

### Pregunta 1

a.- (3 puntos) Un pub posee un único baño que debe ser compartido por damas y varones. El baño es amplio y admite un número ilimitado de personas. El problema consiste en evitar que las damas se encuentren con los varones dentro del baño. Las siguientes funciones coordinan los ingresos y salidas del baño:

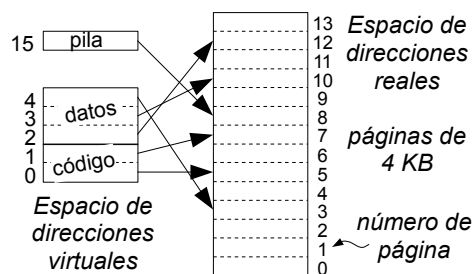
- `void nEntraDama():` Una dama solicita ingresar al baño.
- `void nSaleDama():` Una dama notifica que sale del baño.
- `void nEntraVaron():` Un varón solicita ingresar al baño.
- `void nSaleVaron():` Un varón notifica que sale del baño.

Se requiere que cuando un varón invoca `nEntraVaron`, debe esperar si y solo si hay damas dentro del baño o hay damas esperando ingresar. En caso contrario ingresa de inmediato. De igual forma, cuando una dama invoca `nEntraDama`, espera si y solo si hay varones dentro del baño o hay varones esperando ingresar. Si no, ingresa de inmediato. Cuando sale el último varón del baño, todas las damas en espera deben ingresar y cuando sale la última dama del baño, todos los varones en espera deben ingresar. Con estos requerimientos se logra que múltiples personas usen simultáneamente el baño siempre y cuando sean todas del mismo sexo. Además se evita la hambruna.

Programar las funciones de más arriba usando los procedimientos de bajo nivel de `nSystem` (`START_CRITICAL`, `Resume`, `PutTask`, etc.). Ud. no puede usar otros mecanismos de sincronización ya disponibles en `nSystem` como semáforos, monitores, mensajes, etc. Declare las variables globales que necesite e indique cómo se inicializan. Use una cola (`Queue`) para hacer esperar a las damas y otra cola para los varones en espera.

Observe que cuando el baño está siendo ocupado por damas y hay varones en espera, si llega Ana y luego Pedro, Pedro esperará solo hasta que salga la última dama. Ana continuará esperando hasta que salga el último de los varones. Esto significa que los ingresos no necesariamente ocurren por orden de llegada.

b.- (1,5 puntos) La figura de la izquierda muestra la asignación de páginas virtuales de un proceso a páginas reales de un sistema Unix que implementa `fork` usando la técnica `copy on write`.



Suponga que el proceso llama a `fork`. Indique el contenido de las tablas de páginas del proceso padre y del proceso hijo justo después que el hijo escribe en la página virtual 15 y el proceso padre escribe en la página virtual 3. Incluya en las tablas: número de página virtual, número de página real y atributos de validez y escritura (V y

W).

c.- (1,5 puntos) Considere el siguiente diagrama de lecturas y escrituras (r, w) en memoria de un proceso Unix en un sistema que usa la estrategia del `working set`. Las filas corresponden a las páginas (0, 1, 2, ...) y las columnas a los intervalos de cálculo del `working set` (A, B, C, ...).

página	6	w r	r	r		r		
	5	w r r	r	r			r	
	4	w r		r		r w		r
	3	w r r			r r			r r r
	2	w r			r w			r r
	1	w	r	r	w r	r w	w	r
	0	w r	r	r		w r	r	r
		A	B	C	D	E	F	G

Tiempo

(i) Indique qué accesos a las páginas 4 y 5 podrían producir un `page-fault` (use como notación 3, G, 1<sup>er</sup> acceso). (ii) Suponiendo que se usa el bit D (`dirty`) para optimizar los reemplazos y que producto de un `page-fault` de otro proceso las páginas 2 y 3 fueron escogidas para ser reemplazadas en los intervalos C y F, indique para cada uno de esos 4 reemplazos si fue necesario grabar la página en el área de paginamiento para hacer el reemplazo.

### Pregunta 2

I. (3 puntos) Resuelva el mismo problema del pub de la pregunta 1 considerando ahora una máquina multi-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 que Ud. debe programar son: `entraDama`, `saleDama`, `entraVaron` y `saleVaron`. Para la sincronización Ud. dispone de spin-locks y la función `coreld()`. Para hacer esperar a las damas use un arreglo de 8 punteros a spin-locks y otro arreglo similar para los varones en espera.

**Restricción:** La única forma válida de hacer esperar a una persona es utilizando un spin-lock. Otras formas de `busy-waiting` no están permitidas. No hay `fifoqueues`.

II. (1 punto) Se tiene un archivo que no requiere bloques de indirección doble en una partición Unix con bloques de 2 KB. Se agrega un byte a este archivo y se crea el bloque de indirección doble. Haga un diagrama mostrando inodo, bloques de datos y de indirección. ¿De qué tamaño es el archivo?

III. (1 punto) 5 procesos se encuentra en estado de espera porque hicieron peticiones para leer bloques del disco en el siguiente orden: 900, 100, 800, 600, 700. El último bloque leído fue el 300 y el penúltimo el 200. Indique en qué orden se harán las lecturas de estos 5 procesos cuando la estrategia de scheduling de disco es: (i) `first come first served`, (ii) `shortest seek first`, (iii) método del ascensor (o `look`).

IV. (1 punto) Meltdown es una vulnerabilidad descubierta en los procesadores intel que permite que un proceso pueda leer en modo usuario el contenido del núcleo del sistema operativo. Explique cómo se resuelve esta vulnerabilidad.