75.08 Sistemas Operativos — 4 de Octubre de 2022
Primer Parcial
Nombre y Apellido: [redacted]
Padron: [redacted]

1. Kernel y Procesos (10 ptos.)

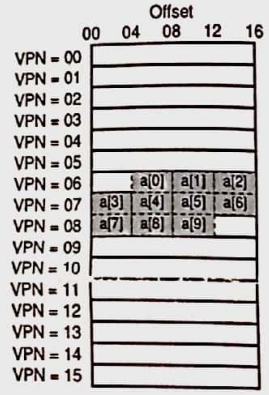
   a. Describa que es un proceso: qué abstrae, cómo lo hace, cuál es su estructura. Además explique el mecanismo por el cual el proceso cree tener la memoria completa de la máquina cuando en realidad solo tiene lo necesario para su funcionamiento.

   b. Cuál / cuáles mecanismos utiliza el kernel para garantizar el aislamiento entre procesos. Estos mecanismos están relacionados con el hardware, porque deben existir y donde se ve su funcionamiento.

INS

2. Memoria (10 ptos.)

   a. Dado el siguiente esquema explique cómo se realizan las traducciones recorriendo el arreglo en un modelo de memoria virtual con tlb y paginación de dos niveles. En el mismo esquema decir cuantos miss, hit , accesos a memoria y traducciones hay.



Responda:

- ☐ En las traducciones hay 3 hits y 7 miss en la TLB.
- ☐ Hay 10 accesos a memoria.
- ☐ En las traducciones hay 7 hits y 3 miss en la TLB.
- ☐ Hay 3 traducciones completas de VA a PA.
- ☐ Hay 3 accesos a memoria en total.
- ☐ Hay 10 traducciones completas de VA a PA.

B. Suponga que virtual address con las siguientes caracteristicas:

4 bit para el segment number
12 bits para el page number
16 bits para el offset

Segment table	Page Table A	Page Table B
0 Page B	0 CAFE	0 F000
1 Page A	1 DEAD	1 D8BF
X invalid	2 BEEF	2 3333
	3 BA11	x INVALID

Traducir las siguientes direcciones virtuales a físicas: 00000000, 20022002, 10022002, 00015555 .

3. Concurrencia y Scheduling (10 ptos.)

   a. Explique con un ejemplo MLFQ.

   b. ¿Cuáles de los siguientes mecanismos son compartidos entre threads de un mismo programa?

      - ☐ Stack Segment
      - ☒ File descriptors
      - ☐ Registros de CPU
      - ☒ Heap
      - ☐ Code segment
      - ☐ METADATA del Thread
      - ☐ Data Segment
      - ☐ Signals

[redacted]

Hoja: 1/2
Fecha: 04/11/2022

1a) Un proceso es una entidad abstracta creada por el Kernel en donde se alojan los recursos del sistema. A un proceso también se lo puede ver como una abstracción de un programa en ejecución.

Un proceso contiene:

- ~~STACK~~ STACK
- Heap: espacio de memoria a lo cable
- Data: variables globales
- Text: código

También contiene File descriptors, datos del Kernel, registros y demás.

Lo que se utiliza para crear la ilusión de que un proceso posee toda la memoria del sistema ~~se llama~~ es virtualización de memoria. Con esto se logra que cada proceso crea que posee toda la memoria del sistema. Esto es a través de un espacio de direcciones que tiene que luego el hardware (puntualmente la MMU) realiza las traducciones de memoria virtual a física.

Entonces, cada proceso al utilizar la memoria del sistema, lo que en realidad hace es usar direcciones virtuales que luego, mediante el espacio de direcciones y la MMU, se mapean a la memoria física real.

Notas:

2B)

1) 0 000 0000
   ↑ segment — page number — offset

   Dir Física = F0000 0000

2) 2 0002 2002

   Dir Física = INVALID ✓

3) 1 0002 2002

   Dir Física = BEEF 2002

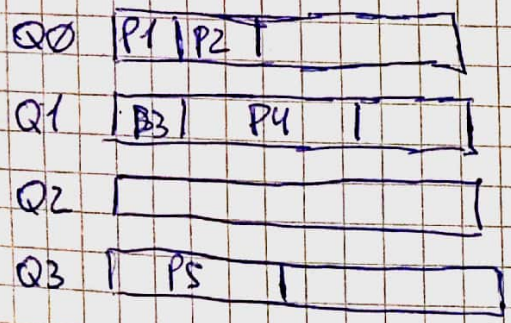
4) 0 0001 5555

   Dir Física = D8BF 5555

~~3)~~

3a)

Llamaremos a cada proceso como P y cada cola la vamos a representar un nivel de prioridad, los denotaremos con Q.

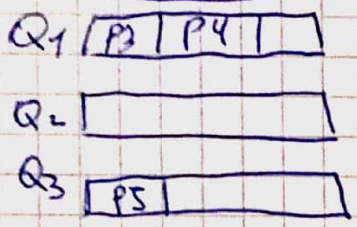


Al momento que se debe ejecutar un nuevo proceso, a este se la va a encolar en alguna de las colas con la prioridad que le sea asignada por el SO. Esta prioridad se puede basar en diversas cosas, como por ejemplo, un tiempo en el que puede llegar a tardar en ejecutarse el proceso.

Vemos que los procesos con mayor prioridad serán

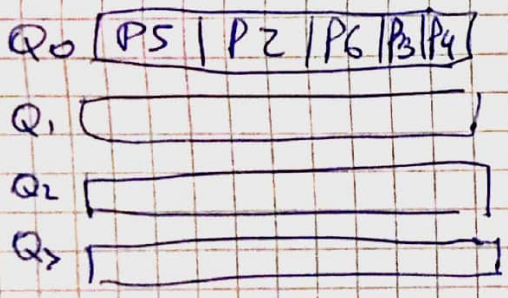
los que se ejecutarán primero.

Supongamos que el proceso P1 termina de ejecutarse → llega un nuevo proceso con mayor prioridad



Se puede observar que a medida que lleguen nuevos procesos a ser ejecutados, el proceso P5 irá a quedarse esperando por mucho tiempo.

Para contrarestar esto, dado un tiempo determinado, el scheduler lo que hará es boostear todos los procesos a primera prioridad.



Luego se volverán a calcular las prioridades. Esto se hace para mantener uno de los principios que deben seguir los schedulers, siendo este el de fairness.