

Sistemas Operativos
Grado en Ingeniería del Software (Grupo 2ºA)
16 de septiembre 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 10MB 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 4KB, calcular el tamaño de 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 10 bits para cada nivel?

C2.- (1,25 p.) En un sistema tipo UNIX, un proceso ejecuta el siguiente programa:

```
1 #include <stdio.h>
2 #include <pthread.h>
3
4 #define N 1024
5 #define NHILOS 2
6 int A[N], B[N], C[N];
7
8 void* suma_vector(void* arg) {
9     int* begin=(int*) arg;
10    int end=(*begin)==0?N/2:N;
11    int i=(*begin);
12
13    for (;i<end;i++){
14        C[i]=A[i]+B[i];
15    }
16    return NULL;
17 }
18
19 void inicializa_vectores(void){ ... }
20
21 void imprime_resultado(void){ ... }
22
23 int main(void) {
24     pthread_t tid[NHILOS];
25     int i;
26     int start[NHILOS]={0,N/2};
27
28     inicializa_vectores();
29
30     for (i=0;i<NHILOS;i++){
31         pthread_create(&tid[i],NULL,suma_vector,
32             &start[i]);
33     }
34
35     printf("Hilos_creados\n");
36
37     for (i=0;i<NHILOS;i++){
38         pthread_join(tid[i],NULL);
39     }
40
41     imprime_resultado();
42
43     return 0;
44 }
```

- a) ¿Además de la cabecera, de qué secciones estará constituido el fichero ejecutable (ELF) de dicho programa?
- b) Indicar qué regiones están presentes en el mapa de memoria del proceso cuando el hilo principal del programa ejecuta la sentencia de la línea 35. Asumir que en este punto los hilos creados ya han comenzado a ejecutar la función `suma_vector()`.

C3.- (1,25 p.) En un sistema que emplea paginación pura con páginas de 1KB, considerar la siguiente secuencia de direcciones virtuales referenciadas por un programa: 40, 50, 1040, 1070, 73, 3081, 1109, 2093, 2094, 4130. Suponiendo que el SO implementa la política de reemplazamiento LRU y que hay dos marcos de página disponibles para el programa (inicialmente vacíos), rellenar la siguiente tabla indicando el contenido de los marcos de página tras cada referencia e indicar el número de fallos de página que se producen.

Núm. Página										
M0										
M1										

Número de fallos de página: _____

- C4.- (1 p.) Implemente las operaciones `lock()` y `unlock()` de un cerrojo de espera activa (*spinlock*) empleando la instrucción máquina `TEST&SET` vista en clase. Esta instrucción máquina realiza de forma atómica las siguientes operaciones: (1) lectura de una posición de memoria (argumento de la instrucción) cuyo valor se almacena en un registro predefinido y (2) escritura de un valor inmediato (pasado como argumento de la instrucción) a la posición de memoria de la que se ha leído previamente. 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;
```

Nota: Suponer que existe una función `id_hilo_actual()` que devuelve el identificador del hilo invocador.

Problemas

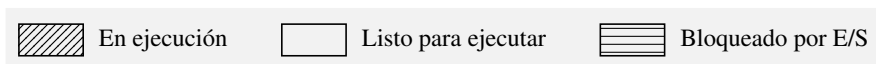
- P1.- (1,5 p.) Un sistema de ficheros tipo UNIX utiliza bloques de disco de 4K bytes. Para el direccionamiento de estos bloques se utilizan punteros de 32 bits. Para indicar el tamaño del fichero y el desplazamiento (offset) de la posición en bytes en las operaciones `read` y `write`, se utilizan números 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*?
- Si se empleasen números de 32 bits para indicar el offset. ¿Cuál será el tamaño máximo de un fichero en este sistema ?

- P2.- (2 p.) Considere un sistema monoprocesador con una política de planificación de procesos round-robin con cuanto 4. Inicialmente, hay 3 procesos en la cola del planificador: P1, P2 y P3 (en este orden). Estos procesos se ejecutan de forma indefinida siguiendo los siguientes patrones de CPU y E/S:

P1 (3-CPU, 5-E/S), P2 (7-CPU, 5-E/S) y P3 (5-CPU, 5-E/S)

- Dibujar un diagrama de tiempos en el que se muestre el estado de los procesos durante las 20 primeras unidades de tiempo. Para representar los estados de los procesos en el diagrama se ha de emplear la siguiente notación:



- Partiendo del diagrama del apartado anterior, calcular el porcentaje de utilización de la CPU y el tiempo de espera de cada proceso durante las 20 primeras unidades de tiempo.
- P3.- (2 p.) Una tribu de salvajes se sirven comida de un caldero con *M* raciones de estofado de misionero. Cuando un salvaje desea comer, se sirve una ración del caldero a menos que esté vacío. Si está vacío deberá avisar al cocinero para que reponga otras *M* raciones de estofado, y entonces se podrá servir su ración. Un número arbitrario de salvajes y el cocinero se comportan del siguiente modo:

```
void Salvaje() {  
    while (true) {  
        obtenerRacion();  
        comer();  
    }  
}
```

```
void Cocinero() {  
    while (true) {  
        prepararRaciones(M);  
    }  
}
```

Implementar las funciones `obtenerRacion()` y `prepararRaciones()` empleando un mutex, dos variables condición y variables compartidas. Cada variable condición se usará para bloquear los hilos de un solo tipo: los salvajes o el cocinero.