Final de Sistemas Operativos

11/02/2020

Nota:

Apellido y Nombre	*Profesor*	*Tomé conocimiento de la nota: (Sólo aplazos)*

Preguntas teóricas					Ejercicios	
1	2	3	4	5	1	2

A) Teoría: Defina explícitamente como VERDADERA o FALSA cada una de estas afirmaciones justificando brevemente.

1) Para no llegar a un estado de sobrepaginación (thrashing) la mejor opción es utilizar asignación de frames fija con sustitución de páginas local.

2) En caso de utilizar un algoritmo de planificación con desalojo, al llegar una interrupción de fin de IO de un KLT la misma será atendida sólo si dicho proceso tiene mayor prioridad (dependiendo el algoritmo que utilice) que el thread en ejecución y amerita el cambio.

3) Si la ejecución de las syscalls wait y signal no se realizara en forma atómica generaría condición de carrera.

4) En algunos casos, como por ejemplo semáforos con espera activa, puede darse un deadlock sin que ocurran las 4 condiciones.

5) Una entrada salida asíncrona puede realizarse de forma síncrona en un ULT distinto y, para el proceso que la realiza, el resultado será el mismo.

B) Práctica: Resuelva los ejercicios justificando las respuestas

1) Un sistema utiliza como planificador de corto plazo VRR con Q = 3 y EXT como FS (con ptrs de 64 bits, bloques de 8KiB, 10 ptrs dir, 1 ind simple, 2 ind dobles). En dicho sistema se pone a correr una simulación de un temporizador; como resultado final se escribe al final del archivo temporizador.txt (cuyo tamaño es de 7654321 bytes) "Llegó a 0!". Como se detecta que existe algún problema se realiza una nueva simulación con **total 1** para poder analizar más fácilmente el mismo.

Proc Decrementadores (PD) (2 instancias)

```
{
  while(1) {    1 CPU
    local t = total    1 CPU
    t --    1 CPU
    log (t)    1 CPU + 2 IO
    total = t    1 CPU
    if (t == 0) {    2 CPU
       signal(semCero)  2 CPU
    }
  }
}
```

Proc informador (PI) (1 instancia)

```
{
  wait(semCero)    2 CPU
  guardarResultado("Llegó a 0!", temporizador.txt )    1 CPU + 1 IO
  finalizarProcesos()    3 CPU
}
```

Variables globales: semCero = 0 // total = 1
La función "finalizarProcesos" finaliza a ambas instancias de los procesos decrementadores y a sí mismo, se ejecuta en forma atómica..
La ejecución de wait y signal son atómicas y bloqueantes.

a) Realizar el Gantt de la simulación de procesos sabiendo que inicialmente en ready se encuentran: PD1 - PI - PD2
b) Analice el pseudocódigo e indique qué tipo de semáforo es **semCero** y para qué se utiliza. ¿Qué problema presenta el código que genera resultados erráticos? ¿Cómo podría solucionarlo?
c) ¿Cuántos bloques se accedieron en temporizador.txt?

2) Un conjunto de procesos se ejecutan en un sistema que tiene 6 frames de memoria totales. Dicho sistema utiliza paginación por demanda, con asignación dinámica y alcance global. Se desconoce el algoritmo, pero se sabe que los frames se asignan en orden ascendente y que esta es una traza de ejecución reciente, para los procesos A, B y C (siendo el número que acompaña la página accedida):

Frame	A1	A2	B1	B2	A3	B1	B0	C2	C1	A1	A3	A2	B2
0	A1	A1	A1	A1	A1	A1	A1	A1	C1	C1	C1	C1	?
1	C3	A2	A2	A2	A2	A2	A2	A2	A2	A1	A1	A1	?
2	B1	B1	B1	B1	A3	A3	A3	A3	A3	A3	A3	A2	?
3	B2	B2	B2	B2	B2	B1	B1	B1	B1	B1	B1	B1	?
4	C0	C0	C0	C0	C0	C0	B0	B0	B0	B0	B0	B0	?
5	C1	C1	C1	C1	C1	C1	C1	C2	C2	C2	C2	C2	?

Indique:
- Qué algoritmo se puede estar usando, completando la última columna (para el pedido B2).
- Si hay algún indicio de que algún proceso (o el sistema) puedan estar en thrashing (sobrepaginación), sabiendo que la localidad de cada proceso es siempre de 3 páginas.

El tiempo de duración del examen final será de 90' a contar desde el momento de comienzo del mismo. Si el alumno por algún motivo comenzará más tarde sólo podrá utilizar el tiempo remanente. Utilice hojas separadas para la teoría / ejercicios.

Final de Sistemas Operativos

11/02/2020

Nota:

Resolución

Teoría

1) Falso. Realmente no es una opción ya que también podría ocurrir sobrepaginación si es que no asignamos la cantidad de frames necesaria para que se acomode la localidad.

2) Falso. La interrupción siempre será atendida al finalizar el ciclo de instrucción actual. Lo que puede cambiar es que el planificador decida replanificar el proceso en ejecución y prefiera desalojarlo para ejecutar el que acaba de pasar a estado nuevo.

3) Verdadero. Las syscalls wait y signal manipulan un recurso común que es el semáforo tanto en forma lectura como de escritura, por lo que aplican las condiciones de bernstein.

4) Falso. Si no se cumplen las 4 condiciones, no hay deadlock.

5) Falso. Al bloquearse el ULT de la E/S, se bloquearía el proceso completo, y el resultado sería similar a usar una E/S asíncrona. La única salvedad sería usar jacketing.

Práctica

1)

a)

PD1	Q = 3									IO	IO	Q ‘= 2			Q = 3			PI READY		
PD2						Q = 3						IO	IO	Q ‘= 2				Q = 3		
PI						en espera														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	17	19	20

PD1		Q = 3								
PD2										
PI		IO			Q = 3			Fin procesos		
20	21	22	23	24	25	26	27	28	29	30

b) **semCero** es un semáforo binario, se utiliza para ordenar la ejecución y evitar que el proceso informador ejecute antes de que el temporizador llegue a 0. Por otro lado, hay un problema de condición de carrera sobre la variable global **total**. La misma es accedida en modo lectura y escritura en forma concurrente por los 3 procesos generando resultados distintos dependiendo del orden de ejecución. Por ejemplo, en el gantt podemos ver que se decrementó 2 veces el contador, pero **total** quedó en cero. Esto se puede solucionar utilizando un semáforo mutex cada vez que se acceda a dicha variable. En el caso del proceso Decrementador tanto la lectura como la modificación y seteo de la variable deben estar en la misma RC para que evitemos las condiciones de carrera.

c) Se escribe al final del archivo cuyo tamaño es 7654321 bytes = 934,36 => quiere decir que el archivo tiene 935 bloques de datos asignados (ya que la asignación es a nivel de bloques). En el último bloque quedan 5199 bytes, por lo que alcanza para escribir “Llegó a 0!” sin tener que asignar otro bloque.
Necesitamos calcular si se requiere acceder a algún bloque de punteros para escribir en el último bloque (bloque 934).
Como el bloque es de 8KiB y el ptr es de 8 bytes => 1024 ptrs por bloque
Por lo tanto, con el primer puntero directo es que podremos encontrar el bloque en cuestión.
Tendremos entonces **2 accesos a bloques**, uno en modo lectura (el de ptr) y otro en modo escritura (el de datos)

2)

- El algoritmo usado es FIFO, por lo que la columna se completa reemplazando el frame 3 con B2
- El sistema está en thrashing, porque prácticamente todos los pedidos ocasionan un fallo. Los procesos a su vez están en thrashing, porque ninguno logra tener su localidad presente en memoria principal al momento de hacer un acceso.

El tiempo de duración del examen final será de 90’ a contar desde el momento de comienzo del mismo. Si el alumno por algún motivo comenzará más tarde sólo podrá utilizar el tiempo remanente. Utilice hojas separadas para la teoría / ejercicios.