

I- Respuesta Breve.

1. Su principal función es la protección de recursos.
2. La principal desventaja es el cambio de contexto ya que este suele más costoso.

$$3. \mu = \sum_{i=1}^n \frac{C_i}{P_i} \leq 1 \Rightarrow \lim_{n \rightarrow \infty} n(2^{\frac{1}{n}} - 1) > \ln(2)$$

$$4(2^{\frac{1}{4}} - 1) > 0.75682 \quad \checkmark \quad U(n)$$

$$n=4 \quad \frac{C_1}{P_1} + \frac{C_2}{P_2} + \frac{C_3}{P_3} + \frac{C_4}{P_4} \leq 1$$

Pero $> U(n)$

$$U(n) \leq \sum_{i=1}^4 \frac{C_i}{P_i} \leq 1$$

$$\Rightarrow \frac{2}{4} + \frac{2}{16} + \frac{2}{16} + \frac{2}{32} = \underline{0.8125}$$

$$\therefore \underline{U(n) \leq 0.8125 \leq 1}$$

$$C_i = 2$$

$$P_1 = 4, P_2 = P_3 = 16$$

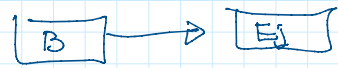
$$P_4 = 32$$

II Falso y Verdadero

1. F : Es un software el cual se divide en varias máquinas virtuales pero en una sola máquina física, "software no ejecuta software"
2. F : No solo se calendariza, requiere una sincronización extra.
3. F : Ya que se requieren más de 1 procesador para un paralelismo, con 1 solo se intercalan los procesos.
4. F : Los procesos son más livianos, no pueden compartir memoria por si solo, los hilos si comparten.
5. F : Este algoritmo no comunica procesos, solo los calendariza.

III - Análisis

- 1- Si un proceso está bloqueado está en espera de un evento de acción entonces si este una vez concluya la espera y tenga lo necesario seguiría el siguiente diagrama



Es desventaja ya que en ejecución puede ya haber 1 proceso y haber conflicto, lo correcto es estar en cola para la ejecución "calendarizar" antes

- 2- Memoria: Registro de frontera, ya que este protege mediante a la autodirección de procesos a espacios libres.

CPU : Se protege con un timer el cual siempre está preguntado al CPU su actividad, además el bit kernel. ←

I/O : Llamados al sistema.

IV Desarrollo.

IV Desarrollo.

1- 00X11 ← Valor mínimo.

Max - Asig. = Requerido

Vmin - requerido = Estado libre

Valor Disponible sería Vmáx del proceso anterior liberado

| | Requerido |
|---|-----------|
| A | 01002 |
| B | 02100 |
| C | 10300 |
| D | 00111 |

si tenemos $X=1$

D = 00000

tendríamos

Disponible (11221) ✓

No podría continuar porque no hay recursos disponibles para los otros procesos

Con $X=2$

D = 00000

C = 00000

Disponible. 11321 ✓

liberado

sería (00211 - 00111) + Vmáx del D+C } $X = \{ \}$

Si $X > 2$

A no se puede ejecutar ← ya que no tiene valores disponibles para ejecutar a pesar que B, C, D se liberaron

$$2- CPU = 1 - p^n$$

p: tiempo de ejecución

$$CPU_{osido} = p^n \Rightarrow P(I/O) \quad n=5$$

$$A: (0,8)^5 = 0,32768$$

$$B: (0,85)^5 = 0,44370$$

$$C: (0,9)^5 = 0,59049$$

$$D: (0,4)^5 = 0,01024$$

$$E: (0,8)^5 = 0,32768$$

$$CPU_{osido} \text{ tot} = 3,65 \times 10^{-9}$$

∴ La probabilidad que D no espec

$$\Rightarrow \text{es } 0,0000365$$

3- Si tenemos que $x < 3$

$$x + (x+3) + (x+3+5) + (x+3+5+6) + (x+3+5+6+9)$$

Con un $3 < x \leq 5$

$$3 + (3+x) + (3+x+5) + (3+x+5+6) + (3+x+5+6+9)$$

Con $x=6$

$$3 + (3+5) + (3+5+x) + (3+5+x) + (3+5+x+9)$$

Con $6 < x < 9$

$$3 + (3+5) + (3+5+6) + (3+5+6+x) + (3+5+6+x+9)$$

Si $x > 9$

$$3 + (3+5) + (3+5+6) + (3+5+6+9) + (3+5+6+9+x)$$

para minimizar x debe ser el valor mas pequeño de x

$$\therefore x < 3$$

ya que + aumenta con $x \geq 3$

En todos los casos se suma siempre x

4-