

## Enunciado de la prueba práctica

El alumno debe realizar un programa denominado <client.c> que se ejecutará como <./client> en el directorio de trabajo donde reside el programa <./monitor3>. El programa monitor propone la realización de 7 prácticas que pueden ejecutarse de forma independiente pero que están enunciadas pensando en que se hagan de forma progresiva.

### Ejercicio 1: Engancharse, leer datos de una memoria compartida y desengancharse

El programa monitor creará un segmento de memoria compartida de 1024 bytes utilizando como clave el DNI del alumno en hexadecimal (Ej:0x1234L) y escribirá en él valores de diferentes tipos de variable que el cliente deberá localizar y leer. En la posición 0 de la memoria compartida escribirá un **entero** cuyo valor será el secreto <1> del ejercicio. Avanzando en la memoria a la posición del segundo **entero**, encontrará un valor que será el Offset o salto en bytes a la posición del siguiente tipo de dato a leer en memoria. En esa nueva posición habrá una **cadena de caracteres** con formato "<999>" donde se indicará el secreto <2>. El proceso monitor no conocerá de momento si el proceso cliente ha identificado adecuadamente los secretos pero en el ejercicio 2 lo podrá descubrir. Para ello el proceso cliente debe guardar los secretos <1> y <2>.

El programa cliente esperará 3 segundos para desengancharse del segmento de memoria. Cuando lo haga, el monitor revelará en pantalla la clave <3> y liberará y eliminará el segmento de memoria compartida.

### Ejercicio 2: Crear una memoria compartida y escribir en ella

El proceso cliente debe crear un segmento de memoria compartida de tamaño 1024 bytes con la clave de su DNI.

Se espera que el cliente escriba la clave <2> en la posición 0 de esa memoria con formato "<999>". Si no conoce ese secreto puede escribir en la posición 0 cualquier número con ese formato (Ej: "<111>") o la cadena de caracteres "HOLA". Si el alumno demuestra que puede escribir correctamente en esa posición, aunque no sepa la clave <2>, el monitor podrá revelar la clave <4>.

Se espera también que el cliente escriba en la memoria en la posición indicada por el número dsp= ("clave <2>"+16) la cadena de texto con formato "<888>" que corresponde a la clave <1> del ejercicio anterior. Si el alumno demuestra que puede escribir correctamente en esa posición, aunque no sepa la clave <1>, el monitor podrá revelar la clave <5>.

Para que el monitor sepa cuándo está la memoria disponible para leer el resultado, es necesario que el programa cliente se desenganche de la memoria en un tiempo no superior a 3 segundos desde que el cliente obtuvo el acceso a esa memoria.

### Ejercicio 3: Crear e inicializar un semáforo

El proceso monitor esperará a que el proceso cliente cree un semáforo con la clave de su DNI y lo inicialice con el valor correspondiente a las tres últimas cifras de su DNI. Cuando el monitor vea que ese semáforo existe y que su valor es correcto revelará en pantalla la clave <6>.

### Ejercicio 4: Crear e inicializar un array de semáforos

El monitor esperará unos segundos hasta que el cliente cree un array de 2 semáforos y los inicialice a 1 el primero y 2 el segundo, momento en el que desvelará la clave <7>.

### Ejercicio 5: Operar en un semáforo. Acceso sincronizado a datos.

El proceso monitor eliminará los recursos anteriores y creará un semáforo (de tamaño 1) y un segmento de memoria compartida (1024 bytes), ambos, con la clave correspondiente al DNI del alumno. El semáforo lo inicializará en rojo (a 0) para controlar el acceso concurrente a la memoria compartida. El monitor irá escribiendo valores **enteros** aleatorios en la posición 0 de la memoria, de modo que si el cliente accede a dicha posición sin sincronizarse encontrará datos incorrectos. Cuando el monitor considere oportuno, escribirá la clave <8> como un entero en la posición 0 y avisará de que existe un dato correcto poniendo el semáforo en verde (a 1). Inmediatamente después, el monitor intentará bloquear el semáforo y eliminar el dato de la memoria. Para obtener la clave <8> de forma correcta, el cliente deberá sincronizarse con el semáforo y evitar que el monitor elimine el dato hasta que el cliente lo lea y modifique. El cliente debe reescribir la clave en la misma posición pero con el signo cambiado. Además, para asegurarse de que se controla el acceso concurrente, el cliente debe esperar 3 segundos dentro de la sección crítica antes de poner en verde el semáforo para desbloquear al monitor. Si se hace esto de forma correcta se desvelará en pantalla la clave <9> con independencia de que se haya desvelado o no la clave <8>.

### Ejercicio 6: Operar en un array de semáforos. Acceso sincronizado a datos.

En este ejercicio se operará de forma similar al ejercicio 5, con la excepción que se utilizará un array de semáforos de tamaño 2 para sincronizar el acceso al dato correcto. El monitor creará de nuevo los segmentos de memoria y el semáforo múltiple con las claves correspondientes al DNI. La operativa es la misma salvo que sólo se dispondrá del dato correcto de la clave <10> en formato entero en la posición 0 de la memoria cuando los dos valores de estado del array estén simultáneamente en verde (a 1). Del mismo modo, tras reescribir el dato cambiado de signo y esperar los 3 segundos para abandonar la sección crítica se desvelará la clave <10>.

### Ejercicio 7: Eliminar los recursos de comunicaciones IPCs creados

El monitor eliminará los recursos anteriores y procederá a crear un segmento de memoria compartida con el DNI del alumno como clave y tamaño 1024 bytes. Posteriormente esperará a que el cliente cree un semáforo de tamaño 1 con la clave del DNI del alumno. El cliente estará pendiente de que el monitor ponga en verde (a 1) ese semáforo para proceder a la eliminación de los recursos creados (memoria y semáforo) en un tiempo máximo de 1 segundo. Si es así, se descubrirá el secreto <12>.

---

## LABORATORIO DE ARQUITECTURA DE SISTEMAS DE LA INFORMACIÓN

Nombre:..... Grupo: .....

DNI:.....

Fecha:.....

Ejercicio 1: Leer shm	Secreto 1:	Secreto 2:	Secreto 3:
Ejercicio 2: Escribir shm	Secreto 4:	Secreto 5:	
Ejercicio 3: Crear sem	Secreto 6:		
Ejercicio 4: Crear array sem	Secreto 7:		
Ejercicio 5: Operar sem	Secreto 8:	Secreto 9:	
Ejercicio 6: Operar array sem	Secreto 10:	Secreto 11:	
Ejercicio 7: Eliminar recursos IPC	Secreto 12:	Firma:	