

ACTIVIDAD EN CLASE

Planificación de Procesos – Sistemas Operativos

Modalidad: Presencial

Tipo: Trabajo en clase

Nombres: Priscila Morales, Doménica Iza, Felipe Merino, Matthew Verdezoto

Organización de grupos

- La clase se organizará en **grupos de tres (3) estudiantes**.
- Cada grupo entregará **un único documento** con el desarrollo completo de la actividad.

Objetivo

Aplicar los principales algoritmos de planificación de procesos utilizados en los sistemas operativos, mediante la elaboración de diagramas de ejecución, el cálculo de métricas de desempeño y el análisis comparativo de resultados.

Datos del problema

Considere el siguiente conjunto de procesos:

Proceso	Tiempo de Llegada	Ráfaga de CPU	Prioridad
P1	0	6	2
P2	1	4	1
P3	2	2	3
P4	3	5	2

Condiciones de trabajo: - Un valor numérico menor indica **mayor prioridad**. - Para el algoritmo Round Robin se debe utilizar un **quantum de 2 unidades de tiempo**.

Actividades a desarrollar por cada grupo

Cada grupo deberá resolver el problema aplicando **todos los algoritmos indicados**, siguiendo las instrucciones detalladas a continuación.

1. Planificación FIFO (First In, First Out)

1. Ordenar los procesos de acuerdo con su tiempo de llegada.

Proceso	Llegada	Ráfaga
P1	0	6
P2	1	4
P3	2	2
P4	3	5

P1 -> P2 -> P3 -> P4

2. Construir el diagrama de Gantt correspondiente.

Tiempos	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P1	P1																	
P2							P2											
P3											P3							
P4													P5					

3. Calcular para cada proceso:
 - Tiempo de espera.
 - Tiempo de retorno.

$$TE = \text{TiempoInicio} - \text{TiempoLlegada}$$

$$TR = \text{TiempoFinalización} - \text{TiempoLlegada}$$

Proceso	Inicio	Fin	TE	TR
P1	0	6	$0 - 0 = 0$	$6 - 0 = 6$
P2	6	9	$6 - 1 = 5$	$10 - 1 = 9$
P3	10	11	$10 - 2 = 8$	$12 - 2 = 10$
P4	12	17	$12 - 3 = 9$	$17 - 3 = 14$

2. Planificación SJF (Shortest Job First – no apropiativo)

1. Considerar únicamente los procesos que hayan llegado al sistema en el momento de selección.
2. Ordenar los procesos según la duración de su ráfaga de CPU.

Proceso	Tiempo de llegada	Ráfaga de CPU	Prioridad		Tiempo		
P1	0	6	2		Proceso Inicia	P1	
P2	1	4	1		Tiempo/termina	6	
P3	2	2	3				
P4	3	5	2		Tiempo	6	
					Proceso Terminado	P1	
					Procesos en Cola	P2	P3 P4
					Ráfagas de los Procesos	4	2 5
					Proceso Elegido (Menor tiempo de rafaga)	P3	
					Tiempo de Llegada P3	2	
					Tiempo en el que termina	(6+2) =	8
					Tiempo	8	
					Proceso terminado	P3	
					Procesos en Cola	P2	P4
					Ráfagas de los procesos	4	5
					Proceso elegido	P2	
					Tiempo de llegada P2	1	
					Tiempo en el que termina P2	(8+4) =	12
					Tiempo	12	
					Proceso Terminado	P2	
					Proceso en Cola	P4	
					Ráfaga del proceso	5	
					Tiempo de llegada P4	3	
					Tiempo en el que termina P4	(12 + 5) =	17

3. Construir el diagrama de Gantt.

Procesos	T. Inicia	T. termina																		
P1	0	6																		
P3	6	8																		
P2	8	12																		
P4	12	17																		

4. Calcular los tiempos de espera y retorno.

Tiempo Retorno	Tiempo de finalización - Tiempo de Llegada	Proceso	Tiempo de llegada	Ráfaga de CPU	Prioridad	Tiempo Finalizacion	Tiempo Retorno	Tiempo Espera
Tiempo Espera	Tiempo de Retorno - Ráfaga de CPU	P1	0	6	2	6	6	0
		P3	2	2	3	8	6	4
		P2	1	4	1	12	11	7
		P4	3	5	2	17	14	9
						Promedio	9,25	5

3. Planificación por prioridades (no apropiativo)

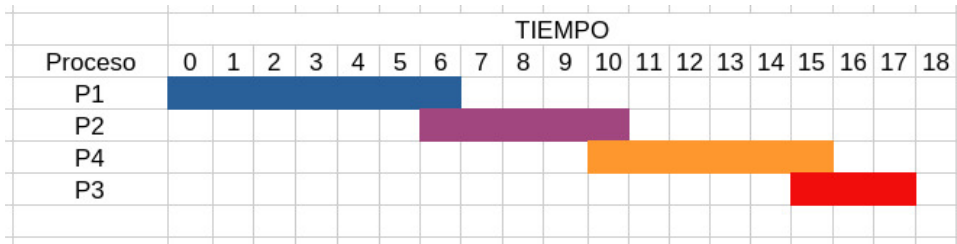
1. Ordenar los procesos según su nivel de prioridad.

Proceso	Tiempo Llegada	Ráfaga CPU	Prioridad
P1	0	6	2
P2	1	4	1
P4	2	5	2
P3	3	2	3
Proceso	Tiempo Inicial	Tiempo final	
P1	0	6	
P2	6	10	
P4	10	15	
P3	15	17	
Tiempo Total	17		

2. En caso de empate, aplicar el criterio FIFO.

En este caso no existe un empate apesar de que P1 y P4 tienen la misma prioridad, P1 y P4 no compiten para ver cual va primero en este caso P1 llega en un tiempo 0 así que entra directo mientras que P4 está en espera porque llega después y espera ser ejecutado en este caso no se aplica FIFO pero el caso de que exista un empate entre 2 o más procesos con la misma prioridad esperando en la cola de listos simultáneamente, el procesador seleccionará aquel que haya llegado primero.

3. Construir el diagrama de Gantt.



4. Calcular los tiempos de espera y retorno.

Proceso	Tiempo Llegada	Ráfaga CPU	Prioridad	Tiempo final	Tiempo Retorno	Tiempo Espera
P1	0	6	2	6	6	0
P2	1	4	1	10	9	5
P3	2	2	3	17	15	13
P4	3	5	2	15	12	7
TOTAL					42	25
PROMEDIO					10.5	6.25

$$\text{Tiempo Retorno} = \text{Tiempo final} - \text{Llegada}$$

$$\text{Tiempo Espera} = \text{Tiempo Retorno} - \text{Ráfaga CPU}$$

4. Planificación Round Robin (Turno rotativo)

1. Utilizar un quantum fijo de 2 unidades de tiempo.
2. Ejecutar los procesos de forma cíclica respetando el orden de llegada.
3. Construir el diagrama de Gantt completo.
4. Calcular los tiempos de espera y retorno.

Proceso	Tiempo de Llegada	Ráfaga de CPU	Prioridad
P1	0	6	2
P2	1	4	1
P3	2	2	3
P4	3	5	2

Dentro de este ejercicio, se utiliza multiprogramación porque hay coexistencia de múltiples procesos en memoria principal, estos comparten un único procesador de manera concurrente, aquí los cuatro procesos estarán presentes simultáneamente listos en la cola.

2. Ejecución de los procesos:

Estado inicial ($t = 0$):

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	0	6	En cola
P2	4	0	4	Sin llegar
P3	2	0	2	Sin llegar
P4	5	0	5	Sin llegar

Estado en el primer quantum ($t = 2$)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	0	6	En cola
P2	4	0	4	En cola
P3	2	0	2	En cola
P4	5	0	5	Sin llegar

Estado en el segundo quantum ($t = 4$):

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	2	4	En cola
P2	4	2	2	En cola
P3	2	0	2	En cola
P4	5	0	5	En cola

Estado en el tercer quantum ($t = 6$)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	4	2	En cola
P2	4	2	2	En cola
P3	2	2	0	Terminado
P4	5	0	5	En cola

Estado en el cuarto quantum ($t = 8$)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	4	2	En cola

P2	4	2	2	En cola
P3	2	2	0	Terminado
P4	5	0	5	En cola

Estado en el quinto quantum (t = 10)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	4	2	En cola
P2	4	2	2	En cola
P3	2	2	0	Terminado
P4	5	2	3	En cola

Estado en el sexto quantum (t = 12)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	4	2	En cola
P2	4	4	0	Terminado
P3	2	2	0	Terminado
P4	5	2	3	En cola

Estado en el séptimo quantum (t = 14)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	6	0	Terminado
P2	4	4	0	Terminado
P3	2	2	0	Terminado
P4	5	2	3	En cola

Estado en el octavo quantum (t = 16)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	6	0	Terminado
P2	4	4	0	Terminado
P3	2	2	0	Terminado
P4	5	4	1	En cola

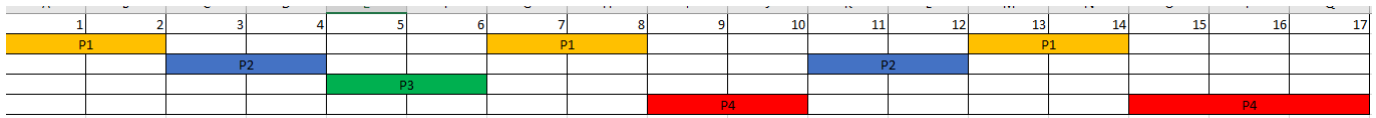
Estado en el noveno quantum (t = 18)

Proceso	Ráfaga inicial	Ráfaga usada	Ráfaga restante	Estado
P1	6	6	0	Terminado
P2	4	4	0	Terminado
P3	2	2	0	Terminado
P4	5	5	0	Terminado

Descripción de la cola de procesos:

Quantum	Tiempo	Proceso Ejecutado	Cola inicio del Quantum	Cola después del Quantum	Eventos
1	0-2	P1	[P1]	[P2, P3, P1]	t=1: Llega P2 t=2: Llega P3
2	2-4	P2	[P2, P3, P1]	[P3, P1, P4, P2]	t=3: Llega P4
3	4-6	P3	[P3, P1, P4, P2]	[P1, P4, P2]	P3 termina (sale del sistema)
4	6-8	P1	[P1, P4, P2]	[P4, P2, P1]	P1 no termina, entonces va al final
5	8-10	P4	[P4, P2, P1]	[P2, P1, P4]	P4 no termina, entonces va al final
6	10-12	P2	[P2, P1, P4]	[P1, P4]	P2 termina (sale del sistema)
7	12-14	P1	[P1, P4]	[P4]	P1 termina (sale del sistema)
8	14-16	P4	[P4]	[P4]	P4 no termina, vuelve a cola (solo él)
9	16-17	P4	[P4]	[] (vacía)	P4 termina (sale del sistema)

3. Diagrama de Gantt



4. Cálculo de los tiempos de espera y retorno

Proceso	T. Llegada	Ráfaga CPU	T. Finalización	T. Retorno (F-L)	T. Espera (R-Ráfaga)
P1	0	6	14	$14 - 0 = 14$	$14 - 6 = 8$
P2	1	4	12	$12 - 1 = 11$	$11 - 4 = 7$
P3	2	2	6	$6 - 2 = 4$	$4 - 2 = 2$
P4	3	5	17	$17 - 3 = 14$	$14 - 5 = 9$

Análisis de resultados

Responder de manera clara y fundamentada:

1. ¿Qué algoritmo presenta el menor tiempo de espera promedio?

	TIEMPO PROMEDIO DE ESPERA
Planificación FIFO (First In, First Out)	5.5
Planificación SJF (Shortest Job First – no apropiativo)	5
Planificación por prioridades (no apropiativo)	6.25
Planificación Round Robin (Turno rotativo)	6.5

Según la tabla construida el algoritmo SJF fue el de menor tiempo de espera promedio, con solo 5 unidades de tiempo, esto se debe a su forma de trabajar, al siempre seleccionar el proceso más corto optimiza de mejor manera el uso del CPU, porque el proceso más corto siempre liberará el CPU más rápido.

2. ¿Cuál de los algoritmos analizados ofrece un comportamiento más equitativo entre procesos? Justifique su respuesta.

Planificación Round Robin (Turno rotativo), es el más equitativo porque asigna un tiempo límite fijo (Quantum) a cada proceso de forma cíclica. Ningún proceso puede monopolizar

la CPU, garantizando que todos avancen periódicamente sin importar su tamaño o prioridad.

3. ¿En qué algoritmo puede producirse el fenómeno de inanición? Explique.

Principalmente en la Planificación por Prioridades y en el SJF, ya que, cuando un proceso de baja prioridad (o de ráfaga muy larga en el caso de SJF) nunca llega a ejecutarse porque constantemente llegan nuevos procesos con mayor prioridad (o ráfagas más cortas), dejándolo en espera indefinida.

Entregable

Cada grupo deberá entregar

- Diagramas de Gantt correspondientes a cada algoritmo.
- Cálculo detallado de los tiempos de espera y retorno.
- Respuestas al análisis de resultados.

Tiempo asignado

90 minutos

Criterios de evaluación

Criterio	Puntaje
Aplicación correcta de los algoritmos	4
Diagramas de Gantt y cálculos	3
Análisis técnico de resultados	3
Total	10 puntos