4️⃣

Practica 4

1. Memoria Virtual

a. Describa qué beneficios introduce este esquema de administración de la memoria.

Beneficios de la memoria virtual:

1. **Uso eficiente de la memoria fisica:** Se intena alojar en memoria la mayor cantidad de paginas necesarias posibles
2. **Aprovechamiento del modelo de localidad:** Se aprovecha del comportamiento natural de los procesos, donde las referencias a datos y programas tienden a agruparse
3. **Capacidad de recuperacion y continuidad:** Permite al programa continuar su ejecucion a pesar de que una pagina no este cargada, seleccionando una victima y reemplazandola con la nueva
4. **Gestion dinamica de recursos:** La asignacion de marcos puede ajustarse a cada proceso

b. ¿En que se debe apoyar el Kernel para su implementación?

Para lograr esto, el kernel se apoya en 4 herramientas

- **TLB:** Necesita gaurdar en cache las ubicaciones que ya conoce para no perder tiempo buscando en tablas gigantes
- **Marcas:** Necesita que el HW le avise que paginas se estan usando, por ejemplo: el bit de referencia (R)
- **Page Faults:** El sistema depende de la capacidad de generar y manejar fallos de pagina, cada vez que se necesita cargar una pagina en un marco y no esta en memoria fisica

- **Frames y Swaps:** El SO debe reservar marcos especificos para descarga asincronica. Si tiene que guardar una pagina modificada en disco, la deja en uno de esos marcos y sigue atendiendo otros procesos sin quedarse colgada

c. Al implementar está técnica utilizando paginación por demanda, las tablas de páginas de un proceso deben contar con información adicional además del marco donde se encuentra la página. ¿Cuál es está información? ¿Por qué es necesaria?

Ademas de saber donde esta la pagina, la tabla necesita 3 etiquetas:

- **Bit de Validez:** *¿Está en RAM?*, Fundamental para saber si el proceso puede seguir o frenar y traer la pagina desde disco.
- **Bit de Referencia:** *¿Se usó recién?*, Sirve para que el algoritmo de reemplazo sepa a quien sacar
- **Bit de Modificacion:** *¿Se modificó?*, Avisa al sistema para saber si la pag se modifico desde que se cargo, si fue asi debe guardarla en disco antes de borrarla

---

2. Fallos de Pagina (Page Faults)

a. ¿Cuándo se producen?

Un fallo de pagina se produce cada vez que es necesario asignar una pagina en un marco de memoria fisica. Ocurre cuando un proceso intenta acceder a una pagina que no se encuentra cargada actualmente en memoria real.

b. ¿Quién es responsable de detectar un fallo de página?

La MMU es la encargada de detectarlo y generar un trap, el kernel resuelve el fallo

c. Describa las acciones que emprende el Kernel cuando se produce un fallo depágina.

1. **Hacer lugar:** Si la memoria esta llena, elige una “pagina victima” para sacar usando un algoritmo
2. **Guardar los cambios:** Si la “pagina victima” fue modificada, se guardan los cambios en disco antes de borrarla; sino, se elimina directamente
3. **Traer la nueva:** Pide al disco la pagina que falta, este pedido tiene prioridad absoluta y se atiende antes que otros
4. **Reinicia:** Actualiza la tabla y le dice al proceso que intente ejecutar la instruccion de nuevo

---

3.

Suponga que la tabla de páginas para un proceso que se está ejecutando es la que se
muestra a continuación:

Pagina	Bit V	Bit R	Bit M	Marco
0	1	1	0	4
1	1	1	1	7
2	0	0	0	-
3	1	0	0	2
4	0	0	0	-
5	1	0	1	0

Asumiendo que:

- El tamaño de la pagina es de 512 bytes.
- Cada direccion de memoria referencia 1 byte.
- Los marcos se encuentran contiguos y en orden en memoria (0, 1, 2...) a partir de la direccion fisica 0

❓ ¿Qué dirección física, si existe, correspondería a cada una de las siguientes direcciones virtuales? (No gestionar ningún fallo de página en caso de producirse)

a. 1052

$p = (1052/512) = 2,\ d = (1052) MOD (512) = 28$

Fallo de pagina, el bit V es 0

b. 2221

$p = (2221/512) = 4,\ d = (2221) MOD (512) = 173$

Fallo de pagina, el bit V es 0

c. 5499

$p = (5499/512) = 10,\ d = (5499) MOD (512) = 379$

Corresponde a una direccion ilegal

d. 3101

$p = (3101/512) = 6,\ d = (3101) MOD (512) = 29$

Corresponde a una direccion ilegal

---

4. Analice cómo impacta el tamaño de una página (pequeña o grande) en paginación por demanda.

Paginas grandes

Lo bueno

- Mas eficientes para el HW
- Menos interrupciones (menos PF)

Lo malo

- Se desperdicia memoria

Paginas chicas

Lo bueno

- Mucha precision (mejor ajuste)

- Se aprovecha cada byte de memoria

Lo malo

- El sistema pasa mucho tiempo detenido atendiendo PF

---

5. Asignacion Fija vs Asignacion Dinamica

a. Describa cómo trabajan estas 2 políticas.

Asignacion Flja:

El kernel decide de entrada cuantos marcos le tocan a cada proceso y ese numero no cambia durante la ejecucion. Hay dos formas de hacer este reparto

- **Reparto equitativo:** Se divide la memoria total por la cantidad de procesos y a todos le toca la misma cantidad
- **Reparto proporcional:** Es mas justo, se asignan marcos segun el tamaño del proceso

Asignacion Dinamica:

El kernel se adapta a las necesidades del momento, los procesos reciben los marcos que necesitan

- Se basa en **Working Set**: El sistema mira que paginas uso el proceso recientemente y le asigna los marcos necesarios para tener esas paginas cargadas
- Si la suma de los marcos que piden todos los procesos supera la memoria, se produce **thrashing**. Esta asignacion intenta evitar eso ajustando el tamaño del working set dinamicamente

b. Dada la siguiente tabla de paginas y teniendo 40 frames en memoria principal, cuantos marcos le corresponden a cada uno

Proceso	Total de Paginas Requeridas
1	15
2	20
3	20

Proceso	Total de Paginas Requeridas
4	8

Reparto equitativo:

- $M = 40$ (Marcos Disponibles)
- $N = 4$ (Numero de Procesos)

✅ $M_p = \frac{M}{N} = 10$

Podemos observar que solo el proceso 4 queda con Marcos ≥ Paginas, el resto no le alcanza

Reparto Proporcional:

🧮 Formula:
$\frac{VP \times m}{VT}$ Siendo **VP** necesidad del proceso, **m** la cantidad de marcos totales y **VT** la suma de todas las necesidades

$VT = 15 + 20 + 20 + 8 = 63$

$m = 40$

Proceso 1:

$\frac{15.40}{63} = 9$

Proceso 2:

$\frac{20.40}{63} = 13$

Proceso 3:

$\frac{20.40}{63} = 13$

Proceso 4:

$\frac{8.40}{63} = 5$

Proceso	Reparto Equitativo	Reparto Proporcional
1	10	9
2	10	13

Proceso	Reparto Equitativo	Reparto Proporcional
3	10	13
4	10	5

c. ¿Cual de los dos repartos usados en b) resulto mas eficiente?

A mi parecer el reparto proporcional llevara a una menor cantidad de fallos de pagina en comparacion al reparto equitativo.

---

6. Reemplazo de paginas

a. Clasifique estos algoritmos de malo a bueno de acuerdo a la tasa de fallos de página que se obtienen al utilizarlos.

FIFO

Es el mas simple y el menos eficiente, trata a los marcos como una cola circular y reemplaza la pagina mas vieja que entro en memoria. No tiene en cuenta localidad ni frecuencia de uso

Segunda Chance

Optimizacion del FIFO. Reduce la tasa de fallos comparado con FIFO ya que evita expulsar paginas que son referenciadas activamente

LRU

Genera una tasa de fallos baja y aceptable pero tiene una contra y es el overhead, osea, el costo de guardar el instante de ultoma referencia.

OPT

El algoritmo perfecto, con la tasa de fallos minima posible. Es imposible de implementar pero a nivel teorico es el algoritmo mas eficiente que se puede hacer

b. Analice su funcionamiento. ¿Cómo se podrían implementar?

FIFO

Se podria implementar con una cola circular

Segunda Chance

Se podria implementar con una cola circular y un bit que indique si el mismo ha sido referenciado

LRU

Se podria implementar guardando para cada pagina cuando fue accedida por ultima vez y calculando entre todas la mas lejana

OPT

Imposible de implementar, requiere concer el futuro.

c. Sabemos que la página a ser reemplazada puede estar modificada. ¿Cómo detecta el Kernel esta situación? ¿Qué acciones adicionales deben realizarse cuando se encuentra ante esta situación?

Cada pagina lleva un bit de modificacion, si el mismo se encuentra en 1 (la pagina fue modificada), el kernel antes de sacarla de memoria debe guardar su estado en disco; en caso de que el mismo este en 0 se elimina directamente

---

7. Alcance del reemplazo

a. Describa como trabaja reemplazo global y local

Reemplazo local:

El kernel se limita a seleccionar una pagina propia del proceso que genero el fallo. El proceso resuelve su falta de espacio desalojando una de sus paginas sin afectar al resto de procesos

Reemplazo global:

El kernel puede elegir como victima una pagina perteneciente a cualquier proceso del sistema. Osea, el kernel puede robarle un marco a otro proceso

b. ¿Es posible utilizar la política de “Asignación Fija” de marcos junto con la política de “Reemplazo Global"? Justifique.

Genera una contradiccion ya que la asignacion fija reparte equitativamente marcos a todos los procesos, que el kernel pueda robarle a otro proceso un marco dejaria de ser equitativo ya que los marcos de un proceso disminuirian y los de otro aumentarian

---

8. Considere la siguiente secuendia de referencias a paginas

> 👉🏽 1, 2, 15, 4, 6, 2, 1, 5, 6, 10, 4, 6, 7, 9, 1, 6, 12, 11, 12, 2, 3, 1, 8, 1, 13, 14, 15, 3, 8

FIFO

Marco / Pagina	1	2	15	4	6	2	1	5	6	10	4	6	7	9	1	6	12	11	12	2	3	1	8	1	13	14	15	3
F1	1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	6	6	6	6	6	6	1	1	1	1	1	1	3
F2		2	2	2	2	2	2	2	2	10	10	10	10	10	10	10	12	12	12	12	12	12	8	8	8	8	8	8
F3			15	15	15	15	15	15	15	15	15	15	7	7	7	7	7	11	11	11	11	11	13	13	13	13	13	13
F4				4	4	4	4	4	4	4	4	4	4	9	9	9	9	9	9	2	2	2	2	2	2	14	14	14
F5					6	6	6	6	6	6	6	6	6	6	1	1	1	1	1	1	3	3	3	3	3	3	15	15
PF?	X	X	X	X	X			X		X			X	X	X	X	X	X		X	X	X	X		X	X	X	X
FIFO																										# X	=	21

Segunda Chance

Marco / Pagina	1	2	15	4	6	2	1	5	6	10	4	6	7	9	1	6	12	11	12	2	3	1	8	1	13	14	15	3
F1	1	1	1	1	1	1	1*	1	1	1	4	4	4	4	4	4	12	12	12*	12*	12*	12*	12	12	12	12	15	15
F2		2	2	2	2	2*	2*	2	2	2	2	2	7	7	7	7	7	11	11	11	11	11	8	8	8	8	8	3
F3			15	15	15	15	15	5	5	5	5	5	5	9	9	9	9	9	9	2	2	2	2	2	13	13	13	13
F4				4	4	4	4	4	4	10	10	10	10	10	1	1	1	1	1	1	3	3	3	3	3	14	14	14
F5					6	6	6	6	6*	6*	6	6*	6*	6*	6*	6*	6	6	6	6	6	1	1	1*	1*	1*	1	1
PF?		X	X	X	X			X		X	X		X	X	X		X	X		X	X	X	X		X	X	X	X
Seg. Chance																										# X	=	20

LRU

Marco / Pagina	1	2	15	4	6	2	1	5	6	10	4	6	7	9	1	6	12	11	12	2	3	1	8	1	13	14	15	3
F1	1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	6	11	11	11	11	11	8	8	8	8	8	3
F2		2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	12	12	12	12	12	12	12	12	12	14	14	14
F3			15	15	15	15	15	5	5	5	5	5	5	9	9	9	9	9	11	2	2	2	2	2	13	13	13	13
F4				4	4	4	4	4	4	10	10	10	10	10	1	1	1	1	1	1	3	3	3	3	3	3	15	15
F5					6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1
PF?	X	X	X	X	X			X		X	X		X	X	X		X	X		X	X	X	X		X	X	X	X
LRU																										# X	=	21

OPT

Marco / Pagina	1	2	15	4	6	2	1	5	6	10	4	6	7	9	1	6	12	11	12	2	3	1	8	1	13	14	15	3
F1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	13	13	13	13
F2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	8	8	8	8	8	8
F3			15	15	15	15	15	5	5	10	10	10	10	9	9	9	9	9	9	9	9	9	9	9	9	14	14	14
F4				4	4	4	4	4	4	4	4	4	7	7	7	7	7	7	7	7	3	3	3	3	3	3	3	3
F5					6	6	6	6	6	6	6	6	6	6	6	6	6	11	11	11	11	11	11	11	11	11	15	15
PF?	X	X	X	X	X			X		X			X	X			X	X	X		X		X		X	X	X	
OPT																										# X	=	17

Algoritmo	Tiempo (PF / 10)
FIFO	2,1 seg
Segunda Chance	2 seg
LRU	2,1 seg
OPT	1,7 seg

9. Sean los procesos A, B y C tales que necesitan para su ejecución las siguientes páginas:

🤔

- **A:** 1, 3, 1, 2, 4, 1, 5, 1, 4, 7, 9, 4
- **B:** 2, 4, 6, 2, 4, 1, 8, 3, 1, 8
- **C:** 1, 2, 4, 8, 6, 1, 4, 1

Inciso A

a. LRU

Proceso	B		A			C		B			A			C			B		C			A			C			B			A			C			A			B	A
Pag Global	2	4	1	3	1	1	2	6	2	4	2	4	1	4	8	1	8	6	1	4	1	5	1	4	3	1	8	----	7	9	4	----	----
F1	B2	B2	B2	B2	B2	B2	B2	B2	B4	A3	A1	C1	C2	B6	B2	B4	A2	A4	A1	A1	A1	C8	B1	B8	C6	C4	C1	A5	A5	A1	A1	A1	
F2		B4	B4	B4	B4	B4	B4	B4	A3	A1	C1	C2	B6	B2	B4	A2	A4	A1	C4	C8	C8	B1	B8	C6	C4	C1	A5	A1	A1	A4	B3	A7	
F3			A1	A1	A3	A3	A3	A3	A1	C1	C2	B6	B2	B4	A2	A4	A1	C4	C8	B1	B1	B8	C6	C4	C1	A5	A1	A4	A4	B3	B1	A9	
F4				A3	A1	A1	A1	A1	C1	C2	B6	B2	B4	A2	A4	A1	C4	C8	B1	B8	B8	C6	C4	C1	A5	A1	A4	B3	B3	B1	B8	A4	
F5						C1	C1	C1	C2	B6	B2	B4	A2	A4	A1	C4	C8	B1	B8	C6	C6	C4	C1	A5	A1	A4	B3	B1	B1	B8	A7		
F6							C2	C2	B6	B2	B4	A2	A4	A1	C4	C8	B1	B8	C6	C1	C4	C1	A5	A1	A4	B3	B1	B8	B8	A7	A9		
F7								B6	B2	B4	A2	A4	A1	C4	C8	B1	B8	C6	C1	C4	C1	A5	A1	A4	B3	B1	B8		A7	A9	A4		
PF?	X	X	X	X		X	X	X			X	X	X	X	X			X	X			X	X	X	X	X	X		X	X			
LRU	# X =																															22	

b. Segunda Chance

Proceso	B		A			C		B			A			C		B		C			A			B			C			A			B	A
Pag Global	2	4	1	3	1	1	2	6	2	4	2	4	1	4	8	1	8	6	1	4	1	5	1	4	3	1	8	----	7	9	4	----	----
F1	B2	B2	B2	B2	B2	B2	B2	B2	**B2***	**B2***	B2	B2	B2	B2	B2	B1	B1	B1	B1	B1	B1	B1	B1	A4	A4	A4	A4	A4	A4	A9	A9	A9	
F2		B4	B4	B4	B4	B4	B4	B4	B4	**B4***	B4	B4	B4	B4	B4	B4	B8	B8	B8	B8	B8	B8	B8	B8	B3	B3	B3	B3	B3	B3	A4	A4		
F3			A1	A1	**A1***	**A1***	**A1***	**A1***	**A1***	**A1***	A1	A1	**A1***	**A1***	**A1***	**A1***	**A1***	A1	A1	A1	A1	A1	**A1***	**A1***	**A1***	A1	A1	A1	A1	A1	A1	A1		
F4				A3	A3	A3	A3	A3	A3	A3	A2	A2	A2	A2	A2	A2	A2	C6	C6	C6	C6	C6	C6	C6	C6	B1	B1	B1	B1	B1	B1	B1	A5		
F5						C1	C1	C1	C1	C1	C1	A4	A4	A4	A4	A4	A4	A4	C1	C1	**C1***	**C1***	**C1***	**C1***	**C1***	**C1***	**C1***	C1	B8	B8	B8	B8	A7		
F6							C2	C2	C2	C2	C2	C2	C2	C4	C4	C4	C4	C4	C4	**C4***	**C4***	C4	C4	C4	C4	C4	B8	A5	A5	A5	A5				
F7								B6	B6	B6	B6	B6	B6	B6	C8	C8	C8	C8	C8	C8	C8	A5	A5	A5	A5	A5	A5		A7	A7	A7				
PF?	X	X	X	X		X	X	X			X	X		X	X		X	X	X			X		X	X	X	X		X	X	X				
Seg. Chance																													# X =			22			

(Los page fault a chequearlos la verdad)

Inciso B

3 frames para cada uno

a. LRU

Proceso A	1	3	1	2	4	1	5	1	4	7	9	4
F1	1	1	1	1	1	1	1	1	1	1	9	9
F2		3	3	3	4	4	4	4	4	4	4	4
F3				2	2	2	5	5	5	7	7	7
PF?	X	X		X	X		X			X	X	
LRU												7

Proceso B	2	4	6	2	4	1	8	3	1	8
F1	2	2	2	2	2	2	8	8	8	8
F2		4	4	4	4	4	4	3	3	3
F3			6	6	6	1	1	1	1	1
PF?	X	X	X			X	X	X		
LRU										6

Proceso C	1	2	4	8	6	1	4	1
F1	1	1	1	8	8	8	4	4
F2		2	2	2	6	6	6	6
F3			4	4	4	1	1	1
PF?	X	X	X	X	X	X	X	
LRU								7

b. Segunda Chance

Proceso A	1	3	1	2	4	1	5	1	4	7	9	4
F1	1	1	1*	1*	1	1*	1*	1*	1*	1	9	9
F2		3	3	3	4	4	4	4	4*	4	4	4*
F3				2	2	2	5	5	5	7	7	7
PF?	X	X		X	X		X			X	X	
Seg. Chance												7

Proceso B	2	4	6	2	4	1	8	3	1	8
F1	2	2	2	2*	2*	2	8	8	8	8*
F2		4	4	4	4*	4	4	3	3	3
F3			6	6	6	1	1	1	1*	1*
PF?	X	X	X			X	X	X		
Seg. Chance										6

Proceso C	1	2	4	8	6	1	4	1
F1	1	1	1	8	8	8	4	4
F2		2	2	2	6	6	6	6
F3			4	4	4	1	1	1*
PF?	X	X	X	X	X	X	X	
Seg. Chance								7

11. Hiperpaginacion (Trashing)

a. ¿Qué es?

La hiperaginacion es un estado en el cual el SO pasa mas tiempo resolviendo fallos de pagina que ejecutando instrucciones. El rendimiento del sistema baja debido a que el CPU se cuelga esperando que se resuelvan los fallos

b. ¿Cuáles pueden ser los motivos que la causan?

Ocurre cuando la demanda total de marcos supera la cantidad de marcos fisicos disponibles en memoria

c. ¿Cómo la detecta el SO?

Usa la estrategia de working set, siguiendo estos pasos

1. Define una ventana de tiempo
2. Calcula el working set de cada proceso
3. Calcula la demanda total de todos los working set
4. Hay hiperpaginacion si la demanda total es mayor a la cantidad de marcos disponibles

d. Una vez que lo detecta, ¿qué acciones puede tomar el Kernel para eliminar este problema?

El kernel debe reducir la demanda total para que sea menor o igual a los marcos disponibles

El kernel debe garantizar que los procesos activos tengan asignados marcos suficientes para cubrir sus working sets

---

12. Analice y compare las técnicas de PFF (Page Fault Frequency) y del Working Set. ¿Como las mismas permiten determinar la existencia de trashing?

El working set lo que hace es definir una ventana de tiempo y calcular el working set en base a cuantas paginas unicas se referenciaron en ese tiempo. Permite determinar la existencia de trashing cuando la demanda total de marcos que es la suma de todos los working set de procesos activos es mayor a la cant de marcos totales

EL PFF es una estrategia para prevenir la hiperapaginacion usando una tasa de fallos de pagina, poniendo un limite superior e inferior de fallos de pagina

- Si se pasa del limite superior, asigna mas marcos al proceso
- SI se pasa del limite inferior, le puede sacar marcos al proceso para darselos a otro

SI todos estan superando su limite superior, se debe suspender el proceso

---

13.

Considere un sistema cuya memoria principal se administra mediante la técnica de paginación por demanda que utiliza un dispositivo de paginación, algoritmo de reemplazo global LRU y una política de asignación que reparte marcos equitativamente entre los procesos. El nivel de multiprogramación es actualmente, de 4.
Ante las siguientes mediciones:

a. Uso de CPU del 13%, uso del dispositivo de paginación del 97%.

b. Uso de CPU del 87%, uso del dispositivo de paginación del 3%.

c. Uso de CPU del 13%, uso del dispositivo de paginación del 3%.

Analizar para cada caso:

- ¿Que sucede?
- ¿Puede incrementarse el nivel de multiprogramación para aumentar el uso de la CPU?
- ¿La paginación está siendo útil para mejorar el rendimiento del sistema?

Caso A. CPU 13%, Dispositivo de Paginación 97%

- El sistema se encuentra en hiperapginacion, el procesador esta tranquilo mientras que el ssitema pasa mas tiempo resolviendo PF que ejecutando instrucciones utiles
- No se puede incrementar la multiprogramacion porque esto agravaria el problema del dispositivo de paginacion, la tasa de fallos aumentaria mas y el cpu trabajaria menos
- La paginacion no esta siendo util, el overhead de la paginacion degrada severamente el rendimiento

Caso B: CPU 87%, Dispositivo de Paginación 3%

- El sistema funciona bien, la cpu tiene un alto uso por lo que los procesos estan ejecutandose bien. El uso del disp. de paginacion es bajo por lo que la tasa de fallos de pagina es cercana a cero
- La CPU esta bastante ocupada como para agregarle mas procesos, esto podria saturarla
- La paginacion esta siendo util ya que permite la ejecucion fluida de los procesos. EL sistema aprovecha la memoria virtual correctamente

Caso C: CPU 13%, Dispositivo de Paginación 3%

- El sistema esta trabajando a mucho menor ritmo del que pdoria, la mayor parte del tiempo esta al pedo. No hay trashing ya que el disp. de paginacion casi no se usa
- Es recomendable aumentar la multiprogramacion para aumentar el uso de cpu

- La paginacion esta presente pero al no haber tantos procesos no se exprime su potencial

---

14.

Considere un sistema cuya memoria principal se administra mediante la técnica de
paginación por demanda. Considere las siguientes medidas de utilización.

- Utilizacion del procesador: 20%
- Utilización del dispositivo de paginación: 97,7%
- Utilización de otros dispositivos de E/S: 5%

Cuales de las siguientes acciones pueden mejorar la utilización del procesador:

a. Instalar un procesador más rápido
b. Instalar un dispositivo de paginación mayor
c. Incrementar el grado de multiprogramación
d. Instalar mas memoria principal
e. Decrementar el quantum para cada proceso

Lo mejor seria instalar mas memoria principal para tener mas marcos disponibles y reducir la necesidad de paginacion

---

15.

La siguiente fórmula describe el tiempo de acceso efectivo a la memoria al utilizar paginación para la implementación de la memoria virtual:

$$TAE = At + [(1 - p) * Am] + [p * (Tf + Am)]$$

Donde:

- TAE = tiempo de acceso efectivo
- p = tasa de fallo de página (0 <= p <=1)
- Am = tiempo de acceso a la memoria real

- Tf = tiempo se atención de una fallo de pagina
- At = tiempo de acceso a la tabla de páginas. Es igual al tiempo de acceso a la memoria (Am) si la entrada de la tabla de páginas no se encuentra en la TLB.

Suponga que tenemos una memoria virtual paginada, con tabla de páginas de 1 nivel, y donde la tabla de páginas se encuentra completamente en la memoria. Servir una falla de página tarda 300 nanosegundos si hay disponible un marco vacío o si la página reemplazada no se ha modificado, y 500 nanosegundos si se ha modificado. El tiempo de acceso a memoria es de 20 nanosegundos y el de acceso a la TLB es de 1 nanosegundo

a. Si suponemos una tasa de fallos de página de 0,3 y que siempre contamos con un marco libre para atender el fallo ¿Cuál será el TAE si el 50% de las veces la entrada de la tabla de páginas se encuentra en la TLB (hit)?
b. Si suponemos una tasa de fallos de página de 0,3; que el 70% de las ocasiones la página a reemplazar se encuentra modificada. ¿Cuál será el TAE si el 60% de las veces la entrada de la tabla de páginas se encuentra en la TLB (hit)?
c. Si suponemos que el 60% de las veces la página a reemplazar está modificada, el 100% de las veces la entrada de la tabla de páginas requerida se encuentra en la TLB (hit) y se espera un TAE menor a 200 nanosegundos.
d. ¿Cuál es la máxima tasa aceptable de fallas de página?

a.

$$At = (0,5 \times 1\ ns) + (0,5 \times 21\ ns)$$

$$At = 0,5 + 10,5$$

$$At = 11\ ns$$

Formula:

$$TAE = 11 + [(1 - 0,3) * 20] + [0,3 * (300 + 20)]$$

$$TAE = 11 + 14 + 96$$

$$TAE = 121\ nanosegundos$$

b.

$$At = (0,6 \times 1\ ns) + (0,4 \times 21\ ns)$$

$$At = 0,6 + 8,4$$

$$At = 9\ ns$$

Calculo de T_f

$$Tf = (0,7 \times 500\ ns) + (0,3 \times 300\ ns)$$

$$Tf = 350 + 90$$

$$Tf = 440\ ns$$

Formula:

$$TAE = 9 + [(1 - 0,3) \times 20] + [0,3 \times (440 + 20)]$$

$$TAE = 9 + 14 + 138$$

$$TAE = 161\ nanosegundos$$

c.

$$At = 1\ ns$$

Calculo de T_f

$$Tf = (0,6 \times 500ns) + (0,6 \times 300ns)$$

$$Tf = 300 + 120$$

$$Tf = 420\ ns$$

Querems que TAE sea menor a 200

$$1 + [(1 - p) * 20] + [p * (420 + 20)] < 200$$

$$21 + 420p < 200$$

$$420p < 179$$

$$p < 0,42619$$

16.Considere el siguiente programa:

```
#define Size 64
int A[Size; Size], B[Size; Size], C[Size;
Size]; int register i, j;
for (j = 0; j < Size; j ++)
    for (i = 0; i < Size; i++)xx
        C[i; j] = A[i; j] + B[i; j];
```

Si asumimos que el programa se ejecuta en un sistema que utiliza paginación por demanda para administrar la memoria, donde cada página es de 1Kb. Cada número entero (int) ocupa 4 bytes. Es claro que cada matriz requiere de 16 páginas para almacenarse. Por ejemplo: A[0,0]..A[0,63], A[1,0]..A[1,63], A[2,0]..A[2,63] y A[3,0]..A[3,63] se almacenará en la primera página. Asumamos que el sistema utiliza un working set de 4 marcos para este proceso. Uno de los 4 marcos es utilizado por el programa y los otros 3 se utilizan para datos (las matrices). También consideramos que para los índices “i” y “j” se utilizan 2 registros, por lo que no es necesario el acceso a la memoria para estas 2 variables.

a. Analizar cuántos fallos de páginas ocurren al ejecutar el programa (considere las veces que se ejecuta C[i,j] = A[i,j] + B[i,j])
b. Puede ser modificado el programa para minimizar el número de fallos de páginas. En caso de ser posible indicar la cantidad de fallos de páginas que ocurren.

a.

Se cargan las paginas 0 de A, B y C. que contiene las filas 0..3, cuando quiera ir a la cuarta le va a pasar que no esta, entonces cada 4 paginas vemos que hay 3 PF, uno por cada matriz.

$$64/4 = 16, \quad 16 \; paginas \; \times \; 3 \; matrices = 48 \; PF$$

Todo esto se va a ejecutar 64 veces por el bucle con el j, asi que esto da

$$48 \; PF \; \times \; 64 \; iteraciones = 3072 \; PF$$

b.

La manera de resolverlo seria cambiar el for grande por el chico, asi se aprovecha al maximo la pagina actual, esto reduciria a 48 PF totales

---

17.

Considere las siguientes secuencias de referencias a páginas de los procesos A y B, donde se muestra en instante de tiempo en el que ocurrió cada una (1 a 78):

Proceso A																																						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	2	1	3	4	3	3	2	1	1	3	4	5	6	1	2	1	3	3	4	5	6	6	6	5	4	3	1	1	2	3	4	5	4	3	2	1	1	1
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
2	2	1	2	1	1	2	3	4	5	6	7	8	6	5	3	1	2	3	4	5	6	5	1	1	2	3	4	5	4	2	1	1	2	3	4	5	1	1

Proceso B																																						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
1	2	3	3	3	4	4	5	1	1	1	2	3	4	4	4	4	1	1	2	3	6	5	6	5	4	6	1	1	1	1	4	5	1	3	2	1	1	2
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
2	2	3	3	4	4	4	3	4	5	4	6	1	1	1	2	3	3	3	4	5	5	5	1	6	2	3	1	2	3	1	1	1	2	2	2	2	3	3

a. Considerando una ventana Δ=5, indique cuál sería el conjunto de trabajo de los procesos A y B en el instante 24 (WSA(24) y WSB(24))

$$WS_A(24) = [20..24]$$

A = { 6, 6, 6, 5, 4 } → Solo paginas unicas = A { 4, 5, 6 }

$$WS_B(24) = [20..24]$$

B = { 6, 5, 6, 3, 2 } → Solo paginas unicas = B { 2, 3, 5, 6 }

b. Considerando una ventana Δ=5, indique cuál sería el conjunto de trabajo de los procesos A y B en el instante 60

(WSA(60) y WSB(60))

$$WS_A(24) = [56..60]$$

A = { 5, 4, 3, 2, 1 } → Solo paginas unicas = A { 5, 4, 3, 2, 1}

$$WS_B(24) = [56..60]$$

B = { 5, 4, 3, 3, 3 } → Solo paginas unicas = B { 3, 4, 5 }

18.

Los dispositivos se pueden clasificar segun su unidad de transferencia

Orientado a bloques

- La informacion se transfiere en bloques de tamaño fijo
- Soporta operaciones de R/W/S, osea que son direccionables

Orientado a flujos

- La informacion se transfiere como flujo de bytes, sin una estructura de bloques definida
- Soporta operaciones de Get, Put y no soportan seek

Ejemplos

Orientado a bloques: Discos rigidos, CD-ROM, DVD

Orientado a Flujos: Teclado, mouse

---

19. Técnicas de E/S. Describa cómo trabajan las siguientes técnicas de E/S

a. E/S programada

La cpu tiene el control directo sobre la operacion de E/S, se encarga de controlar el estado, enviar comandos de R/W

- La cpu emite un comando (ej: read) al modulo de E/S y permanece a cargo de todo el proceso

- La cpu chequea constantemente el estado del disp. de E/S, osea si esta listo, ocupado o huvo un error
- Una vez que

---

27. Describa en forma sintética, cómo es la organización física de un disco, puede utilizar gráficos para mayor claridad.

Un disco rigido esta compuesto por uno o mas platos rigidos que giran sobre un eje

Caras y cabezas

Cada plato tiene 2 caras, existe una cabeza de R/W para cada cara

Pistas

Cada cara se divide en anillos concentricos que se llaman pistas

Sectores

Cada pista se divide en segmentos que se llaman sectores, es la unidad minima de transferencia de datos

Cilindro

Es el conjunto de todas las pistas que tienen el mismo numero a traves de todas las caras. Osea el disco puede acceder a cualquier pista de un mismo cilindro sin mover el brazo

28. La velocidad promedio para la obtención de datos de un disco está dada por la suma de los siguientes tiempos:

La velocidad promedio para la obtención de datos de un disco está dada por la suma de los siguientes tiempos:

Seek Time

Tiempo que tarda el brazo del disco en mover y posicionar la cabeza de R/W sobre el cilindro deseado

Latency Time

Tiempo que transcurre desde que la cabeza ya se posiciono donde tiene que estar hasta que el sector pasa por abajo

Transfer time

Tiempo necesario para leer y escribir los datos efectivamente, osea transferir el sector a memoria principal

---

29.Suponga un disco con las siguientes características:

- 7 platos con 2 caras utilizables cada uno.
- 1100 cilindros, con 300 sectores por pista, donde cada sector es 512 bytes.
- Seek Time de 10 ms
- 9000 RPM.
- Velocidad de Transferencia de 10 MiB/s (Mebibytes por segundos).

a. Calcule la capacidad total del disco.

$$7 \times 2 \times 1100 \times 300 \times 512\ bytes = 2.365.440.000\ bytes$$

b. ¿Cuántos sectores de disco se requerirían si se quiere almacenar 3 MiB(MebiBytes) de datos?

$$1MiB = 2^{20} = 1.048.576\ bytes$$

$$3 \times 1.048.576 = 3.145.728\ bytes$$

$$\frac{3.145.728}{512} = 6.144$$

Se necesitan 6144 sectores

c. Calcule el tiempo de latencia.

Tiempo de una vuelta completa: 9000 → 60.000 milisegundos

$$1\ vuelta = \frac{60.000}{9000} = 6,66...milisegundos$$

Latencia promedio: Tiempo en dar media vuelta = 3,33... milisegundos

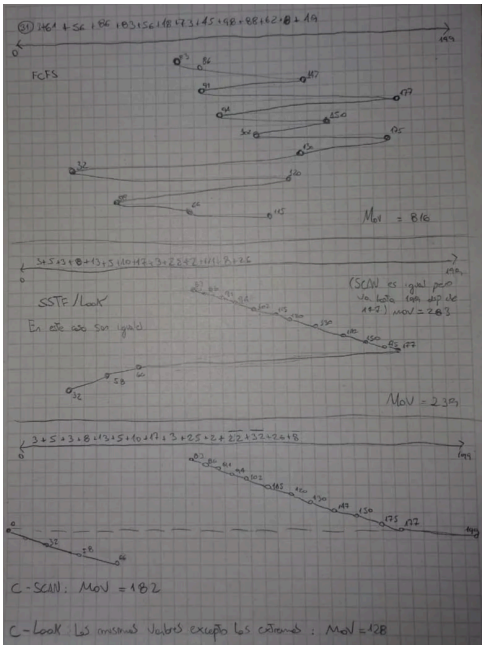
---

31.

El Seek Time es el parámetro que posee mayor influencia en el tiempo real necesario para transferir datos desde o hacia un disco. Es importante que el Kernel planifique los diferentes requerimientos al disco para minimizar el movimiento de la cabeza lecto-grabadora.
Analicemos las diferentes políticas de planificación de requerimientos a disco con un ejemplo: Supongamos un Head con movimiento en 200 tracks (numerados de 0 a 199), que está en el track 83 atendiendo un requerimiento y anteriormente atendió un requerimiento en el track 75.

Si la cola de requerimientos es: {86, 147, 91, 177, 94, 150, 102, 175, 130, 32, 120, 58, 66, 115}. Realice los diagramas para calcular el total de movimientos de head para satisfacer estos requerimientos de acuerdo a los siguientes algoritmos de scheduling de discos:

a. FCFS
b. SSTF
c. Scan
d. Look
e. C-Scan
f. C-Look



32.¿Alguno de los algoritmos analizados en el ejercicio anterior pueden causar inanición de requerimientos?

El SSTF podria provocar inanicion