Planificación y Gestión de procesos

1

Indice

- Objetivos de planificación
- Algoritmos de planificación
- Detección y prevención de bloqueos

2

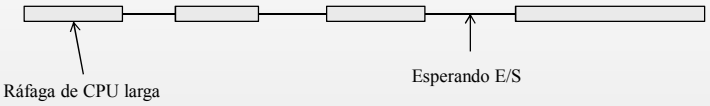
Objetivos de planificación

- El S.O debe decidir qué proceso ejecutar de aquellos que están listos para ejecución
- ***Planificador (scheduler):*** parte del S.O que realiza la decisión mediante el **algoritmo de planificación**
- Objetivos:
  - Equidad
  - Eficiencia
  - Tiempo de respuesta
  - Minimizar tiempo de proceso global
  - Maximizar el nº de trabajos procesados
- Pueden ser contradictorios

3

Comportamiento de los procesos

- Es único e impredecible
- Alternancia de cálculos y E/S
  - Mucho cálculo (CPU bound / limitado a cálculos)



  - Mucha E/S (I/O bound / limitado a E/S)

4

¿Cuándo planificar?

- Al crear un proceso
- Cuando un proceso termina
- Cuando el proceso en ejecución se bloquea (E/S, semáforo …)
- Cuando llega una interrupción
  - Que hace que un proceso que estaba bloqueado pase a listo para ejecución

5

Planificación

- Para asegurar que un proceso no monopoliza el procesador
  - Temporizador hardware
  - Con cada interrupción el S.O toma el control y decide si seguir con el proceso actual o pasar a otro
- Planificación no apropiativa (nonpreemptive):
  - Selecciona un proceso que se ejecuta hasta que se bloquea, termina o libera la CPU de forma voluntaria.
- Planificación apropiativa (preemptive)
  - Selecciona un proceso y deja que se ejecute durante un cierto tiempo.

6

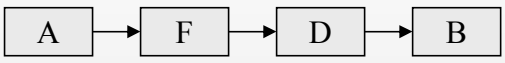
Algoritmos de planificación

- Se pueden utilizar distintos algoritmos de planificación:
  - Planificación por turno rotatorio
  - Planificación por prioridad
  - Lista de espera múltiples
  - Prioridad al más corto
  - Planificación determinada por el comportamiento
  - Planificación a dos niveles

7

Planificación por turno rotatorio

- A cada proceso se le asigna un intervalo de tiempo o cuanto



Lista de procesos

- Cambio de contexto lleva tiempo
  - Si 5 ms, el tiempo utilizado en el cambio:
    - Si cuanto de 20 ms => 20%
    - Si cuanto de 500 ms => 1%
      - Si 10 procesos, 1/2 s, 1 s … en empezar.
- Compromiso entre eficiencia del procesador y velocidad de respuesta

8

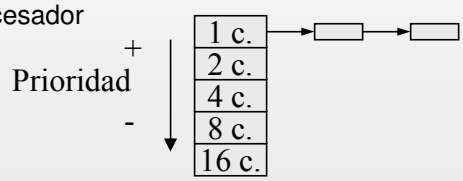
Planificación por prioridad

- Se asignan prioridades a los procesos y se ejecuta el de más alta prioridad
- Se pueden agrupar por prioridades y utilizar turno rotatorio para cada prioridad
- asignación estática o dinámica
- Ejemplo: prioridad = 1/f ;
  f= T utilizado del último cuanto/ T del cuanto
  - 2 ms de 100 => prioridad = 50
  - 50 ms de 100 => prioridad = 2
  - 100 ms de 100 => prioridad = 1

9

Múltiples listas de espera

- Para sistemas que sólo pueden mantener en memoria un proceso
- Baja el nº de transferencias
  - Si un proceso agota su cuanto, se le pasa a la categoría inferior. A los procesos con más cálculo se le da más tiempo de procesador



- Proceso que necesita 100 cuantos:
  - 1 2 4 8 16 32 64
  - Accede cada vez menos frecuentemente al procesador

10

Prioridad al más corto

- Para sistemas por lotes
- Se conoce el tiempo de respuesta de los procesos

A	B	C	D	
8	4	4	4	T. Ejecución
8	12	16	20	T. Respuesta

T. promedio= 14

D	B	C	A	
4	4	4	8	T. Ejecución
4	8	12	20	T. Respuesta

T. promedio= 11

A	B	C	D
ta	tb	tc	td
ta	ta+tb	ta+tb+tc	ta+tb+tc+td

T. promedio= 4ta+3tb+2tc+1td

11

Planificación determinada por el comportamiento

- n usuarios => 1/n de la potencia del procesador
- Se contabiliza:
  - tiempo de procesador consumido
  - tiempo tanscurrido = t. que lleva el proceso en el sistema
- Se calcula:
  - t. al que se tiene derecho
    - t. transcurrido/n
- Se ejecuta el proceso con el valor más bajo para:
  - t. de procesador consumido/ t. al que se tiene derecho
    - 100 / 200 = 0.5
    - 200 / 100 = 2

12

Planificación a dos niveles

- Planificador de alto nivel
  - Se encarga de llevar procesos de disco a memoria y viceversa
- Planificador de bajo nivel
  - Se encarga de pasar de un proceso a otro en memoria principal
- Varios criterios
  - Tiempo en memoria
  - Tiempo de procesador
  - Prioridad
  - Tamaño

13

Planificación de hilos

- Hilos a nivel de usuario (No apropiativa)
  - El S.O planifica procesos
  - Se conmuta de un hilo a otro dentro del mismo proceso cuando el hilo en ejecución termina o cede el procesador a otro hilo (con una llamada especial del paquete de hilos)
  - Si el hilo se bloquea por E/S, el proceso entero se bloquea
- Hilos a nivel de kernel (apropiativa)
  - El S.O planifica hilos
  - Se concede un tiempo a cada hilo independientemente del proceso al que pertenezcan
  - Si el hilo en ejecución se bloquea se puede ejecutar otro hilo del mismo proceso o de otro proceso.
  - El cambio de contexto a un hilo de otro proceso es más costoso

14

Detección y prevención de bloqueos

- Interbloqueos
  - Se dan por la necesidad de uso exclusivo de algunos recursos por parte de los procesos
- Ejemplo: proceso A necesita recurso 1 y 2
  proceso B necesita recurso 2 y 1
  - proceso A pide 1
  - proceso B pide 2
  - proceso A pide 2 => se bloquea
  - proceso B pide 1 => se bloquea
  - Interbloqueo
- En dispositivos de E/S, en bases de datos.

15

Recurso

- Sólo puede ser utilizado por un proceso en un instante dado
- Su uso conlleva:
  - Solicitar el recurso
  - Usar el recurso
  - Dejar el recurso
- Si no está disponible, el proceso debe esperar
- Los procesos se interbloquean si cada uno de ellos espera que se genere algún evento que sólo otro proceso del conjunto puede generar.
  - El evento es normalmente la liberación de un recurso que está siendo utilizado por otro proceso.

16

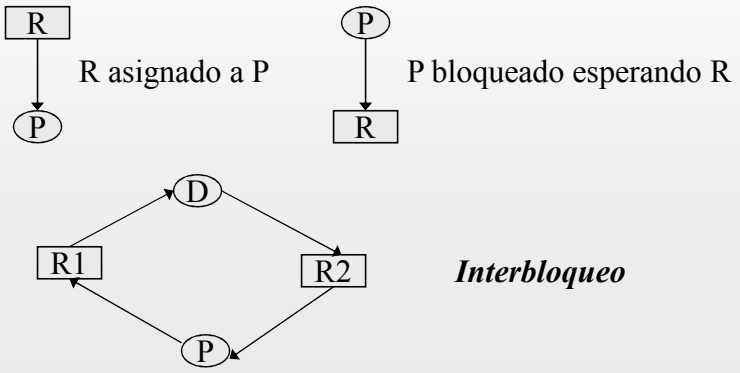
Condiciones para el interbloqueo

- Enunciadas por Coffman
  - ***Exclusión mutua*:** un recurso está asignado a un proceso o está disponible
  - ***Retención y espera*:** Se pueden solicitar más recursos sin abandonar los ya obtenidos
  - ***No expulsión*:** Los procesos tienen que dejar los recursos explícitamente
  - ***Círculo de espera*:** Se produce un círculo de 2 o más procesos

17

Grafos para modelar interbloqueos

- Los grafos sirven para averiguar si una determinada secuencia produce interbloqueo



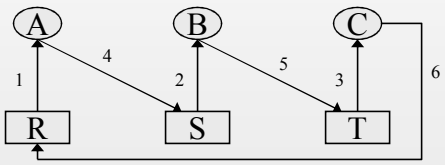
18

Ejemplo

- 3 procesos

A	B	C
Pedir R	Pedir S	Pedir T
Pedir S	Pedir T	Pedir R
Dejar R	Dejar S	Dejar T
Dejar S	Dejar T	Dejar R

- Planificación por turno rotatorio ===== >



1. A pide R
2. B pide S
3. C pide T
4. A pide S
5. B pide T
6. C pide R

- El S.O puede suspender un proceso = >

1. A pide R
2. C pide T
3. A pide S
4. C pide R
5. A deja R
6. A deja S

19

Formas de tratar el interbloqueo

- El algoritmo del avestruz
  - Esconder la cabeza y desentenderse
  - Unix
- Detección y eliminación de interbloqueos
  - Controlar peticiones y liberaciones de recursos
  - Actualizar el grafo de recursos y comprobar que no haya bucles
  - Comprobar periódicamente si hay procesos que han estado bloqueados durante mucho tiempo
- Predicción de interbloqueos
- Prevención de interbloqueos

20

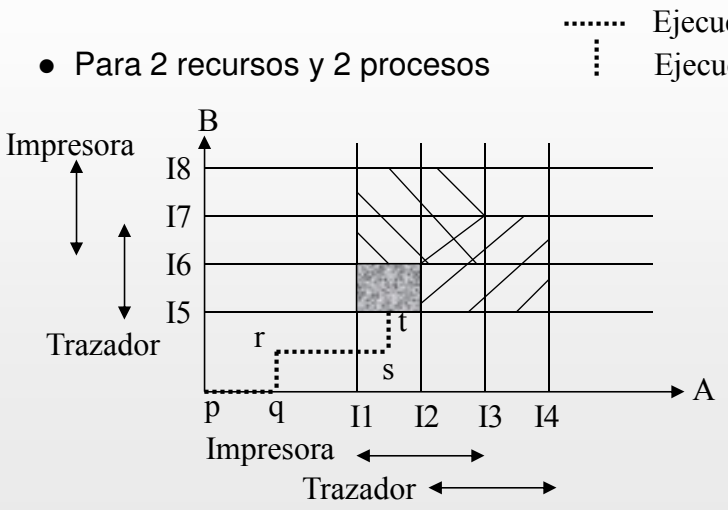
Predicción de interbloqueos

- Evitar los interbloqueos cuando se asignen los recursos
  - Trayectorias de recursos
  - Algoritmo del banquero para un recurso
  - Algoritmo del banquero para múltiples recursos

21

Trayectorias de recursos

- Para 2 recursos y 2 procesos



22

Algoritmo del banquero para un recurso

- Estado del sistema con respecto a la asignación de recursos: lista de procesos, recursos usados y máximo

	usado	max
p1	0	6
p2	0	5
p3	0	4
p4	0	7

Disponible=10

	usado	max
p1	1	6
p2	1	5
p3	2	4
p4	4	7

Disponible=2

	usado	max
p1	1	6
p2	2	5
p3	2	4
p4	4	7

Disponible=1

- No todos los procesos van a utilizar el máximo
- Estado seguro: si hay una secuencia que lleve a todos los procesos a obtener sus máximos

23

Algoritmo del banquero para múltiples recursos

- Ra: recursos asignados en ese momento
- Rn: recursos que se necesitan todavía
- E: recursos existentes
- P: recursos asignados a algún proceso
- D: recursos disponibles

Ra

A	3	0	1	1
B	0	1	0	0
C	1	1	1	0
D	1	1	0	1
E	0	0	0	0
	R1	R2	R3	R4

Rn

A	1	1	0	0
B	0	1	1	2
C	3	1	0	0
D	0	0	1	0
E	2	1	1	0
	R1	R2	R3	R4

E=(6 3 4 2)
P=(5 3 2 2)
D=(1 0 2 0)

P+D=E

24

Algoritmo de comprobación de estado seguro

- Buscar fila F cuyas necesidades de recursos sean menores que D. Si no hay=> estado no seguro, el sistema se puede interbloquear, se necesitan más de los disponibles
- Suponer que el proceso de la fila F pide todos los recursos y termina. Marcar el proceso como terminado y añadir sus recursos a D
- Repetir los dos pasos anteriores hasta terminar todos los procesos. Si se puede terminar => estado seguro

25

Ejemplo

Ra

A	3	0	1	1
B	0	1	0	0
C	1	1	1	0
D	1	1	0	1
E	0	0	0	0

R1 R2 R3 R4

Rn

A	1	1	0	0
B	0	1	1	2
C	3	1	0	0
D	0	0	1	0
E	2	1	1	0

R1 R2 R3 R4

E=(6 3 4 2)
P=(5 3 2 2)
D=(1 0 2 0)

P+D=E

- Estado seguro
  - D D=(2 1 2 1)
  - A D=(5 1 3 2) o E
  - E D=(5 1 3 2) o C
  - C D=(6 2 4 2)
  - B D=(6 3 4 2)

26

Ejemplo

Ra

	R1	R2	R3	R4
A	3	0	1	1
B	0	1	***1***	0
C	1	1	1	0
D	1	1	0	1
E	0	0	0	0

Rn

	R1	R2	R3	R4
A	1	1	0	0
B	0	1	***0***	2
C	3	1	0	0
D	0	0	1	0
E	2	1	1	0

E=(6 3 4 2)
P=(5 3 ***3*** 2)
D=(1 0 ***1*** 0)

P+D=E

- B pide un R3 => ¿estado seguro?

Ejemplo

Ra

	R1	R2	R3	R4
A	3	0	1	1
B	0	1	***1***	0
C	1	1	1	0
D	1	1	0	1
E	0	0	***1***	0

Rn

	R1	R2	R3	R4
A	1	1	0	0
B	0	1	***0***	2
C	3	1	0	0
D	0	0	1	0
E	2	1	***0***	0

E=(6 3 4 2)
P=(5 3 ***4*** 2)
D=(1 0 ***0*** 0)

P+D=E

- E pide un R3 => ¿estado seguro?

Problemas

- Poca aplicación práctica:
  - Los procesos raramente saben la cantidad máxima de recursos que van a utilizar
  - El nº de procesos no es fijo
  - Pueden desaparecer recursos con los que se contaba

29

Prevención de interbloqueos

- Imponer restricciones a los procesos de forma que un interbloqueo no sea posible
- No habrá interbloqueo si no se cumple alguna de las 4 condiciones necesarias para el interbloqueo:
  - Exclusión mutua
    - No se puede eliminar, se puede llegar a condiciones de carrera
  - Retención y espera
    - Solicitar todos los recursos al principio
      - No permite utilización óptima
      - No se sabe a priori
    - Antes de solicitar un recurso, los procesos están obligados a dejar temporalmente los que tienen retenidos
  - No expulsión
    - No aconsejable
  - Círculos de espera

30

Círculos de espera

- Retener un recurso en cada momento
  - No es adecuado
- Dar a los recursos una numeración general.
  - Los recursos se solicitan en orden de numeración

    A - 1
    B - 2
    C - 3
    D - 4
    E - 5

    B y luego D
    pero no D y luego B
  - El grafo no puede tener bucles cerrados
  - Puede ser difícil dar a los recursos una numeración adecuada

31