergencia d_i
- Periodo de muestreo $T(k)_i$
- Instancia J_i

Los parámetros que se han mencionado se ilustran en la Figura 1.3.

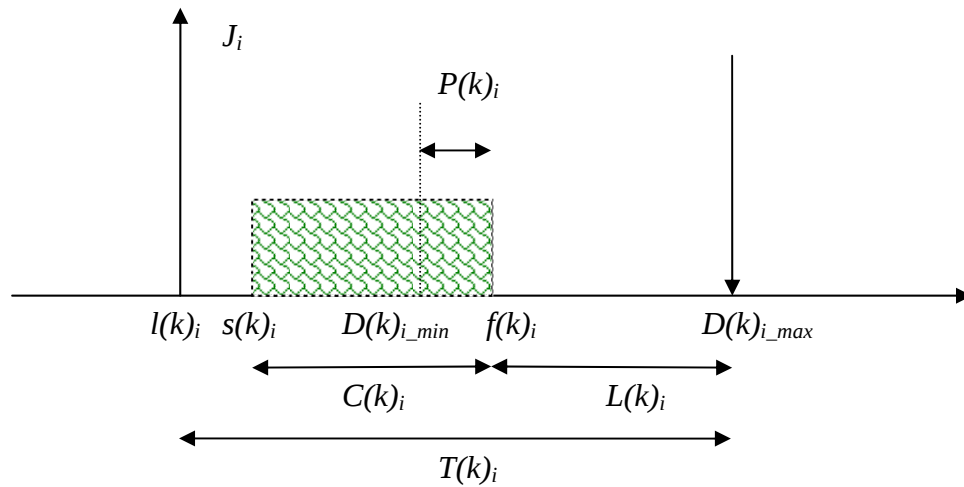


Figura 1.3 Parámetros de una Tarea en Tiempo Real [Lib1].

De acuerdo con modelos estocásticos desarrollados en algunos trabajos [Art12] se puede caracterizar a las TTR con los siguientes parámetros:

- Tiempo de arribo absoluto l_k
- Tiempo de interarribo π_k
- Fase ϕ_k
- Tiempo de inicio s_k

Lo anterior se muestra en la Figura 1.4. Tomando en cuenta que de acuerdo a [Art12] el tiempo de inicio $s_k = l_k + v_k$ donde v_k es una variable aleatoria denominada jitter y que depende del hardware y software del SD donde se encuentre el sistema.

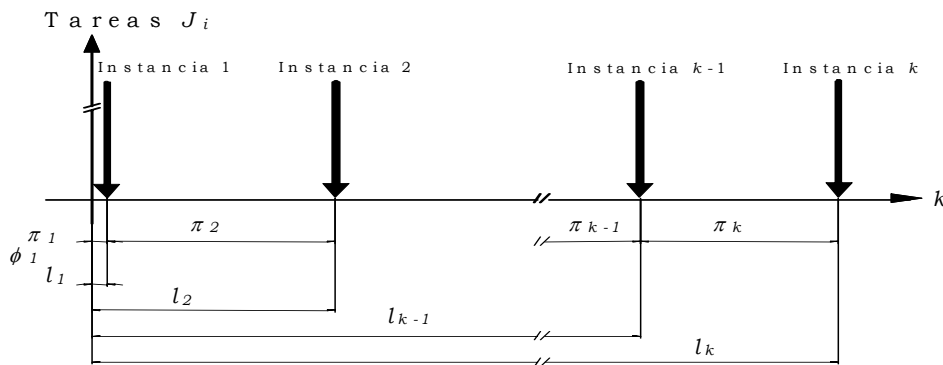


Figura 1.4 Instancias de una Tarea en Tiempo Real.

Con las caracterizaciones anteriores se han hecho modelos de TTR como los mencionados en [Art9] y en donde se da la clasificación de los tipos de TTR. Las

TTR se pueden clasificar de acuerdo a sus tiempos de arribo en [Art7] [Art8] [Art9] [Art10] [Art11] [Art13]:

- TTR Periódicas
- TTR Aperiódicas
- TTR Esporádicas

Las Tareas en Tiempo Real Periódicas (TTRP) son aquellas que se activan regularmente con periodo T_i y tiempo de calculo C_i , ambos conocidos y constantes, la restricción principal es que se tiene que cumplir que $d_i \leq T_i$.

Las Tareas en Tiempo Real Aperiódicas (TTRA) son aquellas que son activadas irregularmente con un periodo desconocido. La restricción es el plazo d_i .

Las Tareas en Tiempo Real Esporádicas (TTRE) son aquellas que se caracterizan por un tiempo de ejecución C_i , y un tiempo mínimo de interarribo T_i entre tiempos de activación.

1.6 Planificadores en Tiempo Real.

Una vez que los procesos o tareas se han descrito tanto de manera general, como en tiempo real, es importante explicar que estas están siendo procesadas por un SD, como puede ser el caso de una computadora que tiene instalado un SO. Este se encarga de darle atención a las tareas que solicitan alguna parte del hardware para poder completar su cometido. Todo lo anterior sería de cierto modo sencillo si solo fuera una tarea la que tuviera que atender el SO. La mayoría de los casos esto no es así, en el SO se están presentando varios procesos a la vez, es decir varias tareas que solicitan ser atendidas por parte de SO y algunas veces estas llegan de manera concurrente.

Se dice que un conjunto de tareas son concurrentes [Art6] cuando se ejecutan simultáneamente en forma no secuencial tomando recursos de CPU en diferentes instantes. Cuando sucede que existen dos o más tareas (concurrentes o no) es necesario que haya un planificador que administre los tiempos en los cuales se toman los recursos del CPU.

Un planificador es un algoritmo que ayuda al SO a tomar las decisiones de cuál proceso es el que debe tomar los recursos del CPU de la computadora [Lib17]. La principal circunstancia por la que el planificador toma esta decisión es debido a que existen dos o más procesos o tareas (concurrentes o no) que están solicitando atención por parte del SO.

Los algoritmos de planificación pueden clasificarse de dos maneras de acuerdo a como manejan las interrupciones de reloj. Estas dos formas de planificar son:

- **No expropiativo o no preferente.** Este algoritmo escoge el proceso que se ejecutará y simplemente le permite ejecutarse hasta que se bloquee o que voluntariamente entregue los recursos del CPU que le fueron asignados.
- **Expropiativo o preferente.** Este algoritmo escoge un proceso y le permite ejecutarse durante algún tiempo. Si al termino de este tiempo no ha terminado de ejecutarse el proceso, se suspende y el algoritmo escoge otro proceso asignandole un tiempo.

La forma en como escoge al proceso y el tiempo que se le asigna al proceso para su ejecución depende del algoritmo que este funcionando para la planificación de los procesos o tareas.

De acuerdo a los procesos que se necesitan administrar y del entorno donde se ejecutan estos procesos es el tipo de algoritmo de planificación que se elige en el sistema operativo. De acuerdo a los objetivos que se plantean en los procesos los algoritmos pueden ser de tres tipos:

- **Por lotes.** En este tipo no hay usuarios esperando en terminales a que se responda con rapidez. Debido a esto se pueden usar algoritmos expropiativos con ventanas de tiempo largos.
- **Interactivos.** En este se tienen usuarios por lo mismo se necesitan de algoritmos expropiativos para poder dar ventanas a usuarios y sus procesos.
- **Tiempo Real.** Aquí no es necesaria la expropiación debido a que los procesos o tareas están delimitadas en restricciones temporales en las cuales deben de terminar de actuar. Por lo general hacen su trabajo en su tiempo y se bloquean inmediatamente.

Los objetivos que se deben cumplir por parte de los algoritmos de planificación en los sistemas de acuerdo al entorno [Lib17] que se acaba de plantear se muestran en la Tabla 1.4.

Tipo de Sistema	Objetivos
Todos los Sistemas	• Equidad. Dar un tiempo de CPU a cada proceso para su ejecución. • Cumplimiento de políticas de planificación. • Equilibrio. Mantener ocupado todos los elementos del sistema.
Sistemas por Lotes	• Rendimiento. Procesar el máximo de tareas por unidad de tiempo.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

	• Tiempo de retorno. Reducir el tiempo entre el inicio y la terminación de cada tarea o proceso. • Utilización de CPU. Mantener ocupado la mayor parte del tiempo el CPU.
Sistemas Interactivos	• Tiempo de respuesta. Responder con rapidez a las solicitudes. • Proporcionalidad. Satisfacer a todos los usuarios
Sistemas en Tiempo Real	• Cumplir los plazos. Evitar la pérdida de datos.

Tabla 1.4. Clasificación de los Sistemas Operativos.

Cada uno de los tipos de sistemas que se mencionaron está relacionado con diferentes tipos de algoritmos de planificación. Estos algoritmos conforman políticas que tratan de cumplir con sus objetivos fundamentales.

En los sistemas por lotes se encuentran las siguientes políticas: Primero en llegar, primero en ser atendido, Trabajo más corto primero, Tiempo restante más corto a continuación, De tres niveles, entre otros.

En los sistemas interactivos se pueden tener los siguientes: Por turno circular, Por prioridades, Múltiples colas, Proceso más corto a continuación, Garantizada, Por lotería, Por porción equitativa.

En los sistemas de tiempo real los algoritmos de planificación que se pueden consideran de momento son los siguientes [Lib1] [Lib2] [Lib3] [Art14] [Art15]:

1.- Algoritmo de la Tasa Monotónica. Rate Monotonic. RM.

Este es un algoritmo de prioridades fijas o estáticas que consiste en darle la mayor prioridad a las tareas que tienen el menor periodo y la menor prioridad a las tareas con mayor periodo. Este algoritmo planifica fuera de línea. Este algoritmo soporta tareas en tiempo real periódicas.

2.- Algoritmo del Primer Plazo más Cercano. Earliest Deadline First. EDF.

Es un algoritmo dinámico que se elige las áreas a ejecutar de acuerdo a sus plazos absolutos. Las tareas con plazos cortos serán consideradas con prioridades altas mientras que las tareas con plazos largos serán consideradas con prioridades bajas. Las prioridades que se asignan a las tareas son dinámicas ya que conforme va transcurriendo el tiempo y una tarea que no se ha considerado para ejecutarse tiene su plazo próximo a vencerse, entonces su prioridad puede aumentar hasta que llegue el momento de ser ejecutada teniendo el algoritmo la posibilidad de desalojar a la tarea que se esta ejecutando en ese

instante. Este algoritmo soporta tareas en tiempo real periódicas, aperiódicas y esporádicas.

3.- Algoritmo del Plazo Monotónico. Deadline Monotonic. DM.

Este algoritmo considera prioridades estáticas donde las tareas tienen un plazo relativo más pequeño que el periodo de ejecución. Cada tarea tiene asignada una prioridad inversamente proporcional al plazo relativo. Soporta tareas periódicas.

4.- Primero en Llegar, Primero en ser Atendido. First Come First Serve. FCFS.

Este planificador esta comúnmente atendiendo tareas periódicas de alta prioridad cuando llegan tareas aperiódicas de baja prioridad y se les atenderá en el orden en el que llegan. Si en el momento en el que una tarea aperiódica pide atención, se esta ejecutando una tarea periódica, la tarea aperiódica tendrá que esperar a que la tarea periódica se termine de ejecutar.

5.- Servidor de Poleo. Pole Serve. PS.

Este algoritmo busca durante un periodo de tiempo de servidor T_s si hay tareas aperiódicas que atender, en dado caso de no encontrar se auto suspende hasta el siguiente periodo para atender tareas periódicas. Si alguna tarea aperiódica llega después de la auto suspensión esta tarea se tendrá que esperar hasta el siguiente periodo.

6.- Algoritmo del Servidor Aplazable. Deferred Serve. DS.

Se crea para mejorar el servidor de poleo al poder dar atención durante todo el periodo a tareas aperiódicas.

Una forma de clasificar a los algoritmos de planificación es de acuerdo a las prioridades fijas o dinámicas [Art14], otra forma de clasificarlos es si se hace la planificación en línea o fuera de línea o si el planificador es para un solo procesador o para multiprocesador [Art15].

1.7 Conclusiones del Capítulo 1.

Con las definiciones que se abordan en este primer capítulo es suficiente para marcar la relevancia que tienen los STR, del alcance, de su aplicación solo limitada por la imaginación. La importancia de los STR es cada vez mayor y se puede observar en muchas situaciones como el hecho de que en todo el mundo existen investigadores que le dedican su trabajo y parte de su vida, así también podemos ver como forma parte cultural de nuestro mundo, como un inciso más de la enseñanza en las escuelas y por dar un ejemplo, se toma el concepto de STR dentro de los diccionarios técnico como el de Oxford [Tes1] [Lib1] [Lib4].

Sin hacer mucho hincapié en las aplicaciones de la electrónica se puede decir que los SD son una herramienta fundamental para el control, la implementación y la simulación de los STR. Aunque los SD han funcionado durante mucho tiempo sin ser en TR, es importante hacer notar que en nuestro mundo todos los procesos son del mundo real y que no podemos pensar que estos procesos se estén controlando por sistemas rápidos o en línea cuando deberían ser en TR. Es tiempo que la dependencia de los SD con los STR sea estrecha y mutua.

Los conceptos que se tienen que tomar en cuenta para el estudio de las computadoras que están en relación con la teoría en TR son muchos sin embargo se pueden simplificar para este trabajo en los procesos o tareas y en los planificadores que administran los recursos de CPU que solicitan estas tareas. Hasta ahora ya se tienen modelos caracterizados y bien tipificados de la tareas en tiempo real [Art8], con estos modelos se puede trabajar en simulaciones de su comportamiento en distintas circunstancias. En lo que respecta a los planificadores que se ocupan de atender a las tareas en tiempo real, existen muchos de donde elegir para hacer pruebas, sin embargo a la fecha se siguen dando mejoras de los que ya existen y otros nuevos que dan posibilidades a distintas aplicaciones.

Una de las aplicaciones más importantes, es la de cubrir la necesidad de la comunicación, y hoy en día los SD cumplen muy bien este trabajo. Sin un SO es casi imposible el manejo adecuado de un SD, como una computadora, por lo mismo su estudio y comprensión son fundamentales para un buen entendimiento de las áreas de estudio que están a su lado, por fuera y dentro del mismo.

CAPÍTULO 2

PLANTEAMIENTO DEL PROBLEMA

2.1 Definición del problema.

Algunos desarrolladores [Lib1] [Lib2] [Lib3] de SO's ven a los planificadores y a sus algoritmos como modelos a seguir para la creación y el mejoramiento de otros algoritmos, en algunas ocasiones han considerado otras alternativas para un mejor desarrollo y así tener un mejor desempeño a la hora de manejar algunos tipos de procesos o tareas en el entorno en donde se quiere hacer accionar este planificador.

Los SD basados en microcontroladores o en microprocesadores son sistemas básicos para solucionar problemas que en el mundo real se presentan. La idea principal implica que cuando los algoritmos de planificación se quieren aplicar a los sistemas digitales en general, estos algoritmos de planificación pueden servir de gran ayuda para resolver estos problemas. Por lo mismo deben tener la forma de poderse medir de manera fácil y sencilla y así puedan servir como referencia al momento de implantarlos en los sistemas digitales.

Cómo deben funcionar los planificadores que actúan en un Sistema Operativo en Tiempo Real (SOTR) como el principal mecanismo de atención a las tareas o procesos que tienen que ser ejecutados por el procesador, es una de las cuestiones que se manejan en la teoría de Tiempo Real (TR) ya que una TTR tiene principalmente restricciones de plazos que se deben de cumplir. Tanto los planificadores convencionales como los que son para TR están bien tipificados a través de algoritmos y no tienen modelos de cómo deben de actuar [Art1] [Art14] [Art15]. Sin embargo pocos han estudiado la posibilidad de tener algún simulador programado con el cual poder ver el comportamiento de la TTR a lo largo de la ejecución de las mismas. El análisis y comportamiento interno de los planificadores se observan como cajas negras en el momento de actuar sobre los procesos o tareas tal y como se observa (Figura 2.1).

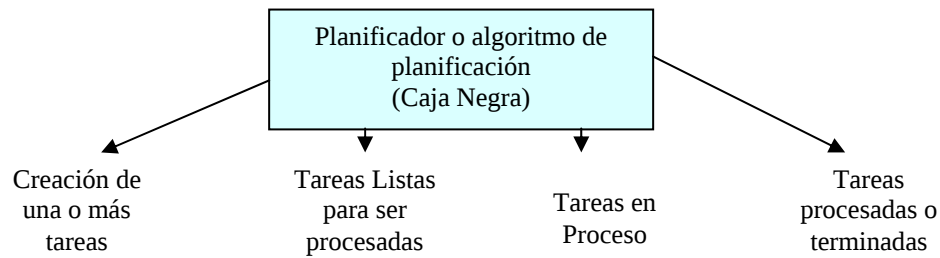


Figura 2.1 Comportamiento interno de un planificador.

2.2 Objetivo General.

- Hacer la descripción y el análisis la planificabilidad de algoritmos de planificación en tiempo real tales como el EDF y FIFO con tareas de tiempo real esporádicas usando para ello, los tiempos de arribo y los plazos mínimo y máximo así como el tiempo de ejecución o cálculo de las mismas a través de un modelo computacional.

2.3 Objetivos específicos.

- Usar modelos [Art9] [Art13] [Art7] de tareas en tiempo real que han caracterizado a las tareas como las esporádicas, periódicas y semiperiódicas para generar señales que puedan ser usadas en los algoritmos de planificación a tratar.
- Simular los algoritmos de planificación en tiempo real EDF y FIFO para hacer pruebas de planificabilidad con las tareas en tiempo real.
- Hacer comparaciones de los dos algoritmos de planificación con las distintas señales de entrada generadas por los modelos de tareas de tiempo real.
- Desarrollar un simulador que pueda analizar el comportamiento de los algoritmos de planificación EDF y FIFO y su vez dar las bases para la simulación del comportamiento de otros algoritmos de planificación.
- Dar las bases para el comportamiento de los algoritmos de planificación en los sistemas digitales aplicados a problemas del mundo real.

2.4 Justificación del proyecto.

Los STR cada vez más, cuentan con una relación con los sistemas del mundo real [Lib1] [Lib2]. Hoy en día se requieren que los sistemas tengan una relación más fuerte con la realidad con la que quieren interactuar. Estos sistemas no son tan comunes en México donde su desarrollo es poco o casi nulo, o demasiado lento en su implementación. Por lo mismo es importante tener un acercamiento con estos,

estudiarlos a profundidad para tenerlos en cuenta en posibles aplicaciones con restricciones de tiempo.

Los STR pueden tener avances importantes si se hacen estudios más profundos sobre las bases teóricas que ya otros científicos han cimentado [Lib1] [Lib2] [Art1] [Art14]. Existen todavía circunstancias donde el conocimiento deja la duda de cómo funcionan, qué representan o cómo están establecidos. Aquí es donde cobra importancia este trabajo. Cuando se aplica la tecnología para descubrir los parámetros de sistemas que aunque están en operación, funcionando de manera si no óptima, si adecuada, entonces se puede comprender mucho mejor su comportamiento y de esta manera hacer mejoras observando su accionar interno.

2.5 Límites.

- Las tareas que se usan deben cumplir con tener distribución conocida que se aproxima a una distribución normal, es decir debe ser estacionario.
- El modelo de las tareas son descritas por una ecuación en diferencias estocástica de primer orden.
- El simulador solo toma en cuenta dos algoritmos de planificación: el EDF y el FIFO.
- El simulador solo presenta resultados para las tareas en tiempo real esporádicas ya que se pueden considerar como la generalización de las Periódicas, las Aperiódicas [Art14].
- Tanto la teoría como los simuladores se logran para comparar pares de tareas.

2.6 Alcances.

- Generar los algoritmos de planificación para los planificadores EDF y FIFO.
- Observar los estados internos de los algoritmos de planificación.
- Observar la complejidad algorítmica al considerar ambos casos.
- Analizar el factor de utilización para ambos algoritmos de planificación.
- En base al factor de utilización establecer la planificabilidad de las tareas.
- Crear simuladores de los algoritmos de planificación EDF y FIFO.

CAPÍTULO 3

CARACTERIZACIÓN Y REPRESENTACIÓN DE TAREAS EN TIEMPO REAL ESPORÁDICAS

3.1 Introducción

En este capítulo se menciona el modelo de tareas en tiempo real esporádicas así como simulaciones de señales que representan de cierta forma estas tareas para dar una idea de cómo podrían llegar en algún momento en un planificador especializado para atenderlas.

3.2 Caracterización de Tareas en Tiempo Real Esporádicas

Como ya se menciona en el Capítulo 1 las tareas en tiempo real esporádicas son aquellas que se caracterizan por un tiempo de ejecución C_i y un tiempo mínimo de interarribo T_i entre tiempos de activación [Art8] [Lib1]. Dentro del trabajo presentado en [Art8] se toma en cuenta que una tarea en tiempo real TTR es una entidad ejecutable J_i que al menos es caracterizada por un tiempo de arribo y una restricción temporal.

Se puede ver que estos parámetros forman parte del modelo presentado en [Art11] y que se aprecian en la figura presentada en el Capítulo 1. La figura se presenta de nuevo en la Figura 3.1.

En la Figura 3.1 se muestran todos los parámetros de una tarea en tiempo real [Art11]. El tiempo de arribo absoluto de una tarea en tiempo real es el tiempo en el cual la instancia pide atención al procesador en relación al origen temporal de referencia [Art12] [Art8].

De acuerdo a los tiempos de arribo de las tareas en tiempo real, una tarea en tiempo real esporádica TTRE se define como aquella donde su tiempo de liberación puede ser representado por una función. Esta función se puede considerar como aleatoria debido a las variaciones que puede tener alrededor de

la función que los describe [Lib4]. De acuerdo con esto su periodo esta acotado como $T_i < \infty \forall i \in N$.

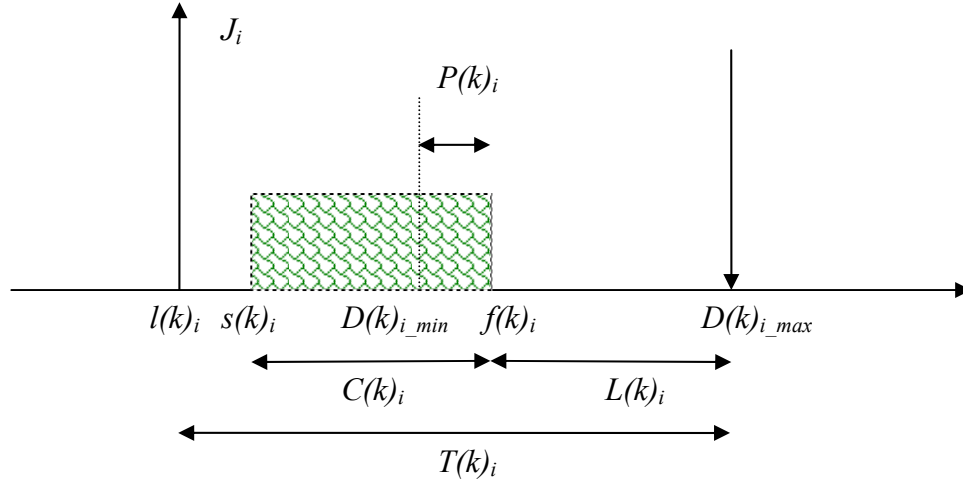


Figura 3.1 Parámetros de una Tarea en Tiempo Real.

El intervalo del periodo es conocido y caracterizado por un periodo de interarribo mínimo generado entre dos activaciones sucesivas. La restricción de tiempo es el plazo D_i y su tiempo de ejecución es constante. Las tareas esporádicas cuentan con dos tiempos de activación: un tiempo de ejecución C_i y un tiempo de interarribo mínimo T_i . Las tareas esporádicas son asociadas con eventos que responden a entradas temporales de señales periódicas, donde el intervalo de repetición de la señal varía y puede ser arbitrariamente largo pero acotado [Lib4] [Art1].

3.3 Características para el Simulador de Tareas en Tiempo Real Esporádicas.

A continuación se presentan los parámetros del simulador de las tareas en tiempo real esporádicas, el cual tiene sus bases en la caracterización de la sección anterior. Una TTR se toma como una tarea que inicialmente tiene los siguientes parámetros. Ver Figura 3.2.

Una TTR tiene un tiempo de arribo L_i , ya sea que arribe para que sea atendida por un microcontrolador o por el SO en la parte del planificador. Como se considera que las tareas pudieran ser concurrentes e independientes, dos tareas pueden tener el mismo tiempo de arribo sin que nada las relacione entre si.

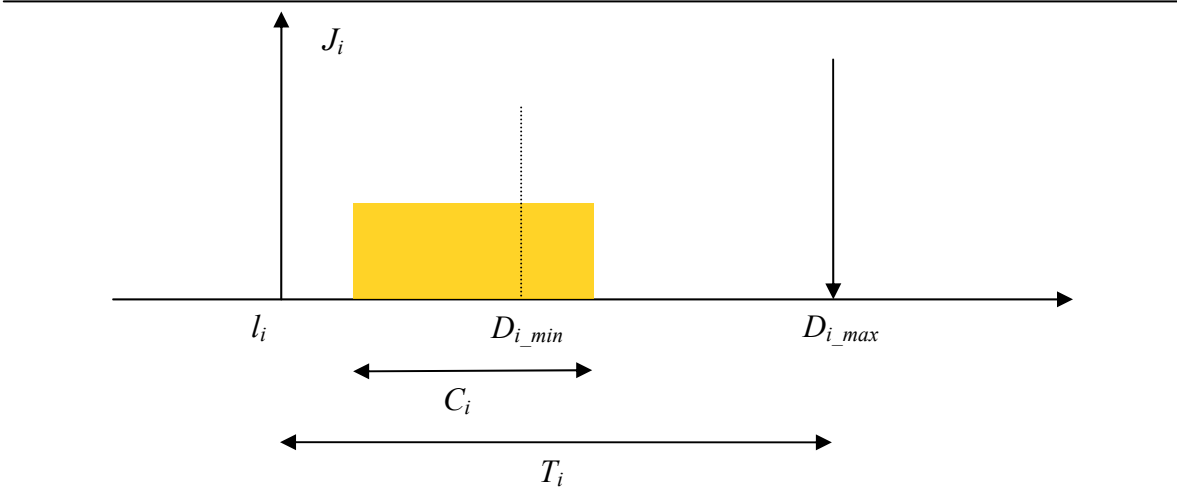


Figura 3.2 Características Usadas en un Simulador de TTR.

El tiempo máximo en el que la tarea debe de terminar de ser atendida es el plazo D_{i_max} . El periodo de ejecución T_i es el intervalo máximo en el cual debe de ser atendida la tarea desde que arriba al dispositivo que la atenderá. T_i es la diferencia entre el plazo máximo y el tiempo de arribo. Lo anterior se muestra en 3.1.

$$T_i = D_{i_max} - L_i \quad 3.1$$

El tiempo D_{i_min} es el plazo mínimo en el cual debe de terminar de atenderse la tarea. Antes de este plazo no esta permitido que se termine la tarea, como se muestra en la Figura 3.3, para no alterar la dinámica del sistema [Art7]. Cada tarea debe de cumplir con los plazos de terminación tanto mínimo como máximo.

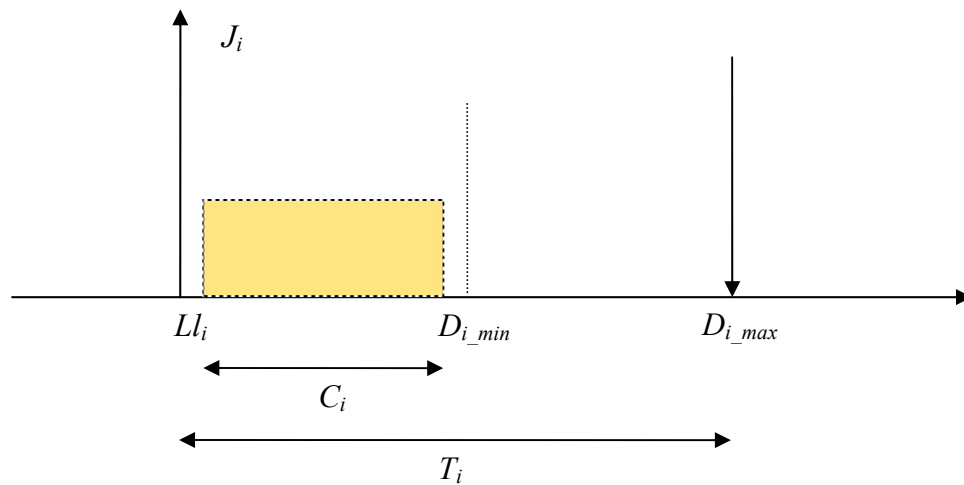


Figura 3.3 Tarea en Tiempo Real que no Cumple con el Plazo Mínimo.

El tiempo de terminación o finalizado de la tarea debe oscilar entre los límites fijados por los plazos máximo y mínimo como se muestra en 3.2.

$$f_i \in [D_{i_min}, D_{i_max}] \quad 3.2$$

El parámetro $C(k)_i$ que es el tiempo aproximado que la tarea tomará los recursos del procesador. Este parámetro se puede ver como un conjunto de operaciones discretas vistas como en la Figura 3.4. Estas operaciones tomaran los recursos del procesador hasta completar la tarea. Dependiendo del tiempo de procesamiento $C(k)_i$ es el número de operaciones que se requieren para terminar la tarea. Esta relación es directa proporcional.

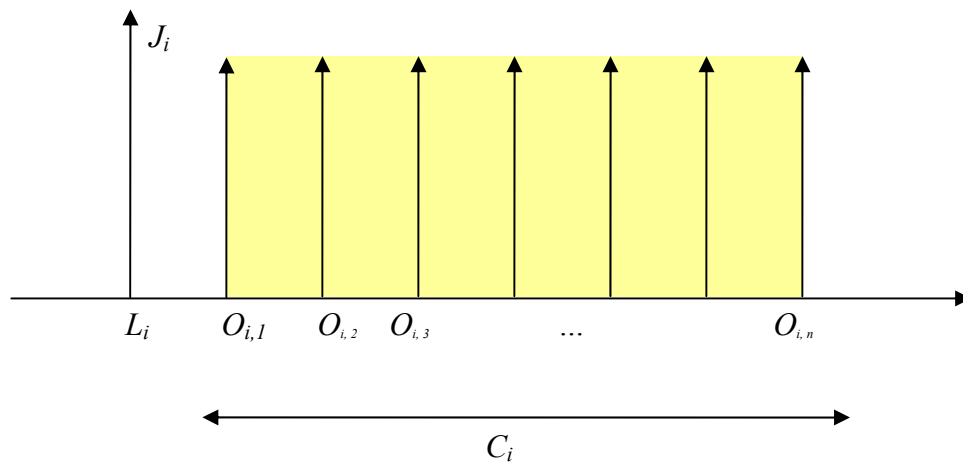


Figura 3.4 Conjunto de Operaciones Discretas de una Tarea en Tiempo Real.

Donde: $\{O_{i,n}\}$ es el conjunto de operaciones que se requieren para completar una tarea (o una simple instancia de la tarea).

3.4 Representación de los Tiempos Arribos de las Tareas en Tiempo Real

Cada tarea a ser programada tomará los parámetros descritos listados en la Tabla 3.1. El parámetro L_i , tiempo de arribo de cada tarea, es aleatorio e independiente de otras tareas que se estén arribando y puede coincidir con los tiempos de arribo de otras tareas. Es generado por el simulador. El plazo mínimo D_{i_min} , es el tiempo mínimo a partir del tiempo de arribo en el cual la tarea se terminará de procesar. En la simulación de las tareas, este se genera de forma aleatoria y es mayor que el tiempo de arribo de la tarea. El plazo máximo D_{i_max} , es el tiempo máximo en el cual debe de terminar de ejecutarse la tarea, es mayor que el plazo mínimo y es generado de forma aleatoria por el simulador.

Característica	Descripción
L_i	Tiempo de arribo
D_{i_min}	Plazo Mínimo
D_{i_max}	Plazo Máximo
$C(k)_i$	Carga

Tabla 3.1 Descripción de los Características de una Tarea en Tiempo Real.

El simulador primero toma en cuenta el tiempo de arribo de la tarea con el cual se tiene un conjunto de tareas que forma un banco de tareas de donde se tomaran más adelante para ejecutarse. Las tareas del banco de tareas se agrupan vectores que las almacenarán como se ve en la Figura 3.5. Los parámetros de las tareas se van guardando en el momento en que se va generando la tarea, como si fuera una tarea que esta siendo solicitada en el momento. En la siguiente sección se presentan resultados de las simulaciones de tareas tomando en cuenta todo lo anterior.

El tiempo de arribo de una tarea y los plazos tanto mínimo como máximo son tiempos discretos lo cual permite un nivel de procesamiento sencillo en la simulación. En lo que respecta al tiempo de cálculo de la tarea, la cual se toma como el tiempo de uso del procesador, esta dividida en n operaciones a realizar para completar la tarea. Por lo mismo este parámetro se considera como un contador decreciente donde cada vez que una operación se realice se hace un decremento en 1 a este contador, es decir un tiempo de cálculo con 10, quiere decir que necesita de 10 operaciones para completar su cometido, necesita de 10 tiempos de uso de procesador o en algunos SD necesitará de 10 ventanas de uso de procesador. Cuando este llegue a 0 la tarea habrá terminado de ejecutarse.

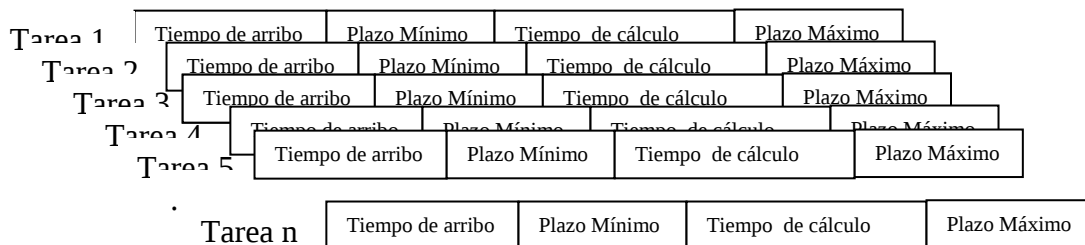


Figura 3.5. Banco de Tareas en Tiempo Real con sus Parámetros.

3.5 Resultados de la Simulación del Generador de Tareas .

El programa de la simulación del tiempo de arribo de las TTRE es desarrollado en MATLAB™, este programa esta en el Programa 1 en la sección Anexos. El programa calcula los tiempos de arribo de una TTRT de forma aleatoria en un

intervalo o ventana de 1 a 1000 unidades de ejecución. En la Figura 3.6 se muestra como están distribuidos estos tiempos de arribo a lo largo del intervalo ya citado el cual representa al eje del tiempo de manera discreta.

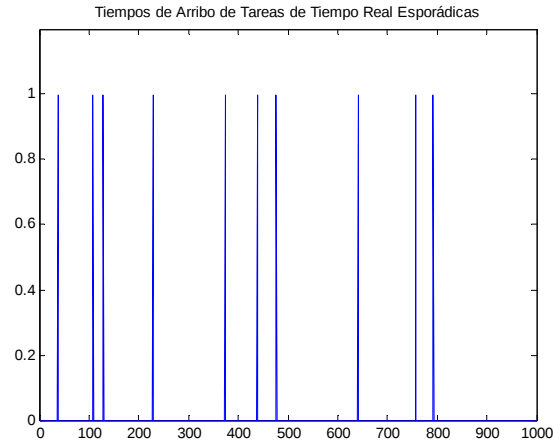


Figura 3.6 Tiempos de arribo a lo largo un intervalo 0-1000 unidades de ejecución. Simulación 1.

Esta figura representa los tiempos de arribo de las tareas en tiempo real esporádicas que son generadas con este simulador. La línea que en la figura representa una tarea tiene como medida un 1 el cual solo significa una medida de arribo. Otra simulación se muestra en la Figura 3.7. Cuando se toma una muestra de 10 simulaciones o de 10 intervalos el comportamiento de dicha muestra se ve en la Figura 3.8.

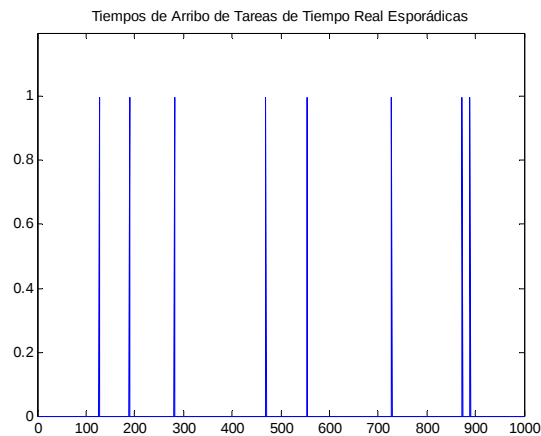


Figura 3.7 Tiempos de arribo a lo largo un intervalo 0-1000 unidades de ejecución. Simulación 2.

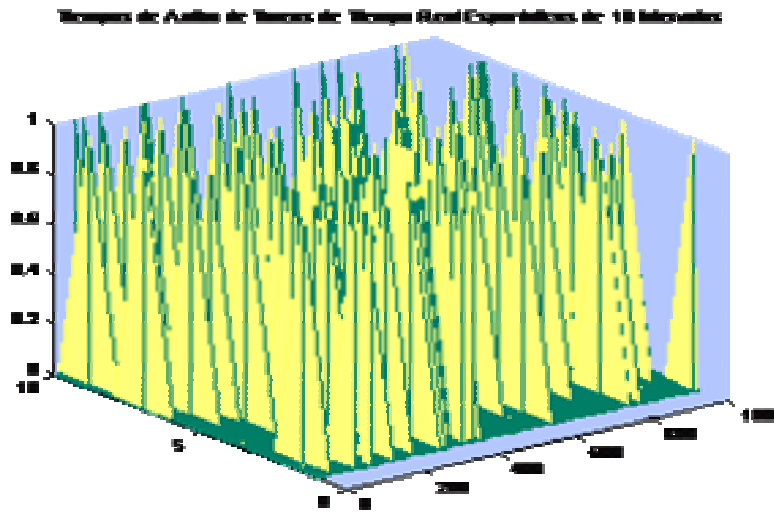


Figura 3.8 Muestra de 10 simulaciones o de 10 intervalos.

Tomando la media de cada una de los intervalos se colocan en una grafica de barras para representar la distribución que representa. Esta grafica se muestra en la Figura 3.9.

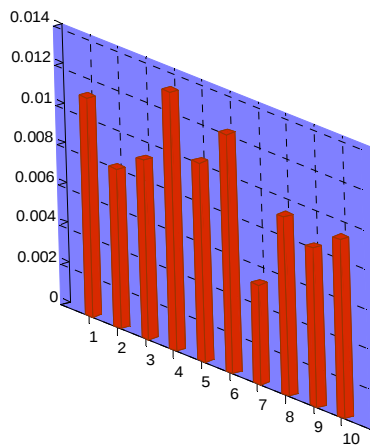


Figura 3.9 Media de cada una de los intervalos en una grafica de barras.

Al tomar 100 intervalos de simulación, la representación en grafica de barras se como en la Figura 3.10.

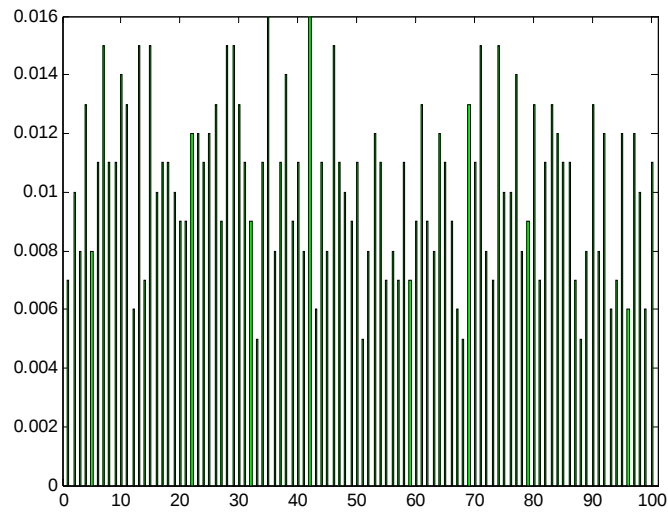


Figura 3.10 100 intervalos de simulación en grafica de barras.

En la Figura 3.11 se muestra el comportamiento de la medias de las tareas después de 100 arribos con 100 repeticiones y para el mismo caso en la Figura 3.12 se muestra el histograma de este comportamiento.

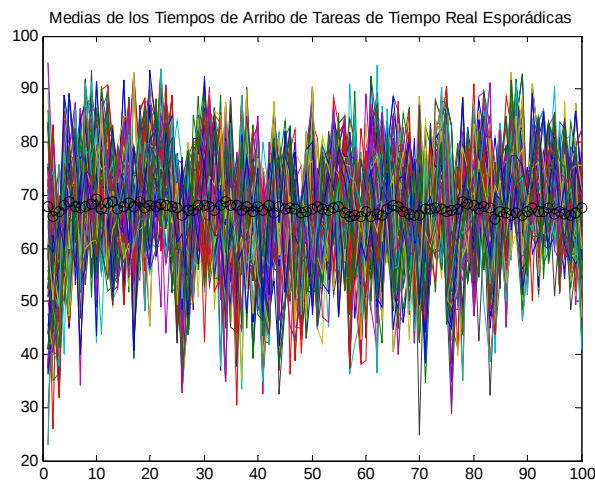


Figura 3.11 Simulación de 100 arribos con 100 repeticiones.

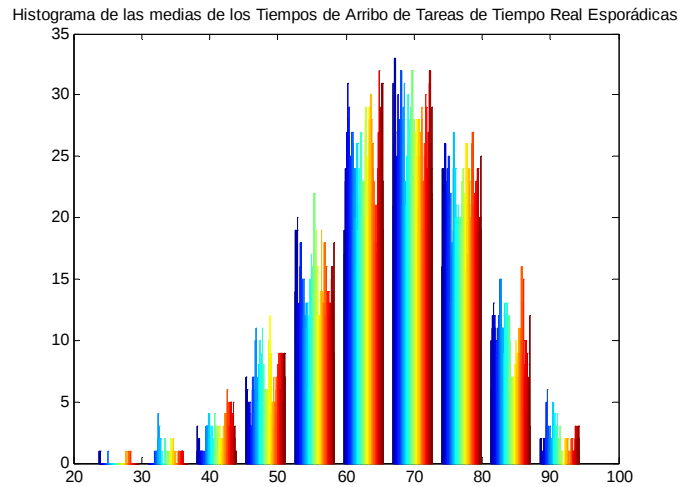


Figura 3.12. Simulación de 100 arribos con 100 repeticiones. Histograma.

3.6 Conclusiones del Capítulo 3

En el análisis de las TTR se tienen algunos modelos [Art9] [Art13] [Art7] que tienen la suficiente información para describir el comportamiento y características de las tareas. Las características permiten que se tenga la referencia adecuada para construir simulaciones como el que en este capítulo se muestra. Con un simulador de TTR se logra un mayor entendimiento de cómo se comportan los procesos que en la teoría de TR se estudian. Esta base da la ventaja hacia el siguiente paso que es hacer que todas las tareas que van arribando a un SD se puedan atender con políticas de planificación. Por el momento una etapa más se cumple al lograr ejemplificar las tareas que en los siguientes capítulos y secciones se usarán.

CAPÍTULO 4

PLANIFICACIÓN EN TIEMPO REAL USANDO EDF PARA TTR ESPORÁDICAS.

4.1 Introducción.

En este capítulo se muestran algunas definiciones para el desarrollo del trabajo de la tesis y que ayudan a ver más claro como se estructura un planificador en tiempo real, cuales son sus características principales y los objetivos que debe de cumplir.

4.2 Definiciones.

Definición 1. (Planificador).

Un planificador es un programa que utiliza un algoritmo de planificación para tomar decisiones de planificación y poderle asignar recursos del sistema a las tareas que lo solicitan. [Lib2].

Definición 2. (Planificación).

Es la asignación de recursos del sistema a esas tareas para ejecutarlas hasta su completa terminación. Esta asignación de recursos a las tareas es de acuerdo a políticas preestablecidas. [Lib2].

Definición 3. (Tareas realizables).

Un conjunto de tareas se dice realizable o “feasible” si todo el conjunto de tareas puede ser completado de acuerdo a un conjunto de restricciones específicas. [Lib2].

Definición 4. (Tareas planificables).

Un conjunto de tareas se dice planificable si existe al menos un algoritmo que pueda producir planificación realizable o “feasible”. [Lib2].

Definición 5. (Tarea dura o “hard”).

Una tarea se dice del tipo dura o “hard” si es completada después de su plazo o “deadline” y esto trae consecuencias catastróficas para el sistema digital, por lo

mismo, en cualquier caso a priori debería de tener una garantía para el peor de los escenarios. [Lib2].

Definición 6. (Tarea suave o “soft”).

Una tarea se dice del tipo suave o “soft” si al perder un plazo disminuye el desarrollo del sistema pero no compromete su correcto comportamiento. [Lib2].

Definición 7. (Tiempo de arribo “L”).

El tiempo de arribo de una tarea es aquel donde la tarea esta lista para ser ejecutada o requiere atención por parte del sistema para utilizar recursos del mismo sistema. [Lib2].

Definición 8. (Tiempo de cómputo o tiempo de ejecución “C”).

Es el tiempo necesario para ejecutar una tarea sin interrupción. [Lib2].

Definición 9. (Plazo o “deadline” “D”).

Es el tiempo que marca el límite de terminación de una tarea para evitar un daño al sistema. [Lib2].

Definición 10. (Tiempo de relajamiento o “laxity”).

También llamado tiempo flojo y es el máximo tiempo que una tarea puede ser demorada desde su activación hasta completarse sin llegar a su plazo. [Lib2].

Definición 11. (Algoritmo de planificación dinámico).

Un algoritmo de planificación dinámico es aquel donde sus decisiones de planificación están basadas en parámetros dinámicos que pueden cambiar sobre la evolución del sistema. [Lib2].

Definición 12. (Algoritmo de planificación estático).

Es aquel donde sus decisiones de planificación están basadas en parámetros fijos o arreglados, asignados a las tareas antes de su ejecución. [Lib2].

Definición 13. (Algoritmo de planificación preventivo).

Una planificación “preemptive” o preventiva es aquella donde las tareas o tarea que se encuentra en ejecución puede ser arbitrariamente suspendida en cualquier tiempo con el fin de asignar recursos de procesador a otra tarea de acuerdo a las políticas de planificación predefinidas. En este tipo de planificación las tareas pueden ser ejecutadas en intervalos de tiempos disjuntos. [Lib2].

Definición 14. (Algoritmo de planificación no preventivo).

Una planificación “non-preemptive” o no preventiva es aquella donde la tarea una vez iniciada es ejecutada hasta ser completada. En este caso, todas las decisiones de planificación son tomadas hasta la terminación de la tarea que esta en ejecución. [Lib2].

Definición 15. (Algoritmo de planificación fuera de línea u “off-line”).

Se dice que un algoritmo de este tipo, ejecuta en su totalidad una tarea antes de su activación. Este tipo de algoritmos permite almacenar las tareas ejecutadas en una tabla y después ser ejecutadas por un despachador. [Lib2].

Definición 16. (Algoritmo de planificación en línea u “on-line”).

Se dice que un algoritmo de este tipo toma sus decisiones de planificación en el momento de ejecutar la tarea o cuando una tarea arriba al sistema y pide atención por parte de este. [Lib2].

Definición 17. (Algoritmo de planificación óptimo).

Un algoritmo de planificación se dice óptimo si este minimiza alguna función de costo dada, definida sobre el conjunto de tareas. Cuando ninguna función de costo es definida y solo existe el objetivo de lograr una planificación realizable, entonces un algoritmo es óptimo si este puede fallar al alcanzar un plazo solo sin ningún otro algoritmo de la misma clase puede alcanzarlo. [Lib2].

Definición 18. (Algoritmo de planificación heurístico).

Se dice heurístico si este tiende a ser óptimo pero no garantiza encontrar el algoritmo óptimo. [Lib2].

Definición 19. (Algoritmo de planificación clarividente).

Se dice clarividente si este conoce el futuro, es decir, si conoce con anticipación los tiempos de arribo de las tareas. Aunque en la práctica o en la realidad este algoritmo es teórico, sirve para comparar el desarrollo de los algoritmos reales contra la mejor posibilidad. [Lib2].

4.3 Las Tareas en Tiempo Real y la Planificación en Tiempo Real.

Las tareas que piden atención por parte del sistema digital tienen las características de ser concurrentes, independientes y no correlacionadas y que además forman un Banco de Tareas. Este banco de tareas se puede ejemplificar como en la Figura 4.1.

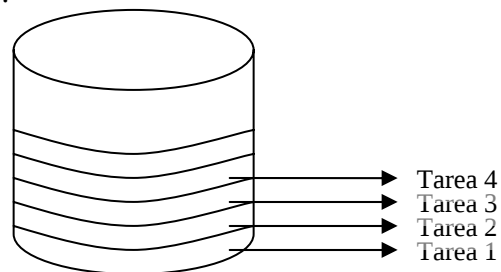


Figura 4.1 Representación de un Banco de Tareas.

Cada tarea en el banco de tareas debe tener las siguientes características, mostradas en la Figura 4.2.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real



Figura 4.2 Características de Tareas en Tiempo Real en el Banco de Tareas.

Cada tarea se puede expresar como:

$$T_i = \{L_i, D_{\min_i}, C_i, D_{\max_i}\} \subset \mathbb{R}^4 \quad 4.1$$

Cada una de las cantidades anteriores se mide en unidades de tiempo discreto y se denotarán adelante con la letra u .

Con esto, la característica de tiempo de ejecución C ilustrada en la Figura 4.2 se puede dividir en n operaciones que en su conjunto forma la tarea en sí. Esto se puede representar con la Figura 4.3.

Tarea i

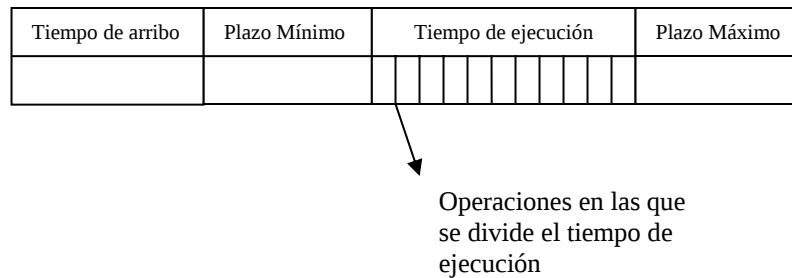


Figura 4.3 Tiempo de Ejecución C Dividido en n Operaciones.

El número de operaciones es variable para cada tarea y se representa por unidades de tiempo discreto. En cada tarea se podría conocer de manera estocástica cuantas de estas divisiones debe tener para su cálculo inmediato o posterior. De esta manera se cuenta con una medida a priori del tiempo discreto aproximado de cálculo de la tarea i -ésima. Esto podría resultar de mucha ayuda para la planificación de las Tareas ya que con estos parámetros de carga, en el momento en el que un planificador pretenda dejar pendiente una Tarea para realizar otra, se podría almacenar la información de donde se dejaron pendientes las Tareas o de otra manera se conoce en que operación en particular se dejó pendiente una Tarea y de esa manera cuando se quiera retomar la Tarea se tenga la información a la mano.

Dentro de las características que se toman en cuenta para la tarea está el tiempo de arribo L , el cual es un escalar que contiene información del tiempo discreto en

el que una tarea esta arribando al sistema digital para ser atendida. Otra característica es el plazo máximo D_{max} en el cual una tarea debe de ser completada, este plazo es un desplazamiento en el tiempo discreto a partir del tiempo de arribo y debe ser parte de la información que proporciona la tarea cuando arriba al sistema. También puede presentarse el caso donde el sistema tenga conocimiento de la tarea y conozca las características de esta, es decir hay una base de conocimientos establecida para algunas o para todas las tareas.

Una característica más es el plazo mínimo $D_{min,i}$ en el cual una tarea debe ser completada, este plazo también es una escalar considerado como un desplazamiento a partir del tiempo de arribo L_i . De igual manera que para el plazo máximo se debe tener información del tiempo en donde la tarea tiene su plazo mínimo.

Por poner un ejemplo de cómo están representadas las tareas en términos de números se puede suponer que en el tiempo discreto una tarea arriba o llega a un sistema digital diseñado para atenderla con los siguientes parámetros:

$L=10\ u$
$D_{min}=21\ u$
$D_{max}=24\ u$
$C(k)=9\ u$

Con estos parámetros se puede decir que existe una ventana con tamaño de 14 donde se debe de atender a la tarea y además la finalización de ella no debe ser menor al límite mínimo de 21 ni máximo de 24. Si lo ponemos en una grafica con escala de tiempo tendríamos lo mostrado en la Figura 4.4.

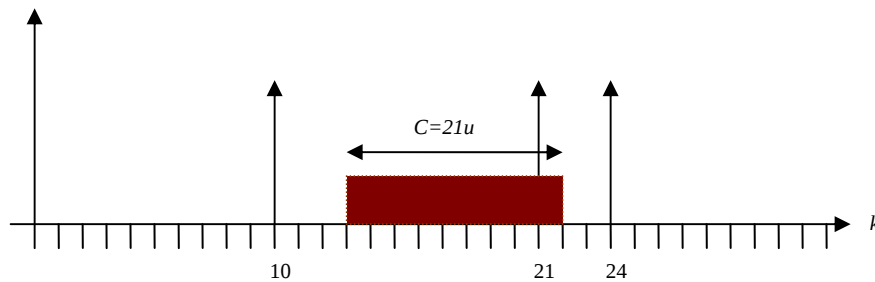


Figura 4.4 Ejemplo de Parámetros de una Tarea en Tiempo Real.

El hecho de que la tarea tenga un tiempo de arribo de $10u$ no implica que el procesador del sistema digital la atienda en ese instante. El sistema digital puede esperar a atender a la tarea siempre y cuando esto no implique que el retraso en la atención no perjudique su total terminación. El procesador podría atenderla a partir del tiempo $12u$ ya que el mínimo plazo para que la tarea sea completada es el de $21u$, si la tarea es atendida en $10u$ u $11u$ entonces el procesador tendría que esperar uno o dos tiempos intermedios para terminarla. En estos tiempos es posible que el procesador del sistema digital pueda atender a otras tareas que lo requieran.

En la Figura 4.5 se muestra como la tarea puede ser atendida en el tiempo 10 y ser terminada sin ser interrumpida en el tiempo 21, lo cual cumple con la restricción temporal de plazo mínimo.

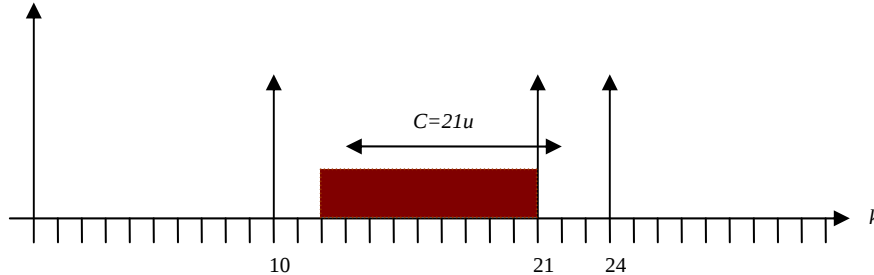


Figura 4.5 Ejemplo 1 de Atención a una Tarea con Parámetros en Tiempo Real.

En la Figuras 4.6, 4.7 y 4.8 se ve algo similar a la Figura 4.5 donde la tarea no es interrumpida y comienza en 13u, 14u y 15u respectivamente. En estas tareas se cumplen de nuevo las restricciones temporales de plazo mínimo con valor de 21u y plazo máximo con valor de 24u. Sin embargo no son las únicas circunstancias que rodean a la atención de la tarea con las características temporales que se han mencionado. En la Figura 4.9 (y solo por dar un ejemplo de las posibilidades que hay para atender a una tarea con restricciones temporales de plazos mínimo y máximo) se ve como una tarea es atendida desde su tiempo de arribo al sistema digital, es decir en 10u. Cuando el la tarea llega hasta el tiempo 16u se interrumpe para luego continuar en el tiempo 20u y terminar en el tiempo 23u.

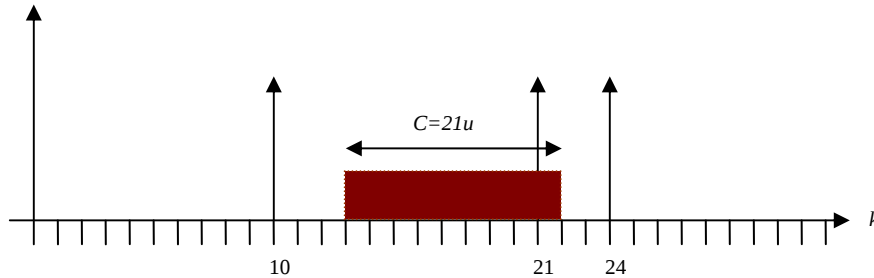


Figura 4.6. Ejemplo 2 de Atención a una Tarea con Parámetros en Tiempo Real.

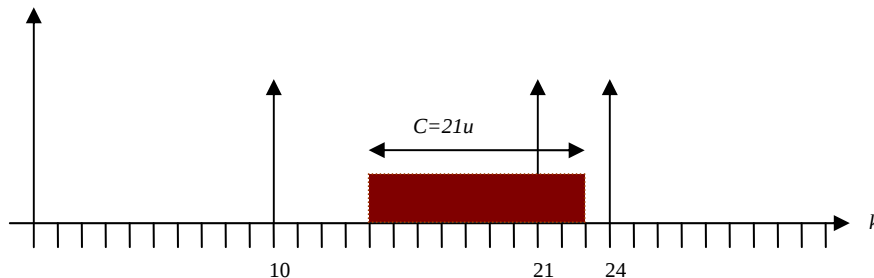


Figura 4.7 Ejemplo 3 de Atención a una Tarea con Parámetros en Tiempo Real.

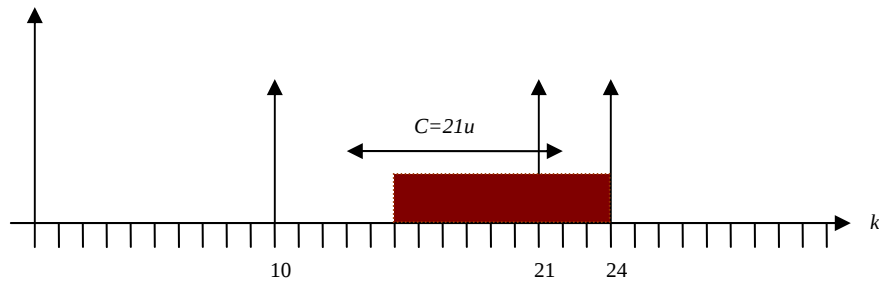


Figura 4.8 Ejemplo 4 de Atención a una Tarea con Parámetros en Tiempo Real.

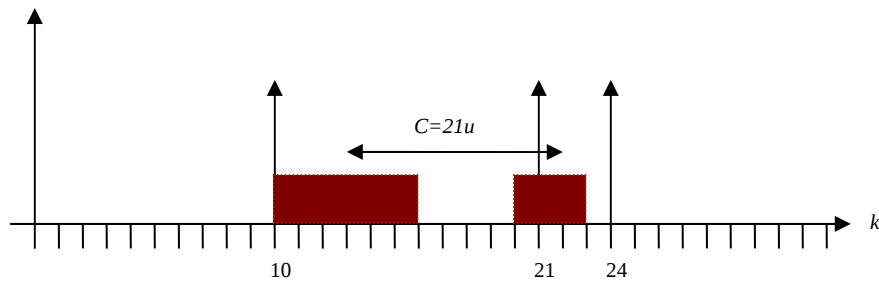


Figura 4.9 Ejemplo 5 de Atención a una Tarea con Parámetros en Tiempo Real.

4.4 Las Tareas en Tiempo Real y el Planificador en Tiempo Real EDF.

Dentro del Sistema Digital (Si es un Sistema de Cómputo, dentro del Sistema Operativo) los algoritmos de planificación se encargan de darle prioridad a las tareas que requieran de atención antes que otras de acuerdo con ciertas políticas establecidas. El planificador EDF Earliest Deadline First ó plazo más cercano primero marca su política en el plazo máximo, es decir, cuando existe más de una tarea pidiendo atención por parte del sistema digital este les asigna una prioridad para su atención de acuerdo al plazo o deadline que debe de cumplir cada tarea como restricción temporal. La prioridad que cada tarea tiene puede cambiar conforme el tiempo avanza, esto quiere decir que la prioridad es dinámica y esta en relación con el número de tareas y los plazos cercanos de cada una.

Poniendo el caso siguiente: Llega una tarea en el tiempo $L=5u$ y tiene las siguientes características:

$L=5u$
$D_{min}=28u$
$D_{máx}=35u$
$C=11u$

Esta tarea debe de atenderse desde el tiempo $5u$ pero no puede terminarse después del tiempo $35u$ ni antes del tiempo $28u$ y además tiene un tiempo de

ejecución de $C=11u$. Se supone también que esta es la primera tarea que llega a pedir atención al sistema digital, es decir el sistema esta relativamente desocupado. Por lo mismo la tarea esta siendo atendida desde que llega y no hay ninguna prioridad que asignar. Después en el tiempo 11 llega otra tarea con las siguientes características:

$L=11u$
$D_{\min}=20u$
$D_{\max}=29u$
$C=9u$

En este momento existen dos tareas pidiendo atención al sistema y entonces el sistema debe de asignar prioridades de acuerdo con el plazo más cercano y además esta prioridad debe ser dinámica conforme los plazos se van cumpliendo.

Hasta antes del tiempo 11 de arribo de la segunda tarea el diagrama de tiempos se puede ejemplificar como en la Figura 4.10.

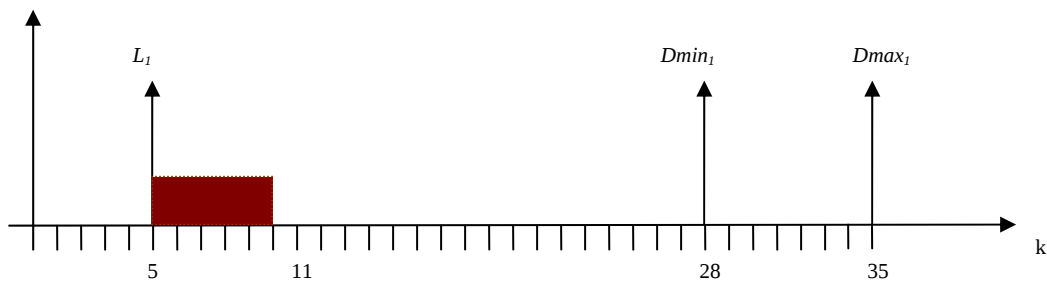


Figura 4.10. Tarea Ejecutada Hasta Antes del Tiempo 11.

En el instante de tiempo 11 cuando arriba la segunda tarea las condiciones de planificación comienzan. Primero, en la Figura 4.11 se resaltan las condiciones que rondan a esta segunda tarea. En este momento la primera tarea ya ha completado 6u de las operaciones que requiere para su ejecución y terminación y le faltan 5u. La segunda tarea tiene un plazo mas próximo por lo mismo la planificación indica que esta tarea debería tener mayor prioridad que la primer tarea. Además la primera tarea no debe de terminar de ejecutarse antes de 28u. Sin embargo hay que tomar en cuenta otros factores que pueden influir en el cálculo de la prioridad.

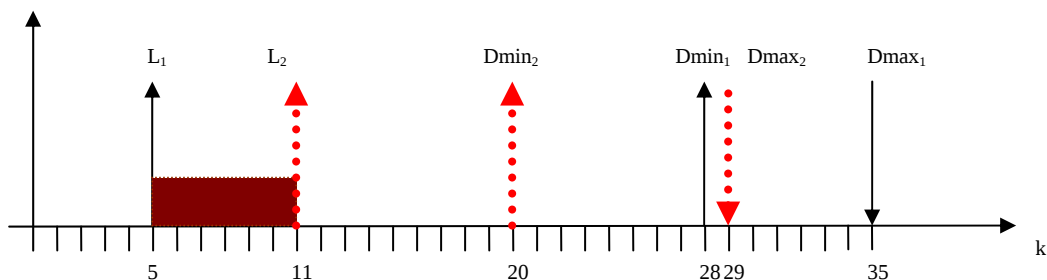


Figura 4.11 Se resaltan las Condiciones que Rondan a la Segunda Tarea.

La forma en como se debe calcular la prioridad requiere tomar en cuenta, el tiempo de arribo, y el plazo, tanto mínimo como máximo de cada tarea.

Para poder hacer los cálculos de prioridades y estados de las tareas que están en ejecución siendo atendidas por el sistema así como de las que no se requieren de parámetros de cálculo como los que se muestran a continuación:

Definición 20. (Tiempo de cómputo parcial. C_{p_i}).

El tiempo de cómputo parcial es el tiempo de cómputo completado de cada tarea hasta el instante más reciente o hasta el instante donde se necesita expropiar los recursos del sistema de la tarea.

Definición 21. (Tiempo de cómputo faltante. C_{f_i}).

Es el tiempo de cómputo que le falta a una tarea para su completa ejecución o terminación. Entonces:

$$C_{f_i} = C_i - C_{p_i} \quad 4.2$$

A partir de que una segunda tarea requiere ser atendida por el sistema se debe realizar una planificación de ambas. Debemos tomar en cuenta que el algoritmo EDF le asigna prioridades dinámicas a las tareas para que estas tengan el turno adecuado y puedan ser ejecutadas. Esta prioridad dinámica P se basa principalmente en el deadline o plazo y tienen la siguiente relación:

$$P_i \propto \frac{1}{D_{\max_i}} \quad 4.3$$

Sin embargo se pueden encontrar otras relaciones que influyen en la prioridad como el Tiempo de cómputo faltante, el cual se relaciona con la prioridad P de manera directa proporcional, es decir, entre más tiempo por ejecutar le quede a la tarea i , más es su prioridad. Esta relación se muestra en la siguiente expresión:

$$P_i \propto C_{f_i} \quad 4.4$$

Regresando al plazo, el parámetro $D_{\max}$ indica un tiempo en el cual se debe de haber terminado de ejecutar cierta tarea. Este parámetro no es movable y no cambia en el tiempo, por lo mismo debe de existir algún parámetro que este relacionado con este plazo y que indique la cercanía hacia él cada vez más conforme avanza el tiempo. Este parámetro se llama $T_{D\max_i}$.

Definición 22. (Tiempo faltante al plazo. $T_{D\max_i}$).

Es el tiempo que falta para que se cumpla el plazo y se calcula en cada unidad de tiempo que transcurre. La forma de representarlo es con la siguiente expresión:

$$T_{D_{\max_i}} = D_{\max_i} - k \quad 4.5$$

Donde k es el tiempo discreto actual de operación en el SD.

La relación del tiempo de 4.5 en relación con la prioridad P_i es inversamente proporcional ya que conforme avanza el tiempo el plazo esta más cerca y el tiempo que resta al plazo es más pequeño por lo que la prioridad debe de incrementarse. Esta relación se muestra en la siguiente expresión:

$$P_i \propto \frac{1}{T_{D_{\max_i}}} \quad 4.6$$

Una aproximación a esta planificación usando las características antes mencionadas se muestra a continuación con la siguiente expresión:

$$P_i = C_{f_i} \left(\frac{D_{\min_i} - L_i}{D_{\max_i} - L_i} \right) \quad 4.7$$

Esta expresión calcula la prioridad de cada tarea que arriba al sistema desde el momento en que arriba sin tomar en cuenta que halla otras tareas pidiendo atención del sistema.

La razón que esta entre paréntesis es la constante de proporcionalidad k y representa la cercanía con los límites mínimo y máximo. C_{f_i} es el tiempo de cálculo que le hace falta para realizar las operaciones que completen la tarea.

Los cálculos y la secuencia de los tiempos que presentan las tareas de acuerdo a la formula de la 4.7 se ven reflejados en la Tabla 4.1 que se muestra a continuación:

Tiempo k	Tarea 1				Tarea 2			
	Tiempo de ejecución C ₁ o tiempos donde la tarea 1 esta usando el procesador	Tiempo de computo parcial Cp ₁	Tiempo de computo faltante Cf ₁	Prioridad	Tiempo de ejecución C ₂ o tiempos donde la tarea 2 esta usando el procesador	Tiempo de computo parcial Cp ₂	Tiempo de computo faltante Cf ₂	Prioridad P ₂
1								
2								
3								
4								
5		0	11	8.43				
6		1	10					
7		2	9					
8		3	8					
9		4	7					
10		5	6					
11		6	5	3.83		0	9	4.5
12		6	5	3.83		1	8	4
13		6	5	3.83		2	7	3.5
14		7	4	3.05		2	7	3.5
15		7	4	3.05		3	6	3
16		8	3	2.3		3	6	3
17		8	3	2.3		4	5	2.5
18		8	3	2.3		5	4	2
19		9	2	1.53		5	4	2

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

20			10	1	0.76			5	4	1.5
21			10	1	0.76			6	3	1
22			10	1	0.76			7	2	0.5
23			11	0	0			7	2	0.5
24								8	1	0.5
25								9	0	0
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										

Tabla 4.1 Cálculos y Secuencia de Tiempos que Presentan las Tareas de acuerdo a 4.6.

Es importante señalar que la barra izquierda de la columna del tiempo de ejecución de la tarea 1 representa el tiempo que tiene la tarea i para ser atendida y terminada desde que arriba al sistema hasta que se cumple el plazo máximo. Los pedazos de color rojo representan los tiempos en que es procesada la tarea i . Aunque ambas tareas cumplen con sus plazos máximos, la tarea 1 no cumple con su plazo mínimo ya que termina antes de este que el se cumpla es decir $D_{min,1}=28$.

Esto se puede arreglar con algoritmia tomando la pregunta obligada: ¿Ha pasado el plazo mínimo de la tarea 1 o 2 o i ? Si ha pasado entonces termina la operación que termina la tarea. Si no bloquea la tarea hasta que su plazo mínimo llegue. Con esto entonces la Tabla 4.1 se vería como la Tabla 4.2 que se muestra a continuación:

Tiempo k	Tarea 1				Prioridad	Tarea 2				Prioridad P_2
	Tiempo de ejecución C_1 o tiempos donde la tarea 1 esta usando el procesador	Tiempo de computo parcial Cp_1	Tiempo de computo faltante Cf_1			Tiempo de ejecución C_2 o tiempos donde la tarea 2 esta usando el procesador	Tiempo de computo parcial Cp_2	Tiempo de computo faltante Cf_2		
1										
2										
3										
4										
5		0	11	8.43						
6		1	10							
7		2	9							
8		3	8							
9		4	7							
10		5	6							
11		6	5	3.83			0	9	4.5	
12		6	5	3.83			1	8	4	
13		6	5	3.83			2	7	3.5	
14		7	4	3.05			2	7	3.5	
15		7	4	3.05			3	6	3	
16		8	3	2.3			3	6	3	
17		8	3	2.3			4	5	2.5	
18		8	3	2.3			5	4	2	
19		9	2	1.53			5	4	2	
20		10	1	0.76			5	4	1.5	
21		10	1	0.76			6	3	1	
22		10	1	0.76			7	2	0.5	
23		10	1	0.76			8	1	0.5	
24		10	1	0.76			9	0	0	
25		10	1	0.76						
26		10	1	0.76						
27		10	1	0.76						
28		11	0	0.76						
29										

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

30									
31									
32									
33									
34									
35									

Tabla 4.2. La Prioridad de la Tarea 1 no se Toma debido al Plazo Mínimo.

En la Tabla 4.2 se puede ver que la prioridad de la tarea 1 no se tomo en cuenta debido a que el plazo mínimo fue el que determino que dicha tarea debía esperar hasta alcanzar dicho plazo. Mientas tanto la tarea 2 terminó un tiempo antes su ejecución sin dejar de cumplir con sus plazos. En algunos sistemas digitales principalmente los computacionales, donde hay Sistemas Operativos SO complejos, el hecho de cambiar de contexto entre varias tareas implica pérdidas en el tiempo de uso de procesador. Tomando la perdida de uso de procesador en el cambio de contexto en el ejemplo de las dos tareas como un tiempo de ejecución perdido la Tabla 4.2 quedaría como la Tabla 4.3 que se muestra a continuación:

Tiempo k	Tarea 1				Prioridad	Tarea 2				Prioridad P ₂
	Tiempo de ejecución C ₁ o tiempos donde la tarea 1 esta usando el procesador	Tiempo de computo parcial Cp ₁	Tiempo de computo faltante Cf ₁			Tiempo de ejecución C ₂ o tiempos donde la tarea 2 esta usando el procesador	Tiempo de computo parcial Cp ₂	Tiempo de computo faltante Cf ₂		
1										
2										
3										
4										
5		0	11	8.43						
6		1	10							
7		2	9							
8		3	8							
9		4	7							
10		5	6							
11		6	5	3.83						
12		6	5	3.83			0	9	4.5	
13		6	5	3.83			1	8	4	
14		6	5	3.83			2	7	3.5	
15		6	5	3.83			2	7	3.5	
16		7	4	3.05			2	7	3.5	
17		7	4	3.05			2	7	3.5	
18		7	4	3.05			3	6	3	
19		7	4	3.05			3	6	3	
20		8	3	2.3			3	6	3	
21		8	3	2.3			3	6	3	
22		8	3	2.3			4	5	2.5	
23		8	3	2.3			5	4	2	
24		8	3	2.3			5	4	2	
25		9	2	1.53			5	4	2	
26		9	2	1.53			5	4	2	
27		9	2	1.53			6	3	1.5	
28		9	2	1.53			6	3	1.5	
29		10	1	0.76			6	3	1.5	
30		10	1	0.76			6	3	1.5	
31		10	1	0.76			7	2	1	
32		10	1	0.76			8	1	0.5	
33		10	1	0.76			8	1	0.5	
34		11	0	0			8	1	0.5	
35		11	0	0			8	1	0.5	
36		11	0	0			9	0	0	
37		11	0	0						
38		11	0	0						

Tabla 4.3. Secuencia de Tiempos Tomando en Cuenta el Cambio de Contexto.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

En esta Tabla 4.3 se ve como el cambio de contexto repetido que algunos Sistemas Digitales tienen puede provocar una gran pérdida de tiempo y ocasionar que no se cumplan los plazos. Sin embargo los SD que hoy día se manejan no sufren por pérdidas en el cambio de contexto debido a las velocidades de procesamiento que manejan, es decir son sistemas muy rápidos y por lo mismo pueden hacer el cambio de contexto más rápido que las operaciones que consumen tiempo de uso de procesador.

La forma de evitar que nos encontremos problemas con los SD que no puedan procesar más rápido los cambios de contexto es con la siguiente expresión:

$$P_i = \frac{C_{f_i}}{T_{D_{\max_i}}} \left(\frac{D_{\min_i} - L_i}{D_{\max_i} - L_i} \right) \quad 4.8$$

Tomando en cuenta esta nueva aproximación para calcular la prioridad se presenta la forma como queda la Tabla 4.3 en la Tabla 4.4. Como se puede ver el número de cambios de contexto es menor aunque la tarea 2 considerada como crítica debido a que tiene un tiempo de ejecución elevado y un periodo de ejecución corto, no se ha cumplido del todo.

Tiempo k	Tarea 1				Tarea 2			
	Tiempo de ejecución C ₁ o tiempos donde la tarea 1 esta usando el procesador	Tiempo de computo parcial Cp ₁	Tiempo de computo faltante Cf ₁	Prioridad	Tiempo de ejecución C ₂ o tiempos donde la tarea 2 esta usando el procesador	Tiempo de computo parcial Cp ₂	Tiempo de computo faltante Cf ₂	Prioridad P ₂
1								
2								
3								
4								
5		0	11	0.281				
6		1	10	0.264				
7		2	9	0.2464				
8		3	8	0.2271				
9		4	7	0.2064				
10		5	6	0.184				
11		6	5	0.1597		0	9	0.25
12		6	5	0.1666		0	9	0.2647
13		6	5	0.1742		1	8	0.2352
14		6	5	0.1825		2	7	0.2333
15		6	5	0.1916		3	6	0.2142
16		6	5	0.2017		4	5	0.1923
17		6	5	0.2129		4	5	0.2083
18		7	4	0.1803		4	5	0.2272
19		7	4	0.1916		4	5	0.2500
20		7	4	0.2044		5	4	0.2222
21		7	4	0.2190		6	3	0.1875
22		7	4	0.2358		6	3	0.2142
23		8	3	0.1916		6	3	0.2500
24		8	3	0.2090		6	3	0.3000
25		8	3	0.2300		7	2	0.2500
26		8	3	0.2555		8	1	0.1666
27		8	3	0.2875		8	1	0.2500
28		9	2	0.2190		8	1	0.5000
29		9	2	0.2555		8	1	∞
30		9	2	0.3066		9	0	
31		9	2	0.3833				
32		10	1	0.2555				
33		11	0	0				
34								
35								
36								

Tabla 4.4. Cálculos y Secuencia de Tiempos que Presentan las Tareas de acuerdo a 4.7.

4.5 Conclusiones del Capítulo 4.

En este capítulo se pudo encontrar una forma de calcular la prioridad dinámica del algoritmo EDF en base a sus propios parámetros y características. Esto ayudará en el siguiente capítulo a la simulación del algoritmo sin tomar en cuenta que la plataforma de simulación es la que se desarrolle. Además da pauta a que los resultados puedan utilizarse en sistemas digitales aplicados a usos específicos.

CAPÍTULO 5

SIMULACIÓN DEL ALGORITMO EDF CON TTR ESPORADICAS.

5.1 Introducción.

En este capítulo se usa la teoría del planificador Earliest Deadline First (EDF), considerando las principales características expuestas en el capítulo anterior, para exponer un algoritmo simulado que sirva como referencia para sistemas digitales que lo requieran.

5.2 Algoritmo de Planificación EDF.

Dentro de la teoría de la planificación en los Sistema Digitales como los computacionales se encuentran algunas clasificaciones de los algoritmos de planificación: Algoritmos de planificación estáticos y algoritmos de planificación dinámicos. En los estáticos se requiere conocer previamente las características del proceso aunque no es necesario ocupar mucho tiempo en línea para hacer los cálculos necesarios. En comparación con los dinámicos que no es necesario conocer previamente las características del proceso a planificar, sin embargo se requiere más tiempo en línea para hacer los cálculos. Sin embargo este tipo de algoritmos proporciona una estructura de un sistema más flexible para reaccionar a los cambios de actividad de una manera anticipada [Art15].

Otra forma de ver los algoritmos de planificación es desde el punto de vista de la prevención de monopolizar al procesador cuando hay tareas múltiples. Cuando se prevé esta situación donde el algoritmo permite que solo se le de un tiempo de uso de procesador a cada tarea o cuando una tarea se deja de ejecutar debido a prioridades de otras entonces se dice que el algoritmo es “preemptive” y cuando sucede lo contrario se le llama “non-preemptive” [Art15].

En el caso del algoritmo Earliest Deadline First (EDF), su clasificación es la que sigue: es un algoritmo dinámico con la característica “preemptive”. Es decir puede asignar prioridades durante el tiempo de ejecución de acuerdo a los plazos cercanos y en caso que alguna tarea se este ejecutando en ese momento y otra tenga una mayor prioridad para ejecución, la tarea anterior se deja de ejecutar para cumplir con la que tiene el plazo más cercano.

5.3 De la Teoría a la práctica. El simulador del Algoritmo EDF.

En un sistema digital como los sistemas computacionales donde muchas tareas requieren ser procesadas, los sistemas operativos juegan un papel muy importante, porque como ya se ha mencionado el SO es el que administra los recursos del sistema, ya sea uso de procesador o uso de periféricos o cualquier hardware que el sistema tenga. Dentro del sistema operativo uno de los programas más importantes es el que planifica la asignación de tiempos de uso del procesador. Este programa es llamado planificador. Existen muchos tipos de planificadores [Lib17] que se diseñaron para diferentes tipos de procesos o tareas, los que son incluidos en este trabajo son los que planifican tareas relacionadas con los sistemas en tiempo real. Debido al auge que han tenido los sistemas en tiempo real (STR) [Art14], el interés sobre la mejora de los mismos así como el estudio de nuevas técnicas constantemente genera más información que sirven para describirlos todavía mejor.

Sin embargo, esta teoría es incompleta si no se pone en práctica. Para evitar este problema con los sistemas que usan un algoritmo de planificación como el EDF, este capítulo se planeo para dar las bases suficientes para generar un simulador que pueda acercarse lo más posible a los que hay trabajando en el mundo real, tomando en cuenta la teoría ya vista en el capítulo anterior.

La idea de ampliar la teoría del algoritmo de planificación en tiempo real EDF no se centra en los sistemas digitales como los computacionales ya que estos pueden estar usando actualmente estos algoritmos de una u otra manera. La idea se centra en los sistemas que aun no usan este algoritmo y que son entre otros los sistemas embebidos o los que están dedicados a tareas específicas en un determinado entorno.

El algoritmo de planificación que se muestra a continuación resuelve el problema de planificar 2 tareas con los siguientes parámetros cada una:

<i>Tarea 1</i>	<i>Tarea 2</i>
$L=5u$	$L=11u$
$D_{\min}=28u$	$D_{\min}=20u$
$D_{\max}=35u$	$D_{\max}=29u$
$C=11u$	$C=9u$

En la Figura 5.1 se muestra el algoritmo que se usa para la simulación en MATLAB™.

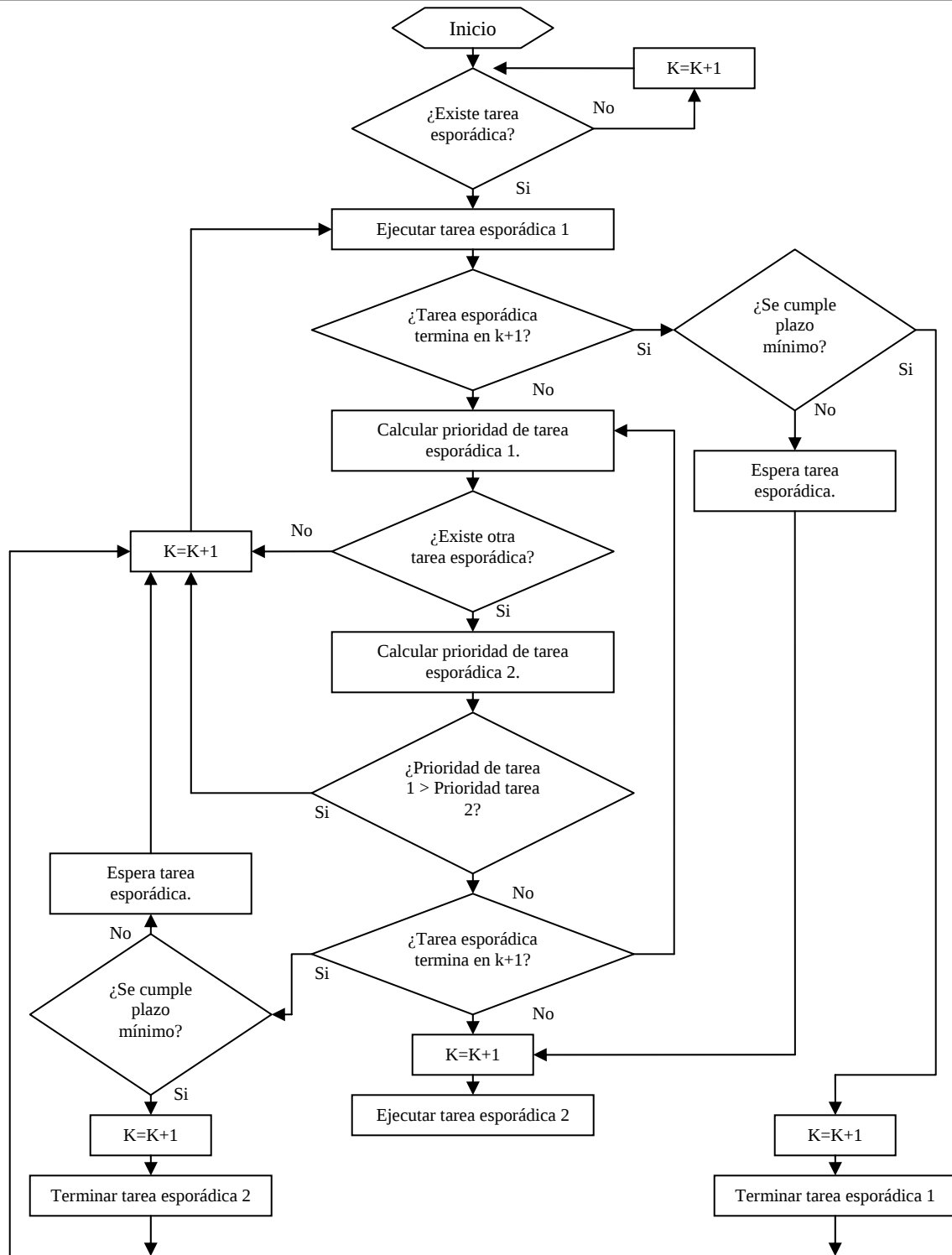


Figura 5.1 Diagrama de Flujo para el Algoritmo EDF Usado en la Simulación en MATLAB™.

A continuación se muestran los resultados de la simulación en MATLAB™ con los datos y formulas dados en el capítulo 4.

En la Figura 5.2 se muestra el progreso de las prioridades dinámicas de dos tareas.

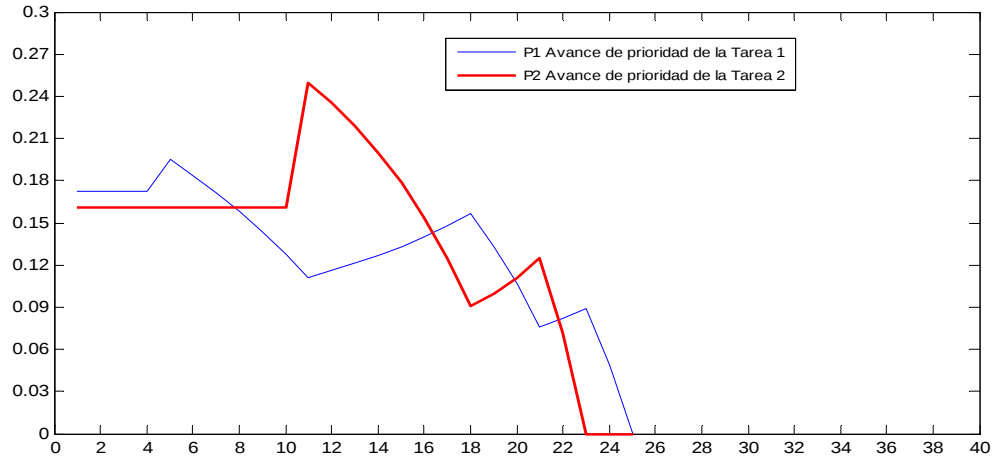


Figura 5.2 Progreso de Prioridades Dinámicas de la Simulación de Dos Tareas.

En la Figura 5.3 se observa el progreso de los tiempos de ejecución de cada tarea. Se observa también que los plazos tanto mínimos como máximos se cumplen.

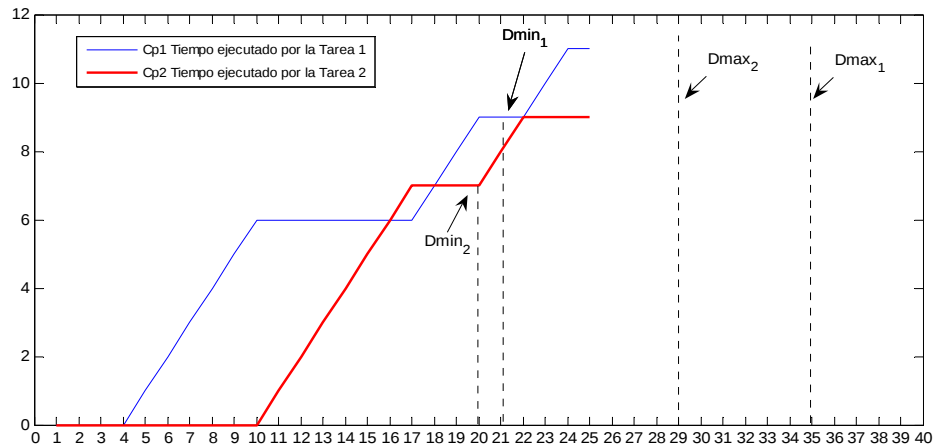


Figura 5.3 Progreso de los Tiempos de Ejecución de la Simulación de Dos tareas.

En la Figura 5.4 se muestra los tiempos de ejecución que faltan por cada una de las tareas.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

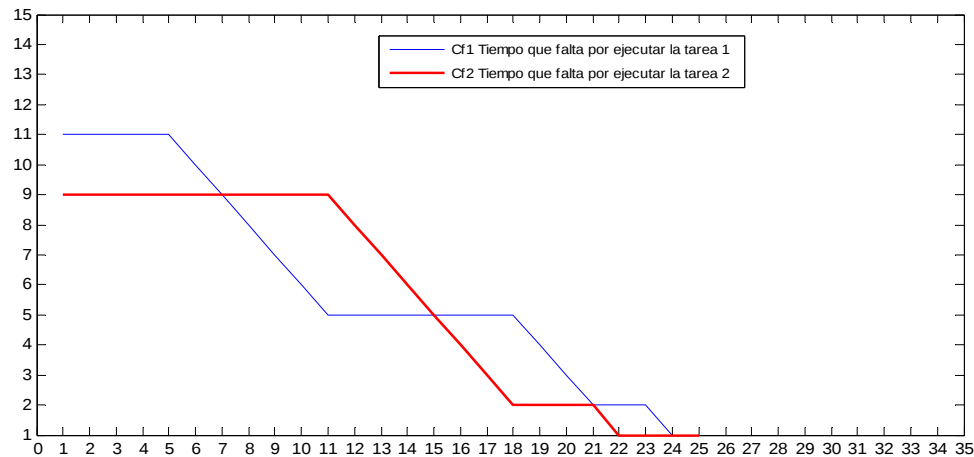


Figura 5.4 Progreso de los Tiempos de Ejecución Faltantes en la Simulación de Dos Tareas.

A continuación se muestra en la Tabla 5.1 un conjunto de 34 ejecuciones con diferentes datos para cada una de las dos tareas en cuestión.

Muestra		<i>L</i>	<i>Dmin</i>	<i>C</i>	<i>Dmax</i>	Funciona
1	T1	5	21	11	35	Si
	T2	11	20	9	29	
2	T1	8	40	30	70	Si
	T2	20	42	32	75	
3	T1	15	40	30	70	Si
	T2	22	42	32	75	
4	T1	15	40	30	92	Si
	T2	22	42	32	80	
5	T1	21	40	30	92	Si
	T2	29	42	32	80	
6	T1	28	40	30	92	Si
	T2	29	42	32	80	
7	T1	28	40	30	92	Si
	T2	31	42	32	80	
8	T1	30	40	30	92	No
	T2	31	42	32	80	
9	T1	29	40	30	92	Si
	T2	31	42	32	80	
10	T1	29	80	56	105	No
	T2	31	64	40	98	
11	T1	29	80	30	105	Si
	T2	31	64	32	98	
12	T1	29	100	30	105	Si
	T2	31	80	32	98	

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

13	T1	17	35	7	40	Si
	T2	21	40	22	58	
14	T1	17	35	17	40	No
	T2	21	40	22	58	
15	T1	16	35	17	40	No
	T2	21	40	22	58	
16	T1	10	35	17	40	Si
	T2	21	40	22	58	
17	T1	10	37	17	40	Si
	T2	21	40	22	58	
18	T1	14	35	17	40	No
	T2	21	40	22	58	
19	T1	12	35	17	40	Si
	T2	21	40	22	58	
20	T1	12	35	3	40	Si
	T2	21	40	5	58	
21	T1	12	35	13	40	No
	T2	21	40	25	58	
22	T1	12	30	13	40	Si
	T2	21	55	25	58	
23	T1	12	30	13	40	Si
	T2	21	55	25	58	
24	T1	12	30	16	40	Si
	T2	21	55	26	58	
25	T1	12	30	20	40	No
	T2	21	55	28	58	
26	T1	12	30	20	40	No
	T2	21	55	24	58	
27	T1	12	30	16	40	No
	T2	21	55	28	58	
28	T1	12	30	16	40	Si
	T2	21	55	25	58	
29	T1	12	30	16	40	Si
	T2	21	55	26	58	
30	T1	12	30	16	40	No
	T2	21	55	27	58	
31	T1	12	30	17	40	Si
	T2	21	55	26	58	
32	T1	12	30	18	40	Si
	T2	21	55	26	58	
33	T1	12	30	19	40	Si
	T2	21	55	26	58	
34	T1	12	35	20	40	No
	T2	21	55	26	58	

Tabla 5.1 Conjunto de 34 Ejecuciones con Diferentes Datos Aleatorios.

En la Tabla 5.1 se muestra como algunas ejecuciones no funcionan debido a que los parámetros no se adecuan con el cálculo de la prioridad, por lo mismo se necesita de algún calculo extra que permita determinar si el par de tareas se ejecutara satisfactoriamente.

El cálculo que se menciona se puede describir aproximadamente como:

$$D_{\max_i} \propto (L_i + C_i) \quad 5.1$$

En 5.1 se resalta la necesidad de que el plazo máximo sea proporcional al tiempo de ejecución relativo a su tiempo de arribo. La constante de proporcionalidad que se puede introducir en la expresión es el grado de multiprogramación n que para este caso es 2 como se ve a continuación en 5.2 y en 5.3:

$$D_{\max_i} = k(L_i + C_i) \quad 5.2$$

$$D_{\max_i} = n(L_i + C_i) \quad 5.3$$

Con estas ecuaciones se vuelven a hacer las mismas pruebas y se muestran los resultados en la Tabla 5.2.

Muestra		L	$D_{\min}$	C	$D_{\max}$	Funciona	Dmax promedio	Dmax requerido	Funciona	Dmax requerido promedio
1	T1	5	21	11	35	Si	32	32	Si	36
	T2	11	20	9	29			40		
2	T1	8	40	30	70	Si	72.5	76	Si	90
	T2	20	42	32	75			104		
3	T1	15	40	30	70	Si	72.5	90	Si	99
	T2	22	42	32	75			108		
4	T1	15	40	30	92	Si	86	90	Si	99
	T2	22	42	32	80			108		
5	T1	21	40	30	92	Si	86	102	Si	112
	T2	29	42	32	80			122		
6	T1	28	40	30	92	Si	86	116	Si	119
	T2	29	42	32	80			122		
7	T1	28	40	30	92	Si	86	116	Si	121
	T2	31	42	32	80			126		
8	T1	30	40	30	92	No	86	120	Si	123
	T2	31	42	32	80			126		
9	T1	29	40	30	92	Si	86	118	Si	122

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

	T2	31	42	32	80			126		
10	T1	29	80	56	105	No	101.5	170	Si	156
	T2	31	64	40	98			142		
11	T1	29	80	30	105	Si	101.5	118	Si	122
	T2	31	64	32	98			126		
12	T1	29	100	30	105	Si	101.5	118	Si	122
	T2	31	80	32	98			126		
13	T1	17	35	7	40	Si	49	48	Si	67
	T2	21	40	22	58			86		
14	T1	17	35	17	40	No	49	68	Si	77
	T2	21	40	22	58			86		
15	T1	16	35	17	40	No	49	66	Si	76
	T2	21	40	22	58			86		
16	T1	10	35	17	40	Si	49	54	Si	70
	T2	21	40	22	58			86		
17	T1	10	37	17	40	Si	49	54	Si	70
	T2	21	40	22	58			86		
18	T1	14	35	17	40	No	49	62	Si	74
	T2	21	40	22	58			86		
19	T1	12	35	17	40	Si	49	58	Si	72
	T2	21	40	22	58			86		
20	T1	12	35	3	40	Si	49	30	Si	41
	T2	21	40	5	58			52		
21	T1	12	35	13	40	No	49	50	Si	71
	T2	21	40	25	58			92		
22	T1	12	30	13	40	Si	49	50	Si	71
	T2	21	55	25	58			92		
23	T1	12	30	13	40	Si	49	50	Si	71
	T2	21	55	25	58			92		
24	T1	12	30	16	40	Si	49	56	Si	75
	T2	21	55	26	58			94		
25	T1	12	30	20	40	No	49	64	Si	81
	T2	21	55	28	58			98		
26	T1	12	30	20	40	No	49	64	Si	77
	T2	21	55	24	58			90		
27	T1	12	30	16	40	No	49	56	Si	77
	T2	21	55	28	58			98		
28	T1	12	30	16	40	Si	49	56	Si	74
	T2	21	55	25	58			92		
29	T1	12	30	16	40	Si	49	56	Si	75
	T2	21	55	26	58			94		
30	T1	12	30	16	40	No	49	56	Si	76
	T2	21	55	27	58			96		
31	T1	12	30	17	40	Si	49	58	Si	76
	T2	21	55	26	58			94		
32	T1	12	30	18	40	Si	49	60	Si	77

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

33	T2	21	55	26	58	Si	49	94	Si	78
	T1	12	30	19	40					
	T2	21	55	26	58					
34	T1	12	35	20	40	No	49	64	Si	79
	T2	21	55	26	58					

Tabla 5.2 Pruebas con el Nuevo Plazo Máximo.

Como se puede ver en la Tabla 5.2 existen pruebas en el simulador que ya funcionaban antes del último cálculo, sin embargo con la necesidad de uniformar los resultados se requiere que todas las pruebas que ya se hicieron y las subsecuentes tengan este parámetro de seguridad y que se puede ver reflejado en los promedios de cada una. Por ejemplo, en una prueba futura se pueden tener los siguientes parámetros mostrados en la Tabla 5.3:

Muestra		<i>L</i>	<i>Dmin</i>	<i>C</i>	<i>Dmax</i>
35	T1	22	35	40	99
	T2	35	55	36	90

Tabla 5.3 Nuevos Parámetro para una Reciente Simulación.

Con los parámetros que se muestran en la Tabla 5.3 no se da la seguridad que se puedan planificar correctamente ambas tareas. Para tener una seguridad de la planificación se requieren plazos como los que se muestran en la Tabla 5.4.

Muestra		<i>L</i>	<i>Dmin</i>	<i>C</i>	<i>Dmax</i>	Funciona	Dmax promedio	Dmax requerido	Funciona	Dmax requerido promedio
35	T1	22	35	40	99	¿?	94.5	124	Si	133
	T2	35	55	36	90			142		

Tabla 5.4 Prueba de Plazo Máximo Requerido.

5.4 Conclusiones del capítulo 5.

Cuando se planifican dos tareas con prioridades dinámicas es importante tomar en cuenta la relación que hay entre los parámetros de cada tarea así como la relación que puede existir entre los parámetros de las dos tareas. En el caso de dos tareas que tienen cada una un tiempo de arribo, un tiempo de ejecución, un plazo mínimo y uno máximo, los parámetros que corresponden a los plazos deben de ser calculados de acuerdo al tiempo de arribo así como al tiempo de ejecución y al grado de multiprogramación existente que es el número de tareas que están actuando en ese momento en el sistema.

CAPÍTULO 6

PLANIFICACION EN TIEMPO REAL USANDO EL ALGORITMO FIFO PARA TTR PERIÓDICAS Y ESPORÁDICAS.

6.1 Introducción.

En este capítulo se usa la teoría del planificador First In First Out (FIFO), considerando las principales características de este para exponer un algoritmo simulado que sirva como referencia para sistemas digitales que lo requieran.

6.2 Algoritmo de Planificación FIFO.

Este algoritmo también llamado FCFS (First Come First Serve) [Lib1] es empleado para atender tareas aperiódicas de baja prioridad que se presentan cuando un conjunto de tareas periódicas de alta prioridad están siendo ejecutadas.

El orden de atención a las tareas periódicas es de acuerdo al siguiente criterio: la primera tarea que llega es la primera que se atiende. Si llega una tarea aperiódica en el momento que se esta ejecutando una tarea periódica la aperiódica tiene que esperar a que la tarea periódica de alta prioridad termine su ejecución para poder después ser ejecutada [Lib2].

Una tarea aperiódica es esencialmente un evento aleatorio y es activada usualmente por una acción externa al sistema [Art15]. En general son eventos que siguen una distribución tal como la de Poisson y que no pueden tener plazos del tipo Hard. Cuando se puede manejar este tipo de tareas con plazos del tipo Hard se les llama esporádicas [Art15].

Para el algoritmo de planificación FIFO, las tareas aperiódicas se tomaran como esporádicas tomando en cuenta las situaciones mencionadas mas arriba. Por lo mismo la caracterización de las tareas esporádicas es la misma que se menciona en capítulos anteriores.

Las tareas periódicas del algoritmo FIFO cuentan con un periodo de ejecución y un tiempo de ejecución [Art1] [Art15]. Para acoplar la caracterización de las tareas esporádicas se propone que el periodo de ejecución es el resultado de la diferencia entre el plazo máximo $D_{\max_i}$ y el tiempo de arribo L_i , es decir:

Definición 23. (Periodo de ejecución).

El periodo de ejecución de una tarea en tiempo real periódica puede expresarse como:

$$T_i = D_{\max_i} - L_i \quad 6.1$$

En lo que respecta al plazo mínimo se puede decir que es un parámetro que no afecta al cálculo de la tarea periódica si lo tiene o si no lo tiene, sin embargo, con el fin de estandarizar las tareas en tiempo real se considera y se toma en cuenta de la siguiente manera: si una tarea periódica que se esta ejecutando esta a punto de terminar su ejecución y no ha cumplido con el plazo mínimo esta tendrá que esperar o bloquearse hasta que transcurra el tiempo que necesita para cumplir este plazo. En este tiempo, si existe una tarea esporádica en espera se puede permitir que se ejecute, hasta que el tiempo del plazo mínimo de la tarea periódica se cumpla y esta pueda ejecutarse. Entonces, la tarea esporádica se bloquea y tendrá que esperar hasta que la tarea periódica se termine de ejecutar. Una vez que la tarea periódica termine se le permite ejecutar a la tarea esporádica.

Tomando los parámetros de las tareas en ejemplo del Capitulo 4, se puede ver como actúa este algoritmo. Recordando los parámetros de este ejemplo se tiene una tarea 1 mostrada en la Tabla 6.1 y una tarea 2 mostrada en la Tabla 6.2.

$L=5u$
$D_{\min}=28u$
$D_{\max}=35u$
$C=11u$

Tabla 6.1 Parámetros para la Tarea 1 Usados en la Simulación FIFO.

$L=11u$
$D_{\min}=20u$
$D_{\max}=29u$
$C=9u$

Tabla 6.2 Parámetros para la Tarea 2 Usados en la Simulación FIFO.

En la Figura 6.1 se muestra como la tarea empieza a ejecutarse desde su tiempo de arribo. En esta misma figura se ve como llega al tiempo 15u y solo falta 1u de

tiempo de ejecución para que quede completa. Es en ese instante la tarea periódica se bloquea y permite que alguna tarea esporádica que halla arribado en un tiempo anterior o igual a ese se ejecute. En este caso una tarea esporádica ha arribado en el tiempo 11u por lo que esta se puede ejecutar en tiempo igual a 15u, como se muestra en la Figura 6.2.

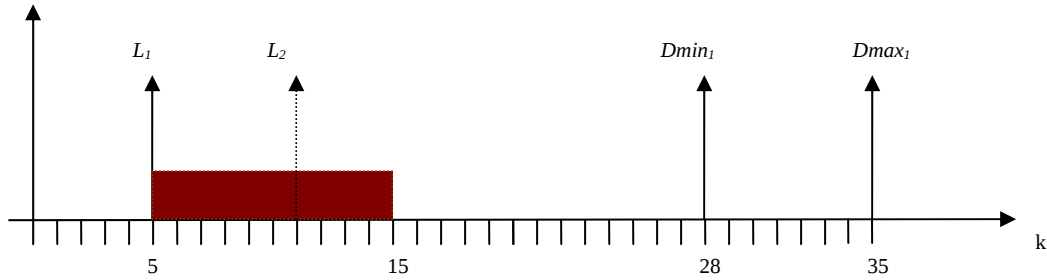


Figura 6.1 La Tarea 1 Empieza a Ejecutarse Desde su Tiempo de Arribo.

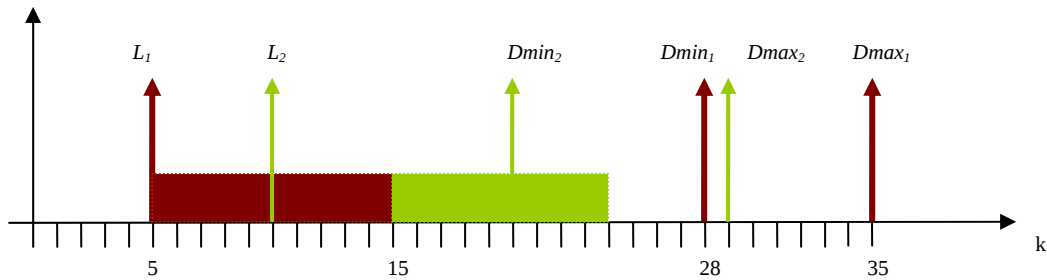


Figura 6.2 La Tarea 2 Empieza a Ejecutarse Desde su Tiempo de Arribo.

En la Figura 6.2 se observa que la tarea 2 esporádica termina su ejecución dentro de los límites que marcan sus plazos y antes que se cumpla el plazo mínimo de la tarea periódica, por lo que la tarea periódica tiene que esperar aun más para poder terminar su ejecución, como se muestra en la Figura 6.3.

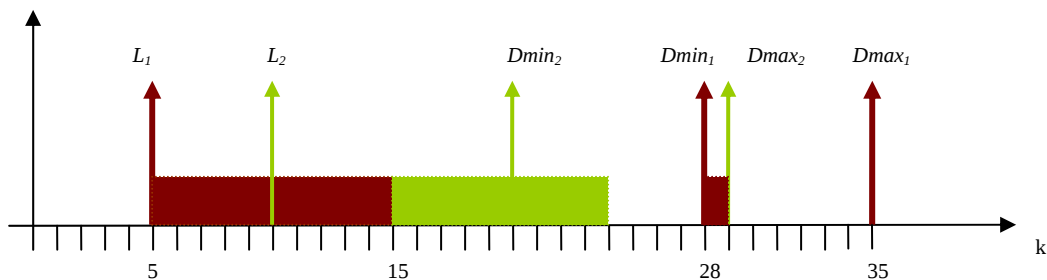


Figura 6.3 Ejemplo 1 de Ejecución de las Tareas 1 y 2.

A continuación se muestra un ejemplo 2 con los parámetros mostrados en la Tabla 6.3 para una tarea periódica y en la Tabla 6.4 para una tarea esporádica.

$L=3u$
$D_{\min}=30u$
$D_{\max}=40u$
$C=22u$

Tabla 6.3 Parámetros de la Tarea 1 Para Ejemplo 2.

$L=11u$
$D_{\min}=22u$
$D_{\max}=45u$
$C=12u$

Tabla 6.4 Parámetros de la Tarea 2 Para Ejemplo 2.

La Figura 6.4, muestra el desenvolvimiento de ambas tareas de acuerdo a sus plazos y características.

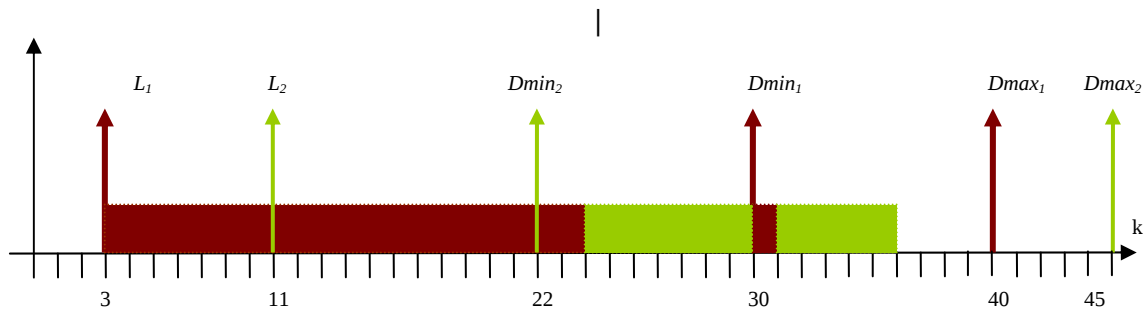


Figura 6.4 Desenvolvimiento Grafico del Ejemplo 2.

Como se ve en la Figura 6.4 la tarea periódica tiene que esperarse para poder terminar. Cuando esto sucede se le permite a la tarea esporádica ejecutarse hasta que la tarea periódica llegue al tiempo adecuado de plazo mínimo para terminar de ejecutarse. En el momento en el que este plazo llega para la tarea periódica y comienza de nuevo a ejecutarse, ahora la tarea esporádica es la que se tiene que esperar a que termine la tarea periódica. Luego de esto la tarea esporádica se termina de ejecutar.

A continuación se presenta otro ejemplo donde se tienen las tareas periódica y esporádica con los parámetros mostrados en las tablas Tabla 6.5 y Tabla 6.6 respectivamente.

$L=3u$
$D_{\min}=23u$
$D_{\max}=35u$
$C=22u$

Tabla 6.5 Parámetros de la Tarea 1 Para Ejemplo 3.

$L=11u$
$D_{\min}=22u$
$D_{\max}=45u$
$C=12u$

Tabla 6.6 Parámetros de la Tarea 2 Para Ejemplo 3.

En la Figura 6.5 se muestra la evolución de ambas tareas dados los parámetros en la Tabla 6.5 y Tabla 6.6.

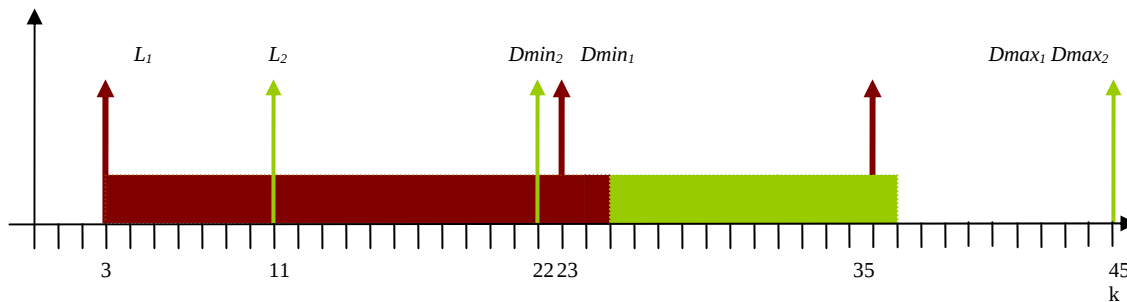


Figura 6.5 Evolución del Ejemplo 3.

Como se puede ver en la Figura 6.5 la tarea periódica al no tener una barrera en el plazo mínimo termina en un solo evento. Luego la tarea esporádica hace lo propio sin tener problemas con los plazos.

Como último ejemplo se muestran dos tareas con los siguientes parámetros mostrados en las tablas Tabla 6.7 y Tabla 6.8.

$L=3u$
$D_{\min}=23u$
$D_{\max}=40u$
$C=22u$

Tabla 6.7 Parámetros de la Tarea 1 Para Ejemplo 4.

$L=11u$
$D_{min}=22u$
$D_{max}=35u$
$C=12u$

Tabla 6.8 Parámetros de la Tarea 2 Para Ejemplo 4.

En la Figura 6.6 se muestra el accionar de la planificación. En este último ejemplo se ve que la tarea esporádica no logra terminar cumpliendo con el plazo máximo. Para evitar esta circunstancia se debe de condicionar la tarea esporádica al requisito del plazo máximo mostrado en el Capítulo 5. Con esto la Tabla 6.8 cambiaría a como se muestra en la Tabla 6.9.

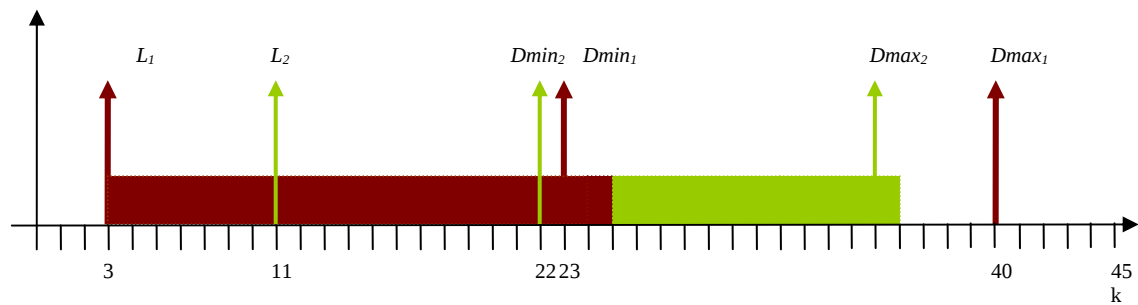


Figura 6.6 Evolución del Ejemplo 4.

$L=11u$
$D_{min}=22u$
$D_{max}=46u$
$C=12u$

Tabla 6.9 Modificación en la Tarea 2 Para Ejemplo 4.

Con esta modificación se tiene certeza que la tarea esporádica se completará en tiempo de acuerdo a sus plazos.

6.3 Conclusiones del Capítulo 6.

En este capítulo se muestra la forma de planificar 2 tareas, una periódica y otra esporádica, de acuerdo al algoritmo de planificación FIFO. De igual manera que sucedió en el capítulo anterior, surge la necesidad de emplear un recurso que permite tener cierta seguridad en el accionar de las tareas esporádicas al restringir el plazo máximo a un mínimo suficiente para que se pueda cumplir con dicho plazo.

Además se muestran diferentes casos que se pueden presentar al utilizar este algoritmo de planificación.

CAPÍTULO 7

SIMULACIÓN DEL ALGORITMO FIFO CON TTR PERIÓDICAS Y ESPORÁDICAS.

7.1 Introducción.

En este capítulo se usa la teoría del planificador First In First Out (FIFO), considerando las principales características expuestas en el capítulo anterior, para exponer un algoritmo simulado que sirva como referencia para sistemas digitales que lo requieran.

7.2 De la Teoría a la práctica. El simulador del Algoritmo FIFO.

El algoritmo de planificación que se muestra a continuación resuelve el problema de planificar 2 tareas con los parámetros mostrados en la Tabla 7.1.

Tarea 1	Tarea 2
$L=5u$	$L=11u$
$D_{\min}=28u$	$D_{\min}=20u$
$D_{\max}=35u$	$D_{\max}=29u$
$C=11u$	$C=9u$

Tabla 7.1 Parámetros para la Simulación del Algoritmo FIFO.

En la Figura 7.1 se muestra el algoritmo que se usa para la simulación en MATLAB.

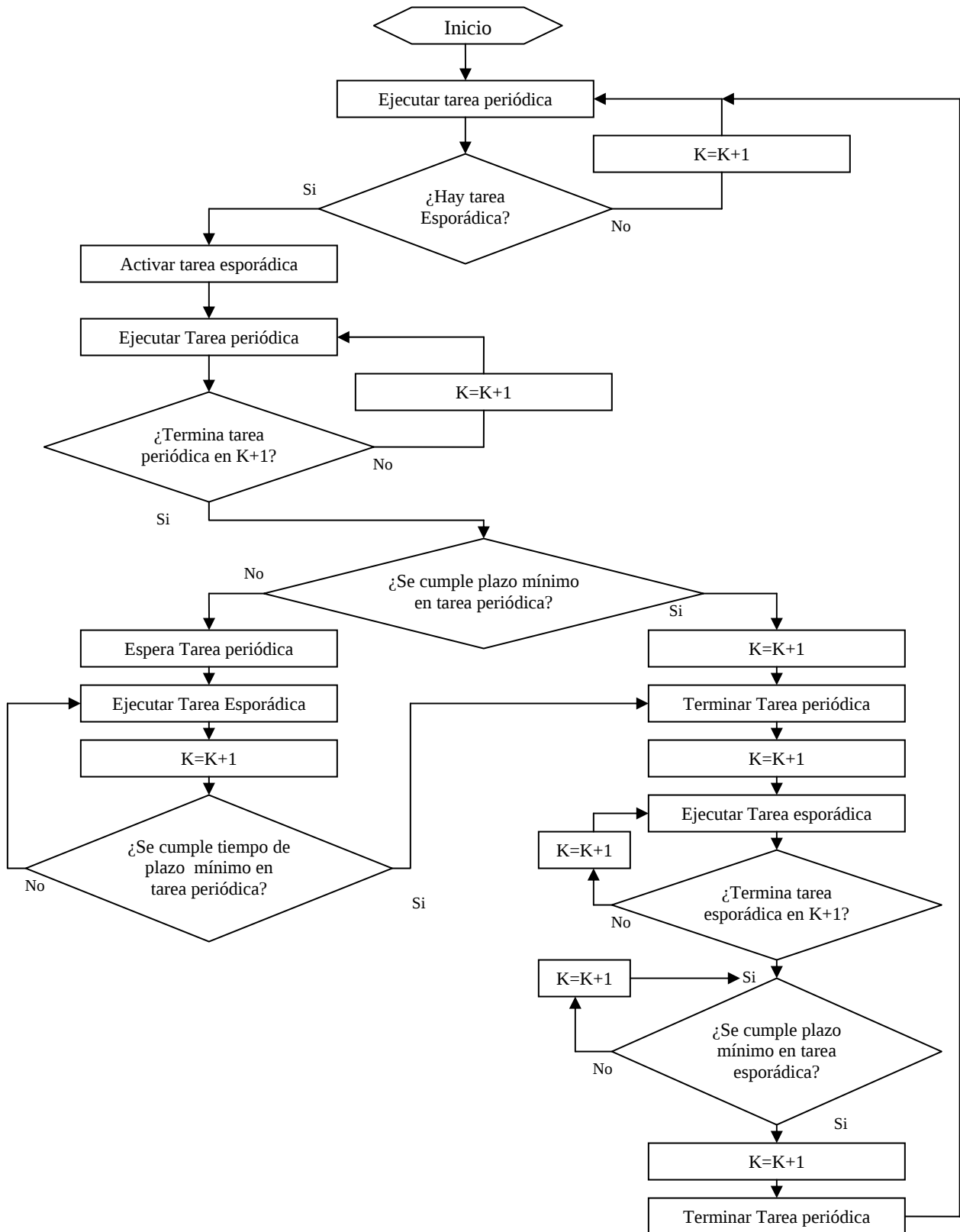


Figura 7.1 Diagrama de Flujo Para el Simulador del Algoritmo FIFO.

En la Figura 7.1 se muestra el desarrollo de las dos tareas, la periódica como la esporádica. En esta Figura 7.1 también se puede observar que debido a las condiciones de plazos que se marcan en el capítulo anterior se pueden cumplir las ejecuciones de las tareas dentro de los plazos.

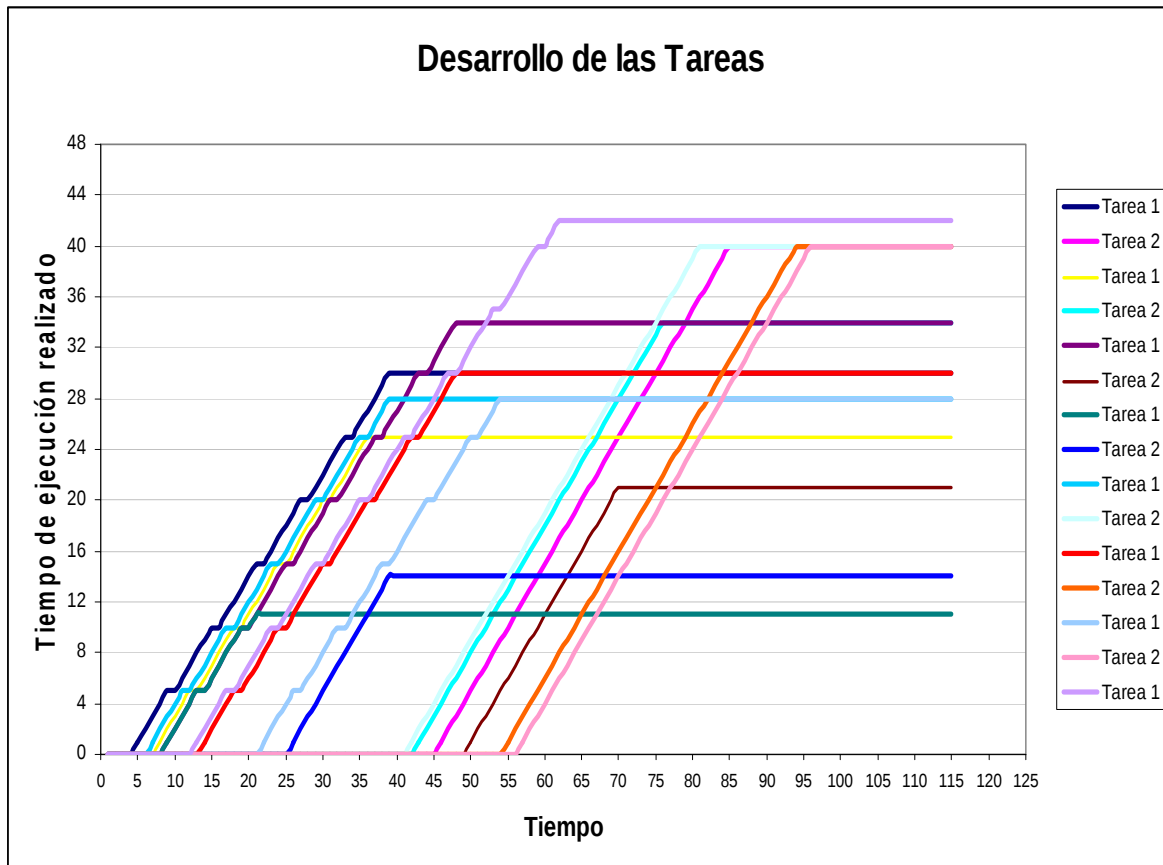


Figura 7.2 Cumplimiento de Plazos Para Varias Muestras en la Simulación.

En las Tablas 7.1a, 7.1b, 7.1c, 7.1d y 7.1e se muestran los resultados de la simulación con diferentes entradas tanto en la tarea periódica como en la tarea esporádica las cuales están referenciadas como tarea 1 y tarea 2 respectivamente.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

	L1=5	L2=20	L1=8	L2=13	L1=9	L2=15	L1=9	L2=15
	Dmin1=20	Dmin2=40	Dmin1=22	Dmin2=23	Dmin1=22	Dmin2=23	Dmin1=22	Dmin2=23
	Dmax1=110	Dmax2=120	Dmax1=66	Dmax2=110	Dmax1=90	Dmax2=72	Dmax1=40	Dmax2=60
	C1=30	C2=40	C1=25	C2=34	C1=34	C2=21	C1=11	C2=14
	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0
7	3	0	0	0	0	0	0	0
8	4	0	1	0	0	0	0	0
9	5	0	2	0	1	0	1	0
10	5	0	3	0	2	0	2	0
11	6	0	4	0	3	0	3	0
12	7	0	5	0	4	0	4	0
13	8	0	5	0	5	0	5	0
14	9	0	6	0	5	0	5	0
15	10	0	7	0	6	0	6	0
16	10	0	8	0	7	0	7	0
17	11	0	9	0	8	0	8	0
18	12	0	10	0	9	0	9	0
19	13	0	10	0	10	0	10	0
20	14	0	11	0	10	0	10	0
21	15	0	12	0	11	0	11	0
22	15	0	13	0	12	0	11	0
23	16	0	14	0	13	0	11	0
24	17	0	15	0	14	0	11	0
25	18	0	15	0	15	0	11	0
26	19	0	16	0	15	0	11	1
27	20	0	17	0	16	0	11	2
28	20	0	18	0	17	0	11	3
29	21	0	19	0	18	0	11	4
30	22	0	20	0	19	0	11	5
31	23	0	20	0	20	0	11	6
32	24	0	21	0	20	0	11	7
33	25	0	22	0	21	0	11	8
34	25	0	23	0	22	0	11	9
35	26	0	24	0	23	0	11	10
36	27	0	25	0	24	0	11	11
37	28	0	25	0	25	0	11	12
38	29	0	25	0	25	0	11	13
39	30	0	25	0	26	0	11	14
40	30	0	25	0	27	0		
41	30	0	25	0	28	0		
42	30	0	25	0	29	0		
43	30	0	25	1	30	0		
44	30	0	25	2	30	0		
45	30	0	25	3	31	0		
46	30	1	25	4	32	0		
47	30	2	25	5	33	0		
48	30	3	25	6	34	0		
49	30	4	25	7	34	0		

Tabla 7.1a Resultados de la Simulación FIFO con Diferentes Entradas.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

50	30	5	25	8	34	1		
51	30	6	25	9	34	2		
52	30	7	25	10	34	3		
53	30	8	25	11	34	4		
54	30	9	25	12	34	5		
55	30	10	25	13	34	6		
56	30	11	25	14	34	7		
57	30	12	25	15	34	8		
58	30	13	25	16	34	9		
59	30	14	25	17	34	10		
60	30	15	25	18	34	11		
61	30	16	25	19	34	12		
62	30	17	25	20	34	13		
63	30	18	25	21	34	14		
64	30	19	25	22	34	15		
65	30	20	25	23	34	16		
66	30	21	25	24	34	17		
67	30	22	25	25	34	18		
68	30	23	25	26	34	19		
69	30	24	25	27	34	20		
70	30	25	25	28	34	21		
71	30	26	25	29				
72	30	27	25	30				
73	30	28	25	31				
74	30	29	25	32				
75	30	30	25	33				
76	30	31	25	34				
77	30	32						
78	30	33						
79	30	34						
80	30	35						
81	30	36						
82	30	37						
83	30	38						
84	30	39						
85	30	40						
86								
87								
88								
89								
90								
91								
92								
93								
94								
95								
96								
97								
98								
99								
100								
101								
102								
103								

Tabla 7.1b Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

	L1=7	L2=20	L1=14	L2=20	L1=22	L2=28	L1=13	L2=28
	Dmin1=22	Dmin2=23	Dmin1=34	Dmin2=23	Dmin1=60	Dmin2=50	Dmin1=32	Dmin2=34
	Dmax1=70	Dmax2=100	Dmax1=90	Dmax2=100	Dmax1=100	Dmax2=130	Dmax1=110	Dmax2=130
	C1=28	C2=40	C1=30	C2=40	C1=28	C2=40	C1=42	C2=50
	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	2	0	0	0	0	0	0	0
9	3	0	0	0	0	0	0	0
10	4	0	0	0	0	0	0	0
11	5	0	0	0	0	0	0	0
12	5	0	0	0	0	0	0	0
13	6	0	0	0	0	0	1	0
14	7	0	1	0	0	0	2	0
15	8	0	2	0	0	0	3	0
16	9	0	3	0	0	0	4	0
17	10	0	4	0	0	0	5	0
18	10	0	5	0	0	0	5	0
19	11	0	5	0	0	0	6	0
20	12	0	6	0	0	0	7	0
21	13	0	7	0	0	0	8	0
22	14	0	8	0	1	0	9	0
23	15	0	9	0	2	0	10	0
24	15	0	10	0	3	0	10	0
25	16	0	10	0	4	0	11	0
26	17	0	11	0	5	0	12	0
27	18	0	12	0	5	0	13	0
28	19	0	13	0	6	0	14	0
29	20	0	14	0	7	0	15	0
30	20	0	15	0	8	0	15	0
31	21	0	15	0	9	0	16	0
32	22	0	16	0	10	0	17	0
33	23	0	17	0	10	0	18	0
34	24	0	18	0	11	0	19	0
35	25	0	19	0	12	0	20	0
36	25	0	20	0	13	0	20	0
37	26	0	20	0	14	0	21	0
38	27	0	21	0	15	0	22	0
39	28	0	22	0	15	0	23	0
40	28	0	23	0	16	0	24	0
41	28	0	24	0	17	0	25	0
42	28	1	25	0	18	0	25	0
43	28	2	25	0	19	0	26	0
44	28	3	26	0	20	0	27	0
45	28	4	27	0	20	0	28	0
46	28	5	28	0	21	0	29	0
47	28	6	29	0	22	0	30	0
48	28	7	30	0	23	0	30	0
49	28	8	30	0	24	0	31	0

Tabla 7.1c Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

50	28	9	30	0	25	0	32	0
51	28	10	30	0	25	0	33	0
52	28	11	30	0	26	0	34	0
53	28	12	30	0	27	0	35	0
54	28	13	30	0	28	0	35	0
55	28	14	30	1	28	0	36	0
56	28	15	30	2	28	0	37	0
57	28	16	30	3	28	1	38	0
58	28	17	30	4	28	2	39	0
59	28	18	30	5	28	3	40	0
60	28	19	30	6	28	4	40	0
61	28	20	30	7	28	5	41	0
62	28	21	30	8	28	6	42	0
63	28	22	30	9	28	7	42	0
64	28	23	30	10	28	8	42	0
65	28	24	30	11	28	9	42	0
66	28	25	30	12	28	10	42	1
67	28	26	30	13	28	11	42	2
68	28	27	30	14	28	12	42	3
69	28	28	30	15	28	13	42	4
70	28	29	30	16	28	14	42	5
71	28	30	30	17	28	15	42	6
72	28	31	30	18	28	16	42	7
73	28	32	30	19	28	17	42	8
74	28	33	30	20	28	18	42	9
75	28	34	30	21	28	19	42	10
76	28	35	30	22	28	20	42	11
77	28	36	30	23	28	21	42	12
78	28	37	30	24	28	22	42	13
79	28	38	30	25	28	23	42	14
80	28	39	30	26	28	24	42	15
81	28	40	30	27	28	25	42	16
82			30	28	28	26	42	17
83			30	29	28	27	42	18
84			30	30	28	28	42	19
85			30	31	28	29	42	20
86			30	32	28	30	42	21
87			30	33	28	31	42	22
88			30	34	28	32	42	23
89			30	35	28	33	42	24
90			30	36	28	34	42	25
91			30	37	28	35	42	26
92			30	38	28	36	42	27
93			30	39	28	37	42	28
94			30	40	28	38	42	29
95					28	39	42	30
96					28	40	42	31
97							42	32
98							42	33
99							42	34
100							42	35
101							42	36
102							42	37
103							42	38

Tabla 7.1d Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).

104							42	39
105							42	40
106							42	41
107							42	42
108							42	43
109							42	44
110							42	45
111							42	46
112							42	47
113							42	48
114							42	49
115							42	50

Tabla 7.1e Resultados de la Simulación FIFO con Diferentes Entradas (Continuación).

7.3 Conclusiones del Capítulo 7.

En primer término se puede constatar que el algoritmo funciona siempre y cuando se respeten las características impuestas en los capítulos anteriores y que marcan la pauta para poder definir de manera factible los límites de los plazos que cada tarea debe de tener y que son resultados importantes para el diseño de las tareas que están caracterizadas de esta manera.

CAPÍTULO 8

ANÁLISIS DE PLANIFICABILIDAD DE LOS ALGORITMOS EDF Y FIFO.

8.1 Introducción.

En este capítulo se hacen mediciones de utilización del procesador mediante el factor de utilización del procesador. También se consideran parámetros para medir la planificabilidad de los algoritmos EDF y FIFO usando las mediciones de los capítulos anteriores.

8.2 Factor de utilización y planificabilidad.

Un planificador es un método para asignar recursos (el tiempo de procesador) [Int0]. Un conjunto de tareas es factible si existe algún planificador que sea capaz de cumplir las restricciones de todas las tareas (en nuestro caso las restricciones temporales: los plazos de ejecución). Un planificador es óptimo si es capaz de planificar correctamente cualquier conjunto de tareas factible.

El factor de utilización [Art1] se escribe con la Ecuación 8.1:

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \quad 8.1$$

Donde:

C_i es el tiempo de ejecución o tiempo de cálculo o tiempo de cómputo de la tarea i .
 P_i es el periodo de ejecución de la tarea i .

En la Ecuación 8.1 se hace un cálculo del uso del procesador por un conjunto de tareas en Tiempo Real. La Ecuación 8.1 se usa para calcular el factor de utilización de las pruebas realizadas en los capítulos anteriores con los algoritmos de planificación EDF y FIFO y que más adelante se muestran.

Para un conjunto de dos tareas con asignación de prioridades arregladas, el límite más alto para el factor de utilización [Art1] se muestra en la Ecuación 8.2:

$$U = 2(2^{1/2} - 1) \quad 8.2$$

La restricción de planificabilidad [Art1] esta dada por 8.3.

$$\sum_{i=1}^n \frac{C_i}{P_i} \leq 1 \quad 8.3$$

Con la Ecuación 8.3 se puede decir que un sistema es planificable si el factor de utilización U es menor o igual a uno. Además para un conjunto de tareas de dos el factor de utilización debe ser al menos igual a 0.828427. Esto es, tomando en cuenta 8.2 y 8.3, un conjunto de dos Tareas en Tiempo Real, cualesquiera que estas sean, es planificable si cumple con los límites mostrados en 8.4 y en 8.5 y que corresponden al cálculo del factor de uso de procesador para un conjunto de dos TTR.

$$2(2^{1/2} - 1) \leq \sum_{i=1}^n \frac{C_i}{P_i} \leq 1 \quad 8.4$$

$$0.828427 \leq U \leq 1 \quad 8.5$$

8.3 Resultados del Factor de Utilización y Planificabilidad con los algoritmos usados en los Capítulos 4, 5, 6 y 7.

En la Figura 8.1 se muestran los cálculos de los parámetros de muestras para las simulaciones del algoritmo EDF tomando en cuenta las ecuaciones de los capítulos anteriores.

En la Figura 8.2 se muestran los resultados de las simulaciones para el algoritmo EDF. La columna llamada P_{maximo} es el periodo completo de ejecución calculado como con 8.6.

$$P_{maximo_i} = D_{max_i} - L_i \quad 8.6$$

Este es el máximo periodo que puede tener la tarea i y que esta limitado por su tiempo de arribo y por el máximo plazo que debe de cumplir.

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

1		2		3		4		5	
L1=5	L2=11	L1=8	L2=20	L1=15	L2=22	L1=15	L2=22	L1=21	L2=29
Dmin1=21	Dmin2=20	Dmin1=40	Dmin2=42	Dmin1=40	Dmin2=42	Dmin1=40	Dmin2=42	Dmin1=40	Dmin2=42
Dmax1=35	Dmax2=29	Dmax1=70	Dmax2=75	Dmax1=70	Dmax2=80	Dmax1=92	Dmax2=80	Dmax1=92	Dmax2=80
C1=11	C2=9	C1=30	C2=32	C1=30	C2=32	C1=30	C2=32	C1=30	C2=32
P=30	P=18	P=22	P=55	P=55	P=58	P=77	P=58	P=71	P=51
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
20	10	58	48	52	49	62	53	62	50

6		7		8		9		10	
L1=28	L2=29	L1=28	L2=31	L1=29	L2=31	L1=29	L2=31	L1=29	L2=31
Dmin1=40	Dmin2=42	Dmin1=40	Dmin2=42	Dmin1=80	Dmin2=64	Dmin1=80	Dmin2=64	Dmin1=100	Dmin2=80
Dmax1=92	Dmax2=80	Dmax1=92	Dmax2=80	Dmax1=105	Dmax2=80	Dmax1=105	Dmax2=98	Dmax1=105	Dmax2=98
C1=30	C2=32	C1=30	C2=32	C1=30	C2=32	C1=30	C2=32	C1=30	C2=32
P=64	P=51	P=64	P=49	P=76	P=49	P=76	P=67	P=76	P=67
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
62	50	61	47	61	46	61	54	71	55

11		12		13		14		15	
L1=17	L2=21	L1=10	L2=21	L1=10	L2=21	L1=12	L2=21	L1=12	L2=21
Dmin1=35	Dmin2=40	Dmin1=37	Dmin2=40	Dmin1=35	Dmin2=40	Dmin1=35	Dmin2=40	Dmin1=35	Dmin2=40
Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58
C1=7	C2=22	C1=17	C2=22	C1=17	C2=22	C1=17	C2=22	C1=3	C2=5
P=23	P=37	P=30	P=37	P=30	P=37	P=28	P=37	P=28	P=37
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
21	25	28	27	28	27	24	26	23	20

16		17		18		19		20	
L1=12	L2=21	L1=12	L2=21	L1=12	L2=21	L1=12	L2=21	L1=12	L2=21
Dmin1=30	Dmin2=55	Dmin1=30	Dmin2=55	Dmin1=30	Dmin2=55	Dmin1=30	Dmin2=55	Dmin1=30	Dmin2=55
Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58
C1=13	C2=25	C1=16	C2=26	C1=16	C2=25	C1=17	C2=26	C1=16	C2=26
P=28	P=37	P=28	P=37	P=28	P=37	P=28	P=37	P=28	P=37
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
27	28	29	34	27	34	29	34	29	34

21		22	
L1=12	L2=21	L1=12	L2=21
Dmin1=30	Dmin2=55	Dmin1=30	Dmin2=55
Dmax1=40	Dmax2=58	Dmax1=40	Dmax2=58
C1=18	C2=26	C1=19	C2=26
P=28	P=37	P=28	P=37
Tarea 1	Tarea 2	Tarea 1	Tarea 2
29	34	29	35

Figura 8.1 Cálculo de los Parámetros de Muestras para las Simulaciones del Algoritmo EDF.

La columna llamada $U1$ es el factor de utilización usando los periodos P_{maximo} para cada par de tareas con 8.7.

$$U1 = \frac{C_1 C_2}{P_{\text{maximo}_1} P_{\text{maximo}_2}} \quad 8.7$$

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real

Muestra		L	Dmin	C	Dmax	Pmáximo	U1	Preal	U2
1	T1	5	21	11	35	30	0.416667	20	0.666667
	T2	11	20	9	29	18		10	
2	T1	8	40	30	70	62	0.529915	58	0.584906
	T2	20	42	32	75	55		48	
3	T1	15	40	30	70	55	0.574074	52	0.613861
	T2	22	42	32	75	53		49	
4	T1	15	40	30	92	77	0.459259	62	0.53913
	T2	22	42	32	80	58		53	
5	T1	21	40	30	92	71	0.508197	62	0.553571
	T2	29	42	32	80	51		50	
6	T1	28	40	30	92	64	0.53913	62	0.553571
	T2	29	42	32	80	51		50	
7	T1	28	40	30	92	64	0.548673	61	0.574074
	T2	31	42	32	80	49		47	
8	T1	29	40	30	92	63	0.553571	61	0.579439
	T2	31	42	32	80	49		46	
9	T1	29	80	30	105	76	0.433566	61	0.53913
	T2	31	64	32	98	67		54	
10	T1	29	100	30	105	76	0.433566	71	0.492063
	T2	31	80	32	98	67		55	
11	T1	17	35	7	40	23	0.483333	21	0.630435
	T2	21	40	22	58	37		25	
12	T1	10	35	17	40	30	0.58209	28	0.709091
	T2	21	40	22	58	37		27	
13	T1	10	37	17	40	30	0.58209	28	0.709091
	T2	21	40	22	58	37		27	
14	T1	12	35	17	40	28	0.6	24	0.78
	T2	21	40	22	58	37		26	
15	T1	12	35	3	40	28	0.123077	23	0.186047
	T2	21	40	5	58	37		20	
16	T1	12	30	13	40	28	0.584615	27	0.690909
	T2	21	55	25	58	37		28	
17	T1	12	30	16	40	28	0.646154	29	0.666667
	T2	21	55	26	58	37		34	
18	T1	12	30	16	40	28	0.630769	27	0.672131
	T2	21	55	25	58	37		34	
19	T1	12	30	16	40	28	0.646154	29	0.666667
	T2	21	55	26	58	37		34	
20	T1	12	30	17	40	28	0.661538	29	0.68254
	T2	21	55	26	58	37		34	
21	T1	12	30	18	40	28	0.676923	29	0.698413
	T2	21	55	26	58	37		34	
22	T1	12	30	19	40	28	0.692308	29	0.703125
	T2	21	55	26	58	37		35	
Promedio del Factor de Utilización							0.541167		0.613251

Figura 8.2 Resultados del Factor de Utilización en las Simulaciones para el Algoritmo EDF.

La columna llamada P_{real} es el periodo que realmente esta ocupando cada tarea desde que comienza su ejecución hasta que termina. La columna llamada $U2$ es el factor de utilización calculado a partir de los periodos que realmente se usan en cada tarea y que se esta calculado a partir de 8.8.

$$U2 = \frac{C_1 C_2}{P_{real_1} P_{real_2}} \quad 8.8$$

Al final de las columnas se muestran los promedios del factor de utilización tanto para $U1$ como para $U2$.

En la Figura 8.3 se muestran los resultados de la simulación del algoritmo FIFO. En la Figura 8.4 se muestran los resultados del factor de utilización para estos resultados.

1		2		3		4		5	
L1=5	L2=20	L1=8	L2=13	L1=9	L2=15	L1=9	L2=15	L1=7	L2=20
Dmin1=20	Dmin2=40	Dmin1=22	Dmin2=23	Dmin1=22	Dmin2=23	Dmin1=22	Dmin2=23	Dmin1=22	Dmin2=23
Dmax1=110	Dmax2=120	Dmax1=66	Dmax2=110	Dmax1=90	Dmax2=72	Dmax1=40	Dmax2=60	Dmax1=70	Dmax2=100
C1=30	C2=40	C1=25	C2=34	C1=34	C2=21	C1=11	C2=14	C1=28	C2=40
P=105	P=100	P=58	P=97	P=81	P=57	P=31	P=45	P=63	P=80
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
34	39	28	33	39	20	12	13	32	39

6		7		8	
L1=14	L2=20	L1=22	L2=28	L1=13	L2=28
Dmin1=34	Dmin2=23	Dmin1=60	Dmin2=50	Dmin1=32	Dmin2=34
Dmax1=90	Dmax2=100	Dmax1=100	Dmax2=130	Dmax1=110	Dmax2=130
C1=30	C2=40	C1=28	C2=40	C1=42	C2=50
P=76	P=80	P=78	P=102	P=97	P=102
Tarea 1	Tarea 2	Tarea 1	Tarea 2	Tarea 1	Tarea 2
34	39	32	39	49	49

Figura 8.3 Cálculo de los Parámetros de Muestras para las Simulaciones del Algoritmo FIFO.

Muestra		L	Dmin	C	Dmax	Pmáximo	U1	Preal	U2
1	T1	5	20	30	110	105	0.341463	34	0.958904
	T2	20	40	40	120	100		39	
2	T1	8	22	25	66	58	0.380645	28	0.967213
	T2	13	23	34	110	97		33	
3	T1	9	22	34	90	81	0.398551	39	0.932203
	T2	15	23	21	72	57		20	
4	T1	9	22	11	40	31	0.328947	12	1
	T2	15	23	14	60	45		13	
5	T1	7	22	28	70	63	0.475524	32	0.957746
	T2	20	23	40	100	80		39	
6	T1	14	34	30	90	76	0.448718	34	0.958904
	T2	20	23	40	100	80		39	
7	T1	22	60	28	100	78	0.377778	32	0.957746
	T2	28	50	40	130	102		39	
8	T1	13	32	42	110	97	0.462312	49	0.938776
	T2	28	34	50	130	102		49	
Promedio del Factor de Utilización							0.401742		0.958937

Figura 8.4 Resultados del Factor de Utilización en las Simulaciones para el Algoritmo FIFO.

8.4 Conclusiones del Capítulo 8.

La planificabilidad de un sistema nos dice si a un conjunto de tareas se les puede asignar recursos del sistema, como uso de procesador, de acuerdo a un algoritmo de planificación, de manera que se completen las tareas dentro de sus plazos y con cálculos correctos.

El factor de utilización es un parámetro que ayuda a identificar la planificabilidad de un sistema. En este capítulo se muestra el cálculo de dicho parámetro. La comparación con los límites propuestos para el análisis de planificabilidad pone en evidencia que el algoritmo EDF no cumple completamente con una buena planificabilidad. El algoritmo FIFO si esta acorde con los límites acorados para decir que son tareas planificables o bien que la planificabilidad es buena.

Cabe añadir que el cálculo del factor de utilización se usa para algoritmos de planificación estáticos y no tanto para dinámicos por lo mismo hace falta desarrollar métodos para medir este parámetro así como para medir la planificabilidad.

CAPÍTULO 9

CONCLUSIONES

9.1 Conclusiones.

Con las definiciones que se abordan en este primer capítulo es suficiente para darnos cuenta de la relevancia que tienen los STR, del alcance, de su aplicación solo limitada por la imaginación.

Sin hacer mucho hincapié en las aplicaciones de la electrónica se puede decir que los SD son una herramienta fundamental para el control, la implementación y la simulación de los STR. Aunque los SD han funcionado durante mucho tiempo sin ser en TR, es importante hacer notar que en nuestro mundo todos los procesos son del mundo real y que no podemos pensar que estos procesos se estén controlando por sistemas rápidos o en línea cuando deberían ser en TR. Es tiempo que la dependencia de los SD con los STR sea estrecha y mutua.

Los conceptos que se tienen que tomar en cuenta para el estudio de las computadoras que están en relación con la teoría en TR son muchos sin embargo se pueden simplificar para este trabajo en los procesos o tareas y en los planificadores que administran los recursos de CPU que solicitan estas tareas. Hasta ahora ya se tienen modelos caracterizados y bien tipificados de la tareas en tiempo real, con estos modelos se puede trabajar en simulaciones de su comportamiento en distintas circunstancias. En lo que respecta a los planificadores que se ocupan de atender a las tareas en tiempo real, existen muchos de donde elegir para hacer pruebas, sin embargo a la fecha se siguen dando mejoras de los que ya existen y otros nuevos que dan posibilidades a distintas aplicaciones.

Una de las aplicaciones más importantes, es la de cubrir la necesidad de la comunicación, y hoy en día los SD cumplen muy bien este trabajo. Sin un SO es casi imposible el manejo adecuado de un SD, como una computadora, por lo mismo su estudio y comprensión son fundamentales para un buen entendimiento de las áreas de estudio que están a su lado, por fuera y dentro del mismo.

En el análisis de las TTR se tienen algunos modelos que tienen la suficiente información para describir el comportamiento y características de las tareas. Las

características permiten que se tenga la referencia adecuada para construir simulaciones como el que en este capítulo se muestra. Con un simulador de TTR se logra un mayor entendimiento de cómo se comportan los procesos que en la teoría de TR se estudian. Esta base da la ventaja hacia el siguiente paso que es hacer que todas las tareas que van arribando a un SD se puedan atender con políticas de planificación. Por el momento una etapa más se cumple al lograr ejemplificar las tareas que en los siguientes capítulos y secciones se usarán.

En este trabajo se pudo encontrar una forma de calcular la prioridad dinámica del algoritmo EDF en base a sus propios parámetros y características. Esto ayudó a la simulación del algoritmo sin tomar en cuenta que la plataforma de simulación es la que se desarrolle. Además da pauta a que los resultados puedan utilizarse en sistemas digitales aplicados a usos específicos.

Cuando se planifican dos tareas con prioridades dinámicas es importante tomar en cuenta la relación que hay entre los parámetros de cada tarea así como la relación que puede existir entre los parámetros de las dos tareas. En el caso de dos tareas que tienen cada una un tiempo de arribo, un tiempo de ejecución, un plazo mínimo y uno máximo, los parámetros que corresponden a los plazos deben de ser calculados de acuerdo al tiempo de arribo así como al tiempo de ejecución y al grado de multiprogramación existente que es el número de tareas que están actuando en ese momento en el sistema.

En este trabajo se muestra la forma de planificar dos tareas, una periódica y otra esporádica, de acuerdo al algoritmo de planificación FIFO. Así surge la necesidad de emplear un recurso que permite tener cierta seguridad en el accionar de las tareas esporádicas al restringir el plazo máximo a un mínimo suficiente para que se pueda cumplir con dicho plazo. Además se muestran diferentes casos que se pueden presentar al utilizar este algoritmo de planificación. En primer término se puede constatar que el algoritmo funciona siempre y cuando se respeten las características impuestas en los capítulos anteriores y que marcan la pauta para poder definir de manera factible los límites de los plazos que cada tarea debe de tener y que son resultados importantes para el diseño de las tareas que están caracterizadas de esta manera.

La planificabilidad de un sistema nos dice si a un conjunto de tareas se les puede asignar recursos del sistema, como uso de procesador, de acuerdo a un algoritmo de planificación, de manera que se completen las tareas dentro de sus plazos y con cálculos correctos.

El factor de utilización es un parámetro que ayuda a identificar la planificabilidad de un sistema. En este capítulo se muestra el cálculo de dicho parámetro. La comparación con los límites propuestos para el análisis de planificabilidad pone en evidencia que el algoritmo EDF no cumple completamente con una buena planificabilidad. El algoritmo FIFO si esta acorde con los límites acorados para decir que son tareas planificables o bien que la planificabilidad es buena.

Cabe añadir que el cálculo del factor de utilización se usa para algoritmos de planificación estáticos y no tanto para dinámicos por lo mismo hace falta desarrollar métodos para medir este parámetro así como para medir la planificabilidad.

REFERENCIAS.

Libros

- Lib1 Pedro Guevara López, José de Jesús Medel Juárez.
Introducción a los Sistemas de Tiempo Real.
Ed. Politécnico. ISBN 970-36-0105-7. México. 2003.
- Lib2 Giorgio C. Buttazzo.
Hard real-time computing systems.
Scuola Superiore S. Anna, Pisa Italia. Kluwer. Academic Publishers. 1997.
- Lib3 Jane W. S. Liu
Real-time Systems.
Ed. Prentice Hall. ISBN 0-13-099651-3. USA. 2000.
- Lib4 José de Jesús Medel Juárez, Pedro Guevara López, Daniel Cruz Pérez
Temas Selectos de Sistemas en Tiempo Real.
Ed. Politécnico. México. 2006.
- Lib5 J. Stankovic y K. Ramamritham
Tutorial on Hard-Real Time Systems.
IEEE Computer Society Press. 1988
- Lib6 Leon W. Couch II
Sistemas de Comunicación Digitales y Analógicos.
México. Prentice Hall. 1998.
- Lib7 Simon Haykin.
Adaptive Filter Theory.
Prentice Hall Information an System science series. 1991.
- Lib8 Frederik Gustafsson
Adaptive Filtering and Change Detection.
Ed. John Wiley & Sons. Linkoping University Sweden. 2000.
- Lib9 Bohumil Psenicka.
Procesamiento Digital de Señales Primera Parte Filtros Digitales
UNAM DIEEC Facultad de Ingeniería. México 1994.
- Lib10 Miles J. Murdocca, Vincent P. Heuring.
Principios de Arquitectura para Computadoras.
Buenos Aires, Argentina. Pearson Educación, 2002.

- Lib11 William Stallings.
Comunicaciones y Redes de Computadores.
Prentice Hall. Madrid 2002.
- Lib12 M. Morris Mano.
Lógica Digital y Diseño de Computadores.
Prentice Hall. México 1982.
- Lib13 Barry B. Brey.
Los Microprocesadores INTEL. Arquitectura, Programación e Interfaces. 8086-80486.
Prentice Hall. 3ª Edición. México 1995.
- Lib14 Barry B. Brey.
Los Microprocesadores INTEL.
Pearson educación. 5ª Edición. 2001.
- Lib15 Eugenio Martín Cuenca, José M^a Angulo Usastegui, Ignacio Angulo Martínez.
Microcontroladores PIC, La clave del diseño.
ED. Thomson. Madrid. 2003.
- Lib16 John P. Hayes.
Introducción al Diseño Lógico Digital.
Addison-Wesley Iberoamericana. EU. 1996.
- Lib17 Andrew S. Tanenbaum.
Sistemas Operativos Modernos.
México. Pearson Educación. 2003.
- Lib18 Delores M. Etter.
Soluciones de Problemas de Ingeniería con MATLAB.
México. Prentice Hall Hispanoamericana. 1998.
- Lib19 Alan burns & Andy Wellings
Realtime Systems and Programming Languages.
Addison Wesley 1996.
- Lib20 Francisco Manuel Márquez García.
Unix. Programación Avanzada.
Addison Wesley 1993.
- Lib21 Laura Lemay, Rogers Cadenhead.
Aprendiendo Java 2 en 21 Días.
Prentice Hall. México. 1999.
- Lib22 Stan Kelly-Bootle.
Como Usar UNIX, Sistem V, version 4.0.
Editorial Limusa S.A. de C.V. Megabyte. México 1993.

Artículos

- Art1 C. L. Liu y James W. Layland.
Scheduling Algorithms for multiprogramming in Hard Real-Time Environments.
1973.
- Art2 John A. Stankovic
Real-Time Computing.
1992.
- Art3 José de Jesús Medel Juárez, Pedro Guevara López, Francisco Ríos Suriano.
Estimación de Parámetros usando los conceptos de filtros de Kalman.
CIC 2002 Vol. I, ISBN 970-18-8590-2.
- Art4 D. Stephen G. Pollock
Wiener-Kolmogorov Filtering, Frequency-Selective Filtering and Polynomial Regression.
University of London. 2005.
- Art5 José de Jesús Medel Juárez, Pedro Guevara López, Alberto Flores Rueda.
Filtro Digital Multivariable en Tiempo Real (Conceptos Básicos).
CIIIE 2003.
- Art6 Daniel Pérez Cruz, José de Jesús Medel Juárez, Pedro Guevara López.
Estimación de los Tiempos de Inicio en Tareas de Tiempo Real Periódicas con Jitter
Estocástico Utilizando la Técnica de Variable Instrumental.
- Art7 Guevara López Pedro, Medel Juárez José de Jesús, Flores Rueda Alberto.
Modelo ARMA para caracterizar tareas en Tiempo Real.
CIC-INDI 2002.
- Art8 Pedro Guevara López, José de Jesús Medel Juárez, Daniel Pérez Cruz.
Modelo Dinámico para una Tarea en Tiempo Real.
Revista Computación y Sistemas. Vol. 8 Num. 1. México. 2004.
- Art9 Pedro Guevara López, José de Jesús Medel Juárez, Daniel Pérez Cruz.
Un modelo General para los Tiempos de Arribo en Tareas de Tiempo Real.
- Art10 Kevin Jeffay, Donald F. Stanat, Charles U. Martel.
On non-preemptive scheduling of periodic and sporadic tasks.
In Proceedings of the 12 th IEEE Symposium on Real-Time Systems (December 1991)
- Art11 Pedro Guevara López, José de Jesús Medel Juárez, Daniel Pérez Cruz.
Estimadores de Parámetros en Tiempo Real y su Aplicación en Sistemas Estocásticos
con Parámetros Variantes en el Tiempo.

- Art12 Pedro Guevara López, José de Jesús Medel Juárez, Daniel Pérez Cruz.
Modelo Estocástico para Tiempos de Arribo en Tareas de Tiempo Real
- Art13 Pedro Guevara López, José de Jesús Medel Juárez, Daniel Pérez Cruz.
Modelo Recursivo para Describir los Tiempos de Arribo de las Tareas de Tiempo Real
Usando Promedios Móviles
- Art14 Alan Burns.
Scheduling Hard Real Time Systems: A Review.
- Art15 N. Audsley, A. Burns.
Real Time Systems Scheduling.
- Art16 K. Jeffay.
Scheduling Task with Shared Resources in Hard-Real-Time Systems

Páginas de Internet

- Industrial Informatics and Real-Time Systems Research Group.
Int0 <http://www.gii.upv.es/personal/iripoll/str/planificacion/html/planif.html>
- QNX Real-time Operating System
Int1 <http://www.qnx.com>
- RT-Linux
Int2 <http://www.fsmlabs.com/>,
<http://rtportal.upv.es/tutorial/01-intro/01-intro.html>.
- QNXCS On-line Manual Pages for QNX
Int3 <http://qnxcs.unomaha.edu/>
- QNX Internet Resources
Int4 <http://www.qnxzone.com/>
- VxWorks RTOS
Int5 <http://www.windriver.com/>
- TRON RTOS
Int6 <http://www.tron.org/>
- Integrity RTOS
Int7 <http://www.ghs.com/products/rtos/integrity.html>
- LinuxOS RTOS
Int8 <http://www.linuxworks.com/rtos/>
- Versatile Real-Time Executive (VRTX)
Int9 <http://www.mentor.com/>
- Nucleus RTOS
Int10 http://www.mentor.com/products/embedded_software/nucleus_rtos/index.cfm/
- Operating System Embedded OSE
Int11 <http://www.eneca.com/>
- ThreadX RTOS
Int12 <http://www.rtos.com/>
- Phoenix RTOS
Int13 <http://www.phoenix-rtos.org/>
- Windows CE
Int14 <http://msdn.microsoft.com/embedded/windowsce/default.aspx>

Wikipedia de RTOS
Int15 <http://en.wikipedia.org/wiki/RTOS>

MATLAB Web Page
Int16 <http://www.mathworks.com/>

Tesis, Informes Técnicos y Manuales

- Tes1 Daniel Cruz Pérez
“Modelo Dinámico para Tareas en Tiempo Real”. Tesis de Maestría en Ciencias de la Computación. CIC-IPN. México. 2004
- Inf1 Pedro Guevara López, José de Jesús Medel Juárez
Un vistazo a los Sistemas de Tiempo Real. Informe Técnico. IPN.
- Man1 Slackware Linux Essentials
<http://www.slackware.com/>
Second Edition
- Man2 Linux Tutorial 6.0
<http://www.linux.org/>

Programa 1. Simulador de Tareas en Tiempo Real Esporádicas

```
%Generador de Tareas en Tiempo Real Esporádicas
```

```
close all;  
clear all;
```

```
%Ventana
```

```
N=1000;  
R=0.99;  
t=1:N;
```

```
M=10;
```

```
%Variables que guardan los vectores de los parámetros
```

```
%TA Tiempo de Arribo
```

```
%Dm Plazo Mínimo
```

```
%DM Plazo Máximo
```

```
%C Carga
```

```
j=1;
```

```
for k=1:M  
    for i=1:N  
        matriz(k,i)=0;  
    end  
end
```

```
for k=1:M  
    for i=1:N  
        aleatorio=abs(rand);  
        if aleatorio>R  
            TA(i)=1;  
            TAA(j)=i;  
  
            Dm(i)=i+round(abs(rand)*(N/10));  
            DmA(j)=Dm(i);  
            DM(i)=i+round(abs(rand)*(N/10));  
            DMA(j)=DM(i);  
  
            while Dm(i)>= DM(i)  
                Dm(i)=i+round(abs(rand)*(N/10));  
                DmA(j)=Dm(i);  
                DM(i)=i+round(abs(rand)*(N/10));  
                DMA(j)=DM(i);  
            end
```

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

```
C(i)=round(abs(rand)*(N/20));
CA(j)=C(i);

while C(i)==0
    C(i)=round(abs(rand)*(N/20));
    CA(j)=C(i);
end

j=j+1;
end
if aleatorio<=R
    TA(i)=0;
    Dm(i)=0;
    DM(i)=0;
    C(i)=0;
end
matriz(k,i)=TA(i);
end
end

%Parámetros transpuesto
TAT=TAA';
CT=CA';
DMT=DMA';
DmT=DmA';
matrizT=matriz';

figure
mediasTA=mean(matrizT);
bar(mediasTA,0.25,'g');

figure
stem(t,TA)
axis([0 1000 0 1.2])
title('Tiempos de Arribo de Tareas de Tiempo Real Esporádicas')

figure
plot(t,TA)
axis([0 1000 0 1.2])
title('Tiempos de Arribo de Tareas de Tiempo Real Esporádicas')

figure
mesh(matriz)
title('Tiempos de Arribo de Tareas de Tiempo Real Esporádicas de 10 Intervalos')
```

Programa 2. Simulador del Algoritmo de planificación EDF

```

close all;
clear all;

k=1

%Banco de Tareas
%Tarea 1 Datos Iniciales
L1=12
Dmin1=30
Dmax1=40
C1=19

for i=1:(L1-1)
    Cp1(i)=0;
    Cf1(i)=C1-Cp1(i);
    TDmax1(i)=Dmax1-k;
    P1(i)=(Cf1(i)/TDmax1(i)) * ((Dmin1-L1)/(Dmax1-L1));
end
Terminada1=0;
Ejecuta1=0;
Espera1=0;
Carga1=0;

%Tarea 2 Datos iniciales
L2=21
Dmin2=55
Dmax2=58
C2=26

for i=1:(L2-1)
    Cp2(i)=0;
    Cf2(i)=C2-Cp2(i);
    TDmax2(i)=Dmax2-k;
    P2(i)=(Cf2(i)/TDmax2(i)) * ((Dmin2-L2)/(Dmax2-L2));
end
Terminada2=0;
Ejecuta2=0;
Espera2=0;
Carga2=0;

while(Terminada1==0 | Terminada2==0)
    if(k>=L1)
        Ejecuta1=1;
        if(k>=L2)
            if(P1(k-1)>P2(k-1))
                Ejecuta1=1;
                Ejecuta2=0;
            else
                Ejecuta2=1;
                Ejecuta1=0;
            end
            if(Cp2<C2)

```

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

```

    TDmax2(k)=Dmax2-k;
    Cf2(k)=C2-Cp2(k-1);
    P2(k)=(Cf2(k)/TDmax2(k)) * ((Dmin2-L2)/(Dmax2-L2));
    else
        Terminada2=1;
        Ejecuta2=0;
        P2(k)=0;
        TDmax2(k)=TDmax2(k-1);
        Cf2(k)=Cf2(k-1);
    end
end
end
if(Cp1<C1)
    TDmax1(k)=Dmax1-k;
    Cf1(k)=C1-Cp1(k-1);
    P1(k)=(Cf1(k)/TDmax1(k)) * ((Dmin1-L1)/(Dmax1-L1));
    else
        Terminada1=1;
        Ejecuta1=0;
        P1(k)=0;
        TDmax1(k)=TDmax1(k-1);
        Cf1(k)=Cf1(k-1);
    end
end
end
if((Carga1==(C1-1) & k<Dmin1)
    Espera1=1;
    Ejecuta1=0;
    if(Terminada2==0)
        Ejecuta2=1;
    end
    if(Espera2==1)
        Ejecuta2=0;
    end
    else
        Espera1=0;
    end
if((Carga2==(C2-1) & k<Dmin2)
    Espera2=1;
    Ejecuta2=0;
    if(Terminada1==0)
        Ejecuta1=1;
    end
    if(Espera1==1)
        Ejecuta1=0;
    end
    else
        Espera2=0;
    end

if(Ejecuta1==1 & Espera1==0)
    Cp1(k)=Cp1(k-1)+1;
end
if(Ejecuta1==0 & k>=L1)
    Cp1(k)=Cp1(k-1);
end
end

```

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

```
if(Ejecuta2==1 & Espera2==0)
    Cp2(k)=Cp2(k-1)+1;
end
if(Ejecuta2==0 & k>=L2)
    Cp2(k)=Cp2(k-1);
end
Carga1=Cp1(k);
Carga2=Cp2(k);
k=k+1
%pause(1);
Cp1T=Cp1';
Cp2T=Cp2';
end
```

Programa 3. Simulador del Algoritmo de planificación FIFO

```

close all;
clear all;

k=1

%Banco de Tareas
%Tarea 1 Datos Iniciales
L1=13
Dmin1=32
Dmax1=110
C1=42
for i=1:L1
    Cp1(i)=0;
    Cf1(i)=C1-Cp1(i);
    TDmax1(i)=Dmax1-k;
    P1(i)=(Cf1(i)/TDmax1(i)) * ((Dmin1-L1)/(Dmax1-L1));
end
Terminada1=0;

%Tarea 2 Datos iniciales
L2=28
Dmin2=34
Dmax2=130
C2=50
for i=1:L2
    Cp2(i)=0;
    Cf2(i)=C2-Cp2(i);
    TDmax2(i)=Dmax2-k;
    P2(i)=0;
end
Terminada2=0;
k=k+1
TareaEsporadica=0;
while(Terminada1==0 | Terminada2==0)
    if(k>=L1 & Terminada1==0)
        for i=1:5
            Cp1(k)=Cp1(k-1)+1;
            if(Cp1(k)>C1)
                Terminada1=1;
                Cp1(k)=Cp1(k-1);
            end
            Cp2(k)=Cp2(k-1);
            TDmax2(k)=0;
            Cf2(k)=0;
            P2(k)=0;
            if(TareaEsporadica==1)
                TDmax2(k)=Dmax2-k;
                Cf2(k)=C2-Cp2(k-1);
                P2(k)=(Cf2(k)/TDmax2(k)) * ((Dmin2-L2)/(Dmax2-L2));
            end
            k=k+1
        end
    end
end

```

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

```
end
if(k>=L2)
    TareaEsporadica=1;
end
Cp1(k)=Cp1(k-1);
if(Terminada1==0 & TareaEsporadica==1)
    Cp2(k)=Cp2(k-1);
    Cf2(k)=Cf2(k-1);
    P2(k)=P2(k-1);
end
if(Terminada1==1 & TareaEsporadica==1)
    Cp2(k)=Cp2(k-1)+1;
    TDmax2(k)=Dmax2-k;
    Cf2(k)=C2-Cp2(k-1);
    P2(k)=(Cf2(k)/TDmax2(k)) * ((Dmin2-L2)/(Dmax2-L2));
    if(Cp2(k)>=C2)
        Terminada2=1;
    end
end
end
k=k+1
end
Cp1T=Cp1';
Cp2T=Cp2';
```

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

Muestra de Datos de la Simulación para las Tareas en Tiempo Real Esporádicas.				
	Tiempo de Arribo	Plazo Mínimo	Carga	Plazo Máximo
	TA	Dm	C	DM
Simulación 1	26	53	15	99
	73	83	21	104
	94	97	47	182
	104	116	34	192
	126	170	28	179
	151	182	35	189
	334	336	7	398
	442	511	27	534
	479	490	14	569
	529	564	27	595
	589	675	33	686
	758	820	21	837
	910	944	2	1004
	972	1040	49	1047
	980	986	30	1043
Simulación 2	58	70	46	144
	132	190	49	206
	179	227	36	255
	226	247	17	257
	335	349	29	350
	386	399	18	481
	402	474	38	494
	419	429	38	491
	557	572	14	624
	752	820	45	848
	771	810	48	849
	857	888	15	941
Simulación 3	203	215	32	263
	335	356	24	358
	550	578	27	642
	609	668	18	674
	694	699	43	709
	706	718	22	747
Simulación 4	73	101	11	134
	232	298	11	331
	289	300	10	314
	611	680	48	694
	672	741	40	753
	695	712	28	788
	829	872	16	905

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

	879	925	48	934	
	891	910	25	972	
	928	998	14	1000	
	86	86	25	129	Simulación 5
	137	137	37	141	
	213	213	48	235	
	202	202	20	220	
	212	212	41	280	
	229	229	2	240	
	234	234	3	248	
	241	241	23	242	
	282	282	48	337	
	533	533	10	551	
	484	484	31	518	
	550	550	45	582	
	593	593	5	643	
	715	715	37	762	
	791	791	29	829	
	784	784	10	862	
	831	831	36	849	
	314	344	16	399	Simulación 6
	514	536	38	559	
	846	857	8	939	
	963	973	47	989	
	127	168	13	201	Simulación 7
	189	220	20	226	
	282	318	13	337	
	470	491	15	520	
	554	591	9	607	
	728	778	40	791	
	873	884	7	935	
	889	960	24	976	
	178	214	32	251	Simulación 8
	231	260	25	300	
	308	372	2	402	
	328	352	16	418	
	364	392	41	422	
	368	429	46	462	
	494	557	41	588	
	671	678	46	768	
	693	755	9	756	
	738	791	41	794	
	792	810	24	848	
	922	929	17	953	
	968	976	35	1063	

Análisis de planificabilidad de los algoritmos EDF y FIFO usando la dispersión de los tiempos de arribo en tareas de tiempo real esporádicas

999	1068	39	1095	
16	29	35	102	Simulación 9
17	93	5	107	
120	122	7	164	
262	324	49	345	
270	348	33	355	
354	391	44	453	
591	622	41	642	
680	756	18	769	
745	754	50	817	
770	802	31	806	
800	839	43	879	
871	909	7	946	
964	985	7	1056	
100	133	7	143	Simulación 10
124	138	32	162	
144	154	7	236	
204	242	1	270	
263	266	16	289	
371	442	35	466	
467	472	16	565	
508	563	5	602	
544	617	22	633	
737	792	11	801	
825	889	44	897	
966	968	47	986	