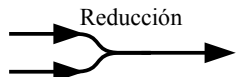
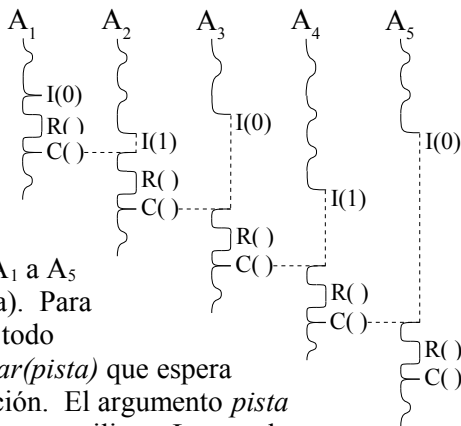


Pregunta 1



Una calle de 2 pistas con tránsito en un solo sentido se reduce a una sola pista. Los automovilistas son representados por threads de nSystem (A_1 a A_5 en el diagrama del ejemplo de la derecha). Para evitar colisiones al llegar a la reducción todo automovilista invoca la función $nIngresar(pista)$ que espera hasta que sea seguro ingresar a la reducción. El argumento $pista$ es 0 o 1 e indica por cual pista viene el automovilista. Luego el automovilista invoca la función dada $reduccion$ para llevar el vehículo a la sección de una sola pista, lo cual toma tiempo. Si 2 automovilistas invocaran $reduccion$ al mismo tiempo ocurriría una colisión. Por último el automovilista invoca la función $nContinuar()$ que notifica la llegada a la zona de una sola pista. A partir de ese instante otro vehículo podrá ingresar a la reducción. En el diagrama estas funciones se abrevian como I, R y C.



Los automovilistas que vienen por la misma pista deber ser atendidos por orden de llegada. Al considerar ambas pistas el orden de atención deber ser alternado, es decir pasa un automovilista de la pista 0, luego uno de la pista 1, luego 0, 1, 0, etc. La alternancia se suspende cuando una pista está desocupada. En el ejemplo cuando A_1 que venía por la pista 0 llama a $nContinuar$, el siguiente vehículo en pasar es A_2 que viene por la pista 1, a pesar de que A_3 que viene por la pista 0 llegó antes que A_2 . Si la pista 1 hubiese estado libre, ahí A_3 hubiese pasado después de A_1 .

Programe las funciones $nIngresar$ y $nContinuar$ usando los procedimientos de bajo nivel de nSystem (START_CRITICAL, Resume, PutTask, etc.). Recuerde que estas funciones deben garantizar la exclusión mutua de las llamadas a $reduccion$ para evitar colisiones. Ud. no puede usar otros mecanismos de sincronización ya disponibles en nSystem como semáforos, monitores, mensajes, etc. Ud. puede agregar campos al descriptor de tarea y usar variables globales pero señale cómo se inicializan.

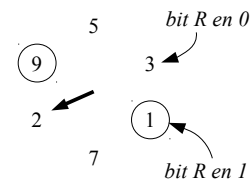
Pregunta 2

Resuelva el mismo problema de la pregunta 1 considerando ahora una máquina octa-core en la que no existe un núcleo de sistema operativo y por lo tanto no hay un scheduler de procesos. Los nombres de las funciones son $ingresar$ y $continuar$. Para la sincronización Ud. dispone de spin-locks y la función $coreId()$. También dispone de las $fifoqueues$. Para recordar por cuál

pista venía un automovilista use un arreglo de tamaño 8. La única forma válida de hacer esperar a un automovilista es utilizando un spin-lock. Otras formas de busy-waiting no están permitidas.

Pregunta 3

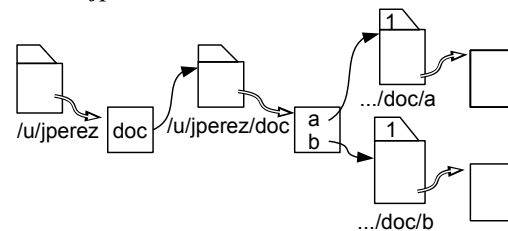
I. Considere un sistema Unix que implementa la estrategia del reloj. El sistema posee 6 páginas reales disponibles y corre un solo proceso. La figura indica el estado inicial de la memoria, mostrando las páginas residentes en memoria, la posición del cursor y el valor del bit R.



Dibuje los estados por los que pasa la memoria para la siguiente traza de accesos a páginas virtuales: 9, 2, 6, 1, 2, 8.

II. ¿Cuál es el tamaño máximo de un archivo que usa un solo bloque de indirección simple en una partición con bloques de 1 KB? ¿Y si la partición posee bloques de 4 KB?

III. La figura muestra a la izquierda varios archivos y directorios de la partición /u. El número que aparece en los inodos de los archivos a y b es el contador de links duros. A la derecha se muestran los comandos que ejecuta el usuario $jperez$:



```
$ cd /u/jperez
$ ln doc/b x
$ mkdir tmp
$ cp x tmp/c
$ ln -s doc/a y
```

Rehaga la figura de la izquierda de acuerdo a los cambios realizados.

IV. Todos los procesadores tienen una instrucción que desactiva las interrupciones. Explique si sería posible que un programador use esa instrucción para suprimir el mecanismo de $time-slicing$ y así impedir que el sistema operativo le quite la CPU a su proceso.

V. Considere 3 tipos de núcleos: un núcleo clásico de Unix para una máquina mono-core, un núcleo moderno para una máquina mono-core y un núcleo moderno para una máquina multi-core. ¿Cómo se garantiza la exclusión mutua al acceder a la cola de procesos “ready” para cada uno de ellos y por qué?