

Sistemas Operativos
Grado en Ingeniería del Software (Grupo 2ºA)
14 de junio de 2013, Curso 2012-2013
Facultad de Informática, UCM

Apellidos, Nombre: _____

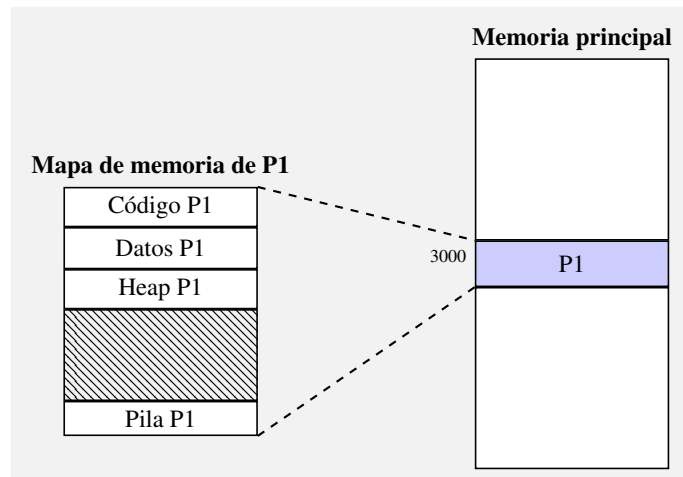
DNI: _____

Cuestiones

C1.- (1 p.) En un sistema con paginación pura, un proceso P utiliza sólo los 8MB de direcciones virtuales más bajas y los 12MB de sus direcciones virtuales más altas de su mapa de memoria.

- a) Suponiendo que se emplean tablas de páginas de un nivel con entradas de 4 bytes, direcciones virtuales de 32 bits y páginas de 1KB, calcular el espacio necesario (en bytes) para alojar la tabla de páginas de P .
- b) ¿Cuál sería el tamaño en bytes de la tabla de páginas si se empleara un esquema de dos niveles con índices de 11 bits para cada nivel?

C2.- (0,75 p.) La memoria principal (MP) de un sistema gestionado mediante asignación contigua almacena el mapa de memoria de un proceso P1 cuyo contenido se muestra en la figura. Dibuje el contenido de la MP después de que P1 cree un proceso P2 con `fork()` y de que P2 cree un hilo. Se ha de indicar también el contenido del mapa de memoria de P2 y la relación existente entre las regiones de P1 y P2. (Asumir que P2 no ha invocado `exec()` y que su mapa de memoria se almacenará a partir de la posición 7000).



C3.- (0,75 p.) Un disco duro tiene las siguientes peticiones pendientes: 94, 179, 33, 118, 10, 120, 61, 53. La cabeza de lectura/escritura está inicialmente sobre el cilindro 49 y su movimiento es hacia cilindros crecientes. Indique el orden de atención de dichas solicitudes ante los algoritmos FCFS, SCAN y C-SCAN.

	1	2	3	4	5	6	7	8
FCFS								
SCAN								
C-SCAN								

C4.- (1 p.) En la práctica 3, el hilo principal del programa hace uso de la función `system()` de la biblioteca estandar de “C”.

```
int system(const char *command);
```

Esta función crea un nuevo proceso shell que invoca el comando que se pasa como parámetro a la función, y devuelve el código de salida (*status*) del shell. El proceso de usuario que invoca `system()` se queda bloqueado hasta que el comando finalice.

Implemente la función `system()` empleando llamadas al sistema. (**Nota:** Para ejecutar un comando se puede invocar al shell de la siguiente forma: `/bin/bash -c '<comando>'`. Ejemplo: `/bin/bash -c 'ls -l'`).

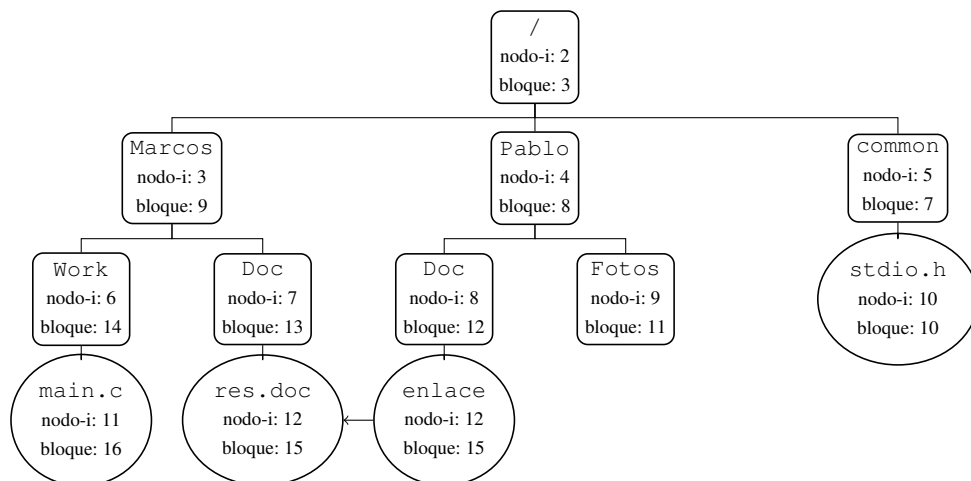
C5.- (1 p.) Implemente las operaciones `lock()` y `unlock()` de un cerrojo de espera activa (*spinlock*) empleando la instrucción máquina XCHG vista en clase. Esta instrucción máquina intercambia de forma atómica el contenido de dos posiciones de memoria (argumentos de la instrucción). Suponer que el tipo de datos (`spinlock_t`) se define de la siguiente forma:

```
typedef struct {
    unsigned int estado;      /* 0 -> abierto, 1 -> cerrado */
    id_hilo propietario;     /* Identificador del hilo propietario del cerrojo */
} spinlock_t;
```

Problemas

P1.- (2 p.) Un sistema de ficheros tipo UNIX utiliza bloques de disco de 2K bytes. Para el direccionamiento de estos bloques se utilizan punteros de 32 bits. Para indicar el tamaño del fichero y el desplazamiento (offset) de la posición en bytes en las operaciones read y write, se utilizan números de 64 bits. Cada nodo-i tiene 10 punteros de direccionamiento directo, 1 puntero indirecto simple y 1 puntero indirecto doble.

- ¿Cuál será el tamaño máximo de un fichero en este sistema suponiendo despreciable el espacio ocupado por el superbloque y la tabla de nodos-i?
- ¿Cuál sería el porcentaje de espacio desaprovechado si el volumen donde está instalado el sistema de ficheros estuviese completamente lleno de ficheros de 7KB cada uno?
- Dada la siguiente estructura de directorios¹, en el que Pablo comparte el fichero `res.doc` de Marcos mediante un enlace rígido de nombre `enlace`, indicar los contenidos de los directorios y de los nodos-i que encontrará el sistema (y el orden en que se los encontrará) al hacer la búsqueda del fichero `enlace` desde el directorio raíz.



¹Los directorios se muestran como rectángulos con bordes redondeados y los ficheros regulares como óvalos y círculos.

- P2.- (1.5 p.) En un sistema monoprocesador, un usuario lanza tres trabajos intensivos en CPU (A, B y C) al mismo tiempo. La estimación de los tiempos de CPU de los procesos es de 4, 2 y 9 unidades de tiempo, respectivamente. Para cada uno de los siguientes algoritmos de planificación determine los tiempos de ejecución y de espera de cada proceso y la productividad. (Suponer que el tiempo de conmutación entre procesos es despreciable).
- a) SJF
 - b) RR con $q=3$
 - c) RR con $q \approx 0$, es decir, empleando la política de compartición ideal del procesador y midiendo los tiempos cuando finaliza cada proceso
- P3.- (2 p.) Proporcionar una solución al problema de los lectores/escritores de forma que sean los escritores los que tengan prioridad de acceso a la sección crítica. Codificar la solución empleando un mutex, dos variables condición y variables compartidas. Cada variable condición se usará para bloquear los hilos de un solo tipo, lectores o escritores.