

SISTEMAS OPERATIVOS I – Examen de TEORÍA – Conv. de Febrero (10-2-2010)

Apellidos: _____ Nombre: _____

D.N.I.: _____ Titulación: _____ Grupo Teoría: _____

1. [1,25] Suponga un Sistema Operativo con una política de planificación de CPU apropiativa por prioridades. Responda a las siguientes cuestiones:
- ¿Qué transiciones de estados de los procesos provocan que haya que tomar una decisión de planificación de CPU?
 - Describa en pseudocódigo el trozo de código del Sistema Operativo común a las diferentes partes del sistema encargadas de disparar (iniciar) dichas transiciones.

2. [1,25] Describa en pseudocódigo las partes del Sistema Operativo que modificaría para implementar un planificador de CPU que siga una política por turnos (*Round Robin*). Suponga que dispone de las siguientes funciones necesarias para utilizar el **TDA Cola**:

```
typedef enum{BLOQUEADOS,LISTOS,...} Cola_t;
InsertarFinalCola(PCB_t process, Cola_t cola);
SacarCabezaCola(PCB_t process, Cola_t cola);
```

3. [0,75] Explique al menos una ventaja de la implementación de hebras a nivel de usuario frente a la implementación de hebras a nivel de núcleo con respecto a la **eficiencia** obtenida.
4. [1,5] En un recinto ferial existe una atracción que consiste en realizar un paseo por barca que tiene las siguientes especificaciones de funcionamiento:
- En la barca sólo caben 3 pasajeros y sólo sale del embarcadero si está llena.
 - Cuando una persona llega a la atracción y no hay sitio ya en la barca, espera en una cola siempre y cuando ésta no tenga una longitud de 50 personas.
 - Cuando la barca vuelve del trayecto, se bajan los tres pasajeros y se montan otros tres.
 - Los pasajeros, una vez disfrutado del viaje, no vuelven a repetir en esta atracción.

La estructura del procedimiento de las personas que quieren acceder a la atracción puede describirse como:

```
Pasajero () {
    Entrar_en_la_barca;
    pasear ()
    Salir_de_la_barca;
}
```

Implemente el código para la función *Pasajero* y resuelva los problemas de sincronización-comunicación que existen en la entrada y salida de la barca utilizando semáforos y las variables que crea oportunas.

5. [0,5] Supongamos que un proceso en ejecución provoca una falta de página. Este hecho, ¿implica que el proceso cambie de estado? Justifique su respuesta en base al funcionamiento del módulo *Gestor de Memoria* en un sistema que utilice memoria virtual paginada.
6. [1,5] Suponga un sistema que utiliza direcciones de **29 bits**. La gestión de memoria se realiza mediante un mecanismo de paginación a dos niveles con la siguiente distribución de bits: los **13 bits** menos significativos son para desplazamiento (*offset*), los **8 bits** siguientes para indexar las tablas de páginas de segundo nivel, y los restantes **8 bits** para indexar la tabla de páginas de primer nivel. La memoria RAM del sistema es de 10 Mbytes, y las páginas del proceso están cargadas en la dirección 32768 de forma consecutiva. Responda a las siguientes cuestiones:
- Describa mediante un dibujo las tablas de páginas para el mapa de memoria mostrado en la Tabla.
 - Traduzca, utilizando las tablas de páginas del apartado anterior, las direcciones 133250 y 250324.

Región	Dirección Virtual inicio	Dirección virtual fin	Tamaño (KB)
Código	0	12288	12
Datos	12289	28929	16,25
Pila	2^{29}	$2^{29}-8192$	8

7. [0,75] Explique con detalle el fenómeno de *Hiperpaginación (thrashing)*. Describa algún método para evitarla.