

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

ACTIVIDAD LABORATORIO NO.1
TRABAJO CON LLAMADAS AL SISTEMA

PRESENTADO POR:
ALEJANDRO DE MENDOZA

PRESENTADO AL PROFESOR:
ING ALBERTO FERREIRA CASTRO

FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA
BOGOTÁ D.C.
30 DE OCTUBRE
2024

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

TABLA DE CONTENIDO

PREPARACIÓN DEL ENTORNO	3
DESARROLLO DEL ENTORNO – ESTUDIO DE ALGORITMOS DE PLANIFICACIÓN.....	3
ALGORITMO FIFO (FIRST IN, FIRST OUT)	4
DESARROLLO ALGORITMO FIFO (FIRST IN, FIRST OUT).....	4
ANÁLISIS ALGORITMO FIFO – TIEMPO ESPERA PROMEDIO, TIEMPO DE RESPUESTA, PENALIZACIÓN Y EFICIENCIA.....	7
ALGORITMO SJF (SHORT JOB FIRST)	7
DESARROLLO ALGORITMO SJF (SHORT JOB FIRST).....	7
ANÁLISIS ALGORITMO SJF – TIEMPO ESPERA PROMEDIO, TIEMPO DE RESPUESTA, PENALIZACIÓN Y EFICIENCIA.....	10
ALGORITMO ASIGNACIÓN POR PRIORIDADES.....	10
DESARROLLO ALGORITMO ASIGNACIÓN POR PRIORIDADES	11
ANÁLISIS ALGORITMO ASIGNACIÓN POR PRIORIDADES – TIEMPO ESPERA PROMEDIO, TIEMPO DE RESPUESTA, PENALIZACIÓN Y EFICIENCIA.....	13
IMPLEMENTACIÓN DE ALGORITMOS EN DEV-C++	13
DESARROLLO ALGORITMO FIFO EN DEV-C++	13
DESARROLLO ALGORITMO SJF EN DEV-C++.....	14
DESARROLLO ALGORITMO ASIGNACIÓN POR PRIORIDADES EN DEV-C++	14
ANÁLISIS DE RESULTADOS	14
BIBLIOGRAFÍA	15

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

PREPARACIÓN DEL ENTORNO

Como preparación del entorno, es necesario indicar que las llamadas al sistema sirven de interfaz a usuarios y procesos para poder llevar a cabo tareas del sistema operativo. Las llamadas al sistema se ofrecen en una biblioteca o librería de funciones de C/C++ que se suelen incluir en los compiladores. La mayoría de los lenguajes de programación incluyen una interfaz de llamadas al sistema. En esta actividad frente a la planificación de procesos y procesadores, que es tema sucesor de las llamadas al sistema y énfasis de desarrollo del laboratorio de esta aula, nos vamos a enfocar en los siguientes algoritmos de planificación utilizados en los sistemas operativos: **FIFO**, **SJF**, **Asignación Por Prioridades**. Con los cuales vamos a determinar el performance de cada uno, su análisis comparativo de eficiencia y su desarrollo en código en la **plataforma Dev-C++** claramente en el lenguaje C++.

DESARROLLO DEL ENTORNO – ESTUDIO DE ALGORITMOS DE PLANIFICACIÓN

Para el desarrollo de estos algoritmos me voy a centrar en el desarrollo del código en la plataforma Dev-C++, en las aulas virtuales del profesor Alberto Ferreira, en la información del aula virtual de Sistemas Operativos, en los libros del aula y en el video explicativo adjunto en la plataforma Padlet de la clase de Sistemas Operativos de la Fundación Universitaria Internacional de la Rioja, de nombre: “Explicación Algoritmos De Planificación De Procesos De La CPU Con Ejemplos | FIFO | SJF | RR | SRTF”.

Dando inicio a la ejecución y desarrollo de la actividad lo primero es partir de una tabla de procesos donde hay cinco procesos A, B, C, D y E:

TABLA DE PROCESOS			
PROCESOS	TIEMPO DE CREACION	TIEMPO DE CPU	PRIORIDADES
A	3	1	2
B	2	5	3
C	0	4	3
D	5	3	1
E	7	2	2

Como podemos ver en la tabla tenemos un ciclo de creación denotado con la letra (C), que es el Tiempo que tarda un proceso en pasar de 'Nuevo' a 'Listo', y después tenemos el tiempo de CPU denotado como el tiempo T o tiempo de ráfaga que es el tiempo que toma un proceso que se está 'Ejecutando'. Ahora con base en esta

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

información debemos determinar los 3 parámetros que son: **Tiempo de Respuesta o Finalización (F)**: Tiempo total para completar el proceso $F=t+E$. **Tiempo de Espera (E)**: Tiempo que un proceso permanece en estado 'Listo'. $E=F-t$. **Penalización (P)**: Proporción del tiempo de respuesta en que el proceso estuvo en 'Listo'. $P=F/t$, donde T es igual al tiempo de CPU, y E es el tiempo de espera que vamos a obtener con base en un diagrama de Gantt que vamos a desarrollar para evaluar y planificar los procesos según el algoritmo a evaluar. Por último, vamos a calcular lo que es la penalización que como se indico es la proporción del tiempo de respuesta en que el proceso estuvo en 'Listo'. $P=F/t$, para de esta manera determinar cuál de los 3 algoritmos nos va a dar el mejor performance buscando siempre que la penalización sea lo más baja posible, al igual que el mínimo valor en cuanto al tiempo de espera y en cuanto al tiempo de respuesta.

ALGORITMO FIFO (First In, First Out)

El algoritmo FIFO, es el esquema más simple de planificación, el primero llegado, primero servido (FCFS). Es un mecanismo cooperativo, cada proceso se ejecuta en el orden en que fue llegando hasta que suelta el control. El despachador es una cola. El algoritmo FIFO reduce al mínimo la sobrecarga administrativa. Este algoritmo dará servicio y salida a todos los procesos siempre que $p \leq 1$. En resumen, con este algoritmo se asigna el procesador al primer proceso que lo solicita, da su salida y el tiempo medio de espera con este algoritmo suele ser bastante alto.

Desarrollo Algoritmo FIFO (First In, First Out)

En este caso el primer proceso que se va a ejecutar es el proceso C, como lo podemos ver en la tabla que muestro a continuación:

TABLA DE PROCESOS			
PROCESOS	TIEMPO DE CREACION	TIEMPO DE CPU	PRIORIDADES
A	3	1	2
B	2	5	3
C	0	4	3
D	5	3	1
E	7	2	2

Ahora desarrollando el modelo Gantt queda de la siguiente manera:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X														
B															
A															
D															
E															
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

FIFO indica que el primero que ingresa es el primero que sale, entonces inicia el proceso C y va a utilizar toda la CPU durante esos cuatro periodos:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B														
A														
D														
E														
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Mientras el proceso C se ejecuta en la CPU aparece el proceso B en el tiempo 2 y el proceso A en el tiempo 3:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B			X											
A				X										
D														
E														
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El algoritmo analiza cuál es el proceso que ahora se va a ejecutar y en este caso sería el proceso B por su llegada. Cuando C termine su proceso y termine de generar uso de la CPU, entonces el proceso B va a dar inicio y va a ejecutarse durante 5 tiempos:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B			X		X	X	X	X	X					
A				X										
D														
E														
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Mientras se ejecuta B en el tiempo, en el tiempo 5 apareció el proceso D, por lo que va a ser el sucesor del proceso A, y de igual manera mientras se ejecuta el proceso B, en el tiempo 7 aparece el proceso E por lo que sería el sucesor del proceso D:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B			X		X	X	X	X	X					
A				X										
D						X								
E								X						
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Cuando termina el proceso B, procedemos a ejecutar el proceso A que se va a desarrollar durante un (1) solo intervalo de reloj o un periodo de tiempo:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B			X		X	X	X	X	X					
A				X						X				
D						X								
E								X						
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ahora continua el proceso D que cuenta con 3 intervalos de tiempo o de reloj en el CPU para utilizar:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)													
C	X	X	X	X										
B			X		X	X	X	X	X					
A				X						X				
D						X					X	X	X	
E								X						
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

Por último, tenemos el proceso E sucesor del proceso D y este proceso se va a ejecutar durante 2 periodos de reloj o intervalos de tiempo de uso de la CPU:

No Apropiativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X	X	X	X											
B			X		X	X	X	X							
A				X						X					
D						X					X	X	X		
E								X						X	
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Procedemos entonces a determinar y visualizar el **tiempo de espera** de cada proceso. En el proceso C, este proceso no tuvo ningún tipo de espera en la cola ya que su inicio fue inmediato y entro directamente a ejecutarse, entonces su espera fue de 0:

No Apropiativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X	X	X	X											
B					X	X	X	X	X						
A										X					
D						X					X	X	X		
E								X					X	X	
Clock	0			3	4	5	6	7	8	9	10	11	12	13	14

Alejandro De Mendoza:
No hubo espera en este proceso.

En el proceso B se tuvo que esperar 2 intervalos de tiempo:

No Apropiativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)															
C	X	X	X	X												
B			X		X	X	X	X	X							
A										X						
D											X	X	X			
E													X	X		
Clock				2	3	4	5	6	7	8	9	10	11	12	13	14

Alejandro De Mendoza:

Tiempo de espera en (color verde claro).

En el proceso A, este proceso tuvo que generar una espera de 6 intervalos de tiempo:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X	X	X	X											
B			X		X	X	X	X	X						
A				X						X					
D											X	X	X		
E														X	
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

En el proceso D, se tuvo que generar una espera de 5 intervalos de tiempo:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X	X	X	X											
B			X		X	X	X	X	X						
A				X						X					
D						X					X	X	X		
E														X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Y, en el proceso E, se tuvo que generar una espera de 6 intervalos de tiempo:

No Apropriativo	FIFO - FCFS (FIRST TO COME, FIRST TO SERVE)														
C	X	X	X	X											
B			X		X	X	X	X	X						
A				X						X					
D						X					X	X	X		
E								X						X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Ahora procedemos a desarrollar una tabla donde se indican los tiempos de espera, los tiempos de respuesta y la penalización:

PARAMETROS Y LISTAS A EVALUAR	
Quantum (q):	Intervalo de tiempo de ejecución
Tiempo de Creación (C):	Tiempo que tarda un proceso en pasar de 'Nuevo' a 'Listo'
Tiempo o Ráfaga de CPU (t):	Tiempo que un proceso está en 'Ejecutando'
Tiempo de Respuesta o Finalización (F):	Tiempo total para completar el proceso $F=t+E$
Tiempo de Espera (E):	Tiempo que un proceso permanece en estado 'Listo'. $E=F-t$
Penalización (P):	Proporción del tiempo de respuesta en que el proceso estuvo en 'Listo'. $P=F/t$

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

- Los **tiempos de espera** entonces son: $C = 0$, $B = 2$, $A = 6$, $D = 5$, $E = 6$.
- El cálculo del **tiempo de respuesta** es (tiempo de espera + tiempo de CPU): $C = 0+4 = 4$, $B = 2+5 = 7$, $A = 6+1 = 7$, $D = 5+3 = 8$, $E = 6+2 = 8$
- El cálculo de la **penalización** (Tiempo de respuesta / Tiempo de CPU), nos da: $C = 4/4 = 1$, $B = 7/5 = 1.4$, $A = 7/1 = 7$, $D = 8/3 = 2.667$, $E = 8/2 = 4$.

A continuación, la tabla ilustrativa:

TIEMPO DE ESPERA		TIEMPO DE RESPUESTA		PENALIZACIÓN	
C	0	C	4	C	1
B	2	B	7	B	1,4
A	6	A	7	A	7
D	5	D	8	D	2,6666667
E	6	E	8	E	4
E(Media)	3,8	F(Media)	6,8	P(Media)	3,21333333

Análisis Algoritmo FIFO – Tiempo Espera Promedio, Tiempo de Respuesta, Penalización y Eficiencia

En el algoritmo FIFO, su funcionamiento se basa en una cola donde los procesos se colocan en el orden en que llegan, y se atienden en ese mismo orden. Ahora podemos determinar que el tiempo de espera medio es de 3.8, y es el menos eficiente frente a los otros dos algoritmos; el SJF y el algoritmo de Asignación Por Prioridades. El tiempo de respuesta medio es de 6.8, también es el menos eficiente frente a los otros dos algoritmos. El tiempo de penalización medio es de 3.21, y también es el menos eficiente frente a los otros dos algoritmos. El tiempo de penalización es un tiempo bastante alto en lo que es el proceso E. El proceso estuvo listo para ejecutarse en muchos intervalos de la CPU, frente a la proporción que uso la CPU. En conclusión, este algoritmo es el menos eficiente de los 3 para este desarrollo.

ALGORITMO SJF (Short Job First)

El algoritmo SJF, es un algoritmo de planificación de procesos en sistemas operativos que selecciona el proceso con el tiempo de ejecución más corto para ejecutarse primero. Este algoritmo es eficiente en la minimización del tiempo promedio de espera, no interrumpe un proceso en ejecución si llega otro más corto. Cuando el procesador está libre se asigna el proceso que tiene la siguiente ráfaga más corto. Si dos procesos tienen la misma ráfaga se utiliza el algoritmo FIFO para decidir.

Desarrollo Algoritmo SJF (Short Job First)

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

El proceso C va a ejecutarse de inmediato, con un tiempo de espera de 0, y se va a ejecutar durante 4 periodos:

No Apropiativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A															
B															
E															
D															
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

El proceso B se crea en el tiempo 2 y el proceso A en el tiempo 3, durante la ejecución del proceso C:

No Apropriativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A				X											
D															
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Entonces el algoritmo tiene que decidir, **cuál de los dos procesos conlleva el tiempo más corto de CPU**. El proceso A se tiene que ejecutar durante un periodo y B se tiene que ejecutar durante 5 periodos, por ende, A va a ser el proceso que se va a ejecutar primero:

No Apropiativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A				X	X										
D															
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Ahora viene la **sorpresa** que al terminar de ejecutarse el proceso A en el periodo 5, entra el proceso D en este mismo periodo y tiene un tiempo de CPU de 3, por ende, el proceso B sigue continuando en espera y se ejecuta el proceso D:

No Apropiativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A	Alejandro De Mendoza: Este proceso tiene un tiempo de CPU de 5.			X	X										
D							X								
E															
B				X											
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Se ejecuta el proceso de D que toma 3 periodos de tiempo:

No Apropriativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A				X	X										
D						X	X	X							
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Ahora nuevamente nos damos la sorpresa que en el último ciclo de tiempo de ejecución del proceso D, entra el proceso E:

No Apropiativo	SJF - SHORT JOB FIRST														
C	X	X	X	X											
A				X	X										
D						X	X	X							
E								X							
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

El algoritmo desarrolla análisis y como este proceso E tiene un tiempo de CPU de 2, por ende, el proceso B queda nuevamente en espera y se ejecuta el proceso E:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X						
B			X											
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Se ejecuta el proceso E que maneja un total de 2 periodos:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X											
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ejecutamos el proceso B, con un total de 5 periodos de tiempo en CPU:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ahora, determinamos el **tiempo de espera** de los procesos, y el proceso C, no tuvo ningún tipo de espera en la cola, por ende, la espera en este proceso fue de 0:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X			X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El tiempo de espera de A ahora nos quedó en 1:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X										
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El tiempo de espera del proceso D nos da un total de 0 periodos de tiempo al igual que el proceso C, ya que se ejecutó de inmediato:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El tiempo de espera del proceso E, da un tiempo de espera de 1:

No Apropriativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

Finalmente, el tiempo de espera del proceso B tiene un total de 8 periodos de tiempo de espera, se incrementó en dos periodos de tiempo frente al algoritmo anterior FIFO y se iguala al de prioridades, se vuelve un proceso crítico en este escenario:

No Apropiativo	SJF - SHORT JOB FIRST													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Desarrollamos los tiempos de espera, los tiempos de respuesta y la Penalización:

- El cálculo de **los tiempos de espera** es: C = 0, A = 1, D = 0, E = 1, B = 8.
- El **cálculo del tiempo de respuesta** es el siguiente (tiempo de espera + tiempo de CPU): C = 0+4 = 4, A = 1+1 = 2, D = 0+3 = 3, E = 1+2 = 3, B = 8+5 = 13.
- El **cálculo de la penalización** (Tiempo de respuesta / Tiempo de CPU), nos da:
C = 4/4 = 1, A = 2/1 = 2, D = 3/3 = 1, E = 3/2 = 1.5, B = 13/5 = 2.6.

La tabla entonces queda de la siguiente manera:

TIEMPO DE ESPERA		TIEMPO DE RESPUESTA		PENALIZACIÓN	
C	0	C	4	C	1
A	1	A	2	A	2
D	0	D	3	D	1
E	1	E	3	E	1,5
B	8	B	13	B	2,6
E(Media)	2	F(Media)	5	P(Media)	1,62

Análisis Algoritmo SJF – Tiempo Espera Promedio, Tiempo de Respuesta, Penalización y Eficiencia

El tiempo de espera medio es de 2, mejor que el algoritmo FIFO e igual que el de Asignación de Prioridades. El tiempo de respuesta medio es de 5, mejor que el FIFO e igual que el de Prioridades. El tiempo de penalización medio es de 1.62, mejor que el FIFO e igual que el de Prioridades. El tiempo de penalización es más eficiente que el FIFO e igual que el de Prioridades. Ahora el proceso B es crítico en este algoritmo ya que el tiempo de respuesta del proceso fue ineficiente en relación a la proporción que uso de la CPU que es de 5 ciclos de tiempo. Este algoritmo es más eficiente que el algoritmo FIFO e igual que el de Asignación de Prioridades.

ALGORITMO ASIGNACIÓN POR PRIORIDADES

El algoritmo de asignación por prioridades, es un método de planificación de procesos en sistemas operativos en el que cada proceso recibe una prioridad, y el proceso con la prioridad más alta se ejecuta primero. En la asignación por prioridades el sistema

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

operativo les asigna prioridades a los procesos para poder evaluar cuál sería el proceso más corto y ejecutar el proceso más prioritario para asignarlo a la CPU.

Desarrollo Algoritmo Asignación Por Prioridades

El proceso C, es el primero en llegar y la CPU lo ejecuta de manera inmediata:

No Apropiativo	ASIGNACIÓN POR PRIORIDADES														
C	1	1	1	1											
A															
B															
D															
E															
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

En la ejecución de C se crea el proceso B en el tiempo 2 y el A en el tiempo 3:

No Apropiativo	ASIGNACIÓN POR PRIORIDADES														
C	X	X	X	X											
A				X											
D															
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

El planificador tiene que tomar una decisión, y debe buscar y verificar cual es el proceso más prioritario entre A y B, frente a la siguiente tabla:

TABLA DE PROCESOS			
PROCESOS	TIEMPO DE CREACION	TIEMPO DE CPU	PRIORIDADES
A	3	1	2
B	2	5	3
C	0	4	3
D	5	3	1
E	7	2	2

Vemos entonces la imagen del diagrama de Gantt:

No Apropiativo	ASIGNACIÓN POR PRIORIDADES														
C	X	X	X	X											
A				X											
D															
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Al tener una mayor prioridad A frente a B, entonces se corre primero el proceso de A. Es importante indicar que si tanto el proceso A como el proceso B hubiesen tenido la misma prioridad se evaluaría cuál de los dos procesos es el de menor tiempo o el más corto para poder ejecutarse, que de igual manera sería el proceso A:

No Apropiativo	ASIGNACIÓN POR PRIORIDADES														
C	X	X	X	X											
A				X	X										
D															
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Ahora el proceso D entra en ejecución en el tiempo 5 y su nivel de prioridad 1 es mayor al de la prioridad de B, por lo que entra a correr primero este proceso:

No Apropiativo	ASIGNACIÓN POR PRIORIDADES														
C	X	X	X	X											
A				X	X										
D						X	X	X							
E															
B			X												
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

En el tiempo 7 aparece el proceso E con una prioridad de 2 mientras se ejecuta D:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X						
B			X											
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

E va a ser entonces el proceso que se va a ejecutar primero:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X											
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Y por último ejecutamos el proceso B con un total de 5 ciclos de tiempo periodos:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Ahora el tiempo de espera en el proceso C, es claro, fue de 0:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El tiempo de espera de A fue 1:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El tiempo de espera del algoritmo D es 0 ya que se ejecutó de inmediato:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El proceso E, tuvo que generar una espera de 1 intervalo de tiempo:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

El proceso B, tuvo que generar una espera de 8 intervalos de tiempo:

No Apropriativo	ASIGNACIÓN POR PRIORIDADES													
C	X	X	X	X										
A				X	X									
D						X	X	X						
E								X	X	X				
B			X								X	X	X	X
Clock	0	1	2	3	4	5	6	7	8	9	10	11	12	13

Desarrollamos los tiempos de espera, los tiempos de respuesta y la Penalización:

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

- El cálculo de **los tiempos de espera** es: $C = 0$, $A = 1$, $D = 0$, $E = 1$, $B = 8$.
- **El cálculo del tiempo de respuesta** es (tiempo de espera + tiempo de CPU), $C = 0+4 = 4$, $A = 1+1 = 2$, $D = 0+3 = 3$, $E = 1+2 = 3$, $B = 8+5 = 13$.
- El **cálculo de la Penalización** (Tiempo de respuesta / Tiempo de CPU), nos da: $C = 4/4 = 1$, $A = 2/1 = 2$, $D = 3/3 = 1$, $E = 3/2 = 1.5$, $B = 13/5 = 2.6$.

La tabla entonces queda de la siguiente manera:

TIEMPO DE ESPERA		TIEMPO DE RESPUESTA		PENALIZACIÓN	
C	0	C	4	C	1
A	1	A	2	A	2
D	0	B	3	B	1
E	1	D	3	D	1,5
B	8	E	13	E	2,6
E(Media)	2	F(Media)	5	P(Media)	1,62

Análisis Algoritmo Asignación Por Prioridades – Tiempo Espera Promedio, Tiempo de Respuesta, Penalización y Eficiencia

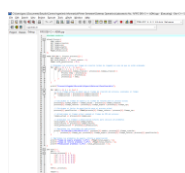
El tiempo de espera medio es de 2, mejor que el algoritmo FIFO e igual a SJF. El tiempo de respuesta medio es de 5, mejor que el algoritmo FIFO e igual a SJF. El tiempo de penalización medio es de 1.62, mejor que el algoritmo FIFO e igual a SJF. El tiempo de penalización está un poco alto, mejor frente al algoritmo FIFO, y es igual al algoritmo SJF. El proceso B al igual que el algoritmo SJF fue el proceso crítico en este caso ya que el proceso estuvo listo para ejecutarse en muchos intervalos de la CPU, frente a la proporción de la misma. Se comprueba entonces que este algoritmo al igual que el algoritmo SJF, es más eficiente que el algoritmo FIFO.

IMPLEMENTACIÓN DE ALGORITMOS EN DEV-C++

El desarrollo de los 3 algoritmos se ejecutó en la plataforma Dev – C++. Se adjuntan los programas **en documentos aparte** de acuerdo a lo indicado. (*NOTA: las imágenes se pueden ampliar para comprobar resultados y códigos sin dañar pixeles, se redujeron por restricción de espacio en el trabajo*).

Desarrollo Algoritmo FIFO en Dev-C++

A continuación, la imagen del código implementado para su desarrollo:



Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

A continuación, la salida de la consola:

Desarrollo Algoritmo SJF en Dev-C++

A continuación, la imagen del código implementado para su desarrollo:

A continuación, la salida de la consola:

Desarrollo Algoritmo Asignación Por Prioridades en Dev-C++

A continuación, la imagen del código implementado para su desarrollo:

A continuación, la salida de la consola:

ANÁLISIS DE RESULTADOS

En este estudio, se llevó a cabo una comparación detallada entre tres algoritmos de planificación de CPU: FIFO, SJF y el Algoritmo de Asignación por Prioridades, a través de la evaluación de parámetros como el tiempo de espera, el tiempo de respuesta, la penalización y la eficiencia, por lo que hemos podido extraer varias conclusiones. El algoritmo **FIFO**, aunque sencillo y fácil de implementar, a menudo resulta en mayores tiempos de espera y penalizaciones, especialmente en secuencias de procesos donde los tiempos de ejecución varían considerablemente. Por otro lado, **SJF** demostró ser más eficiente en términos de tiempo de espera y penalización, ya que prioriza los trabajos de menor duración. Este enfoque permite una utilización más óptima del

Asignatura	Datos del alumno	Fecha
Sistemas operativos	Apellidos:	
	Nombre:	

CPU, minimizando los periodos de inactividad y reduciendo significativamente los tiempos promedio de espera y respuesta. Ahora en cuanto al **Algoritmo de Asignación por Prioridades** aporta una dimensión adicional al considerar la importancia relativa de los procesos (Prioridad). En casos donde las prioridades están bien definidas y reflejan adecuadamente la urgencia de los procesos, este algoritmo puede superar a los otros en términos de tiempo de respuesta y penalización. A continuación, la tabla de los resultados de penalización finales para los tres algoritmos, que impacta directamente en la eficiencia, y se busca es la menor penalización:

PENALIZACIÓN	
FIFO	3,21333333
SJF	1,62
PRIORIDADES	1,62

Como podemos ver en esa actividad los algoritmos más **eficientes** son el **SJF** y el algoritmo de **Asignación Por Prioridades**, que tienen un mismo margen **menor** de penalización al FIFO. Por esto, se concluye que FIFO es útil en sistemas donde la simplicidad y la equidad son primordiales. SJF y el Algoritmo de Asignación por Prioridades ofrecen ventajas significativas en contextos donde la minimización de tiempos de espera y respuesta es crítica y donde el objetivo es claro; minimizar el tiempo de espera, el tiempo de respuesta y la penalización que impacta sobre la eficiencia como se ha denotado en el desarrollo de esta actividad.

BIBLIOGRAFÍA

A continuación, la bibliografía implementada:

- ✓ Tema 2: Estructura de los sistemas operativos. Agosto 2024 2Q. Y Tema 3: Planificación de procesos. Agosto 2024 2Q.
- ✓ Clases virtuales con el profesor Alberto Castro Laerte.
- ✓ Greg Gagne, Peter Baer Galvin, Abraham Silberschatz. Fundamentos de sistemas operativos.
- ✓ FNE Profesor, Plataforma YouTube, Explicación Algoritmos De Planificación De Procesos De La CPU Con Ejemplos | FIFO | SJF | RR | SRTF
- ✓ Plataforma Padlet, Información brindada por el profesor Alberto Castro Laerte.