

Enunciado

Se pretende adaptar un sistema operativo multiproceso a un pequeño dispositivo portátil. Dicho dispositivo dispone de una batería limitada y se pretende prolongar el tiempo de uso por medio de la hibernación del sistema sobre un soporte no volátil en el instante en el que el sistema se encuentra ocioso.

Las características del sistema operativo original son:

- Sistema UNIX-POSIX.
- Modelo con memoria virtual (hasta 4Gb) con tablas de páginas de 2 niveles.
- No hay preasignación de swap.
- La existencia de procesos de kernel (del estilo kswapd o kupdate), pero no existe ningún proceso nulo (*idle process*).

El dispositivo portátil tiene las siguientes características:

- Memoria física 4MB RAM y 1MB ROM
- Soporte de almacenamiento no volátil (24MB)
- Procesador que implementa interrupciones software
- Hardware y periféricos de entrada/salida

El soporte de almacenamiento no volátil debe usarse tanto para archivos y datos de los programas como para área de swap y salvaguarda del estado del sistema. Hacer un uso eficiente de este espacio se considera clave en el diseño del sistema.

- a) ¿Cuáles son las estructuras asociadas a procesos e interrupciones que sería necesario salvaguardar?
- b) ¿Cuáles son las estructuras asociadas a la gestión de memoria que se tendrán que salvar?
- c) En que variaría la estrategia de salvaguarda (y posterior recuperación) del estado de tratarse en cada uno de los siguientes casos:
 - (i). Área de salvaguarda 4MB y área de swap 10MB.
 - (ii). Área de salvaguarda 2MB y área de swap 12MB.
- d) Si la condición de hibernación del sistema es que pase más de 3 minutos “ocioso”, ¿cómo se podría disparar dicho proceso?. Para responder a esta pregunta considere:
 - ⊙ Que el tiempo consumido por el proceso de salvaguarda del estado es necesariamente largo.
 - ⊙ La prioridad e importancia de otras tareas del sistema.
 - ⊙ El supuesto que se va considerar es el visto en la pregunta anterior como caso (ii).
 - ⊙ La consistencia del sistema es **fundamental**.
- e) Indique con fragmentos de pseudo-código comentado las tareas a realizar en las rutinas y procedimientos apropiados para articular el proceso de hibernación del sistema, tal y cómo lo ha desarrollado en el punto anterior.

CUESTIÓN OPTATIVA

- f) Desde el punto vista general, indique qué complicaciones podría suponer la aplicación de lo dicho anteriormente a un sistema SMP (*symmetric multiprocessor*).

Solución

- a) Las estructuras que se tendrán que salvaguardar en un proceso de hibernación son aquellas correspondientes al estado de ejecución del sistema operativo y sus procesos. No sería necesario salvaguardar información que se pueda ser reconstruida de otro modo.
 - La lista de bloques de control de proceso BCPs y su estado (lista de listos, bloqueados, etc.) **deben ser guardados** puesto que representa el estado del sistema.
 - El vector de interrupciones corresponde a una tabla (en la mayoría de casos estática) de punteros a funciones. Esta parte **no sería necesaria salvaguardar** pues en un proceso normal de arranque del núcleo sería inicializada correctamente.
 - Referente a las interrupciones sí **sería necesario salvaguardar**, a priori, el nivel de interrupción en el momento de la hibernación, así como las interrupciones pendientes
 - Las estructuras relativas a mecanismos entre procesos, tales como pipes, semáforos, regiones de memoria compartida, tablas de ficheros abiertos, también **deberían almacenarse**. Estas estructuras, y otras más, no están directamente relacionadas con la gestión

Se pretende adaptar un sistema operativo multiproceso a un pequeño dispositivo portátil

de procesos y no sería necesario mencionarlas en este apartado.

En resumen, si se guarda el BCP de un proceso, ahí se tiene toda la información necesaria para volver a poner en marcha dicho proceso (registros, tabla de descriptores de archivos, manejadores de señales, etc.). A nivel del planificador se necesitan las estructuras que éste usa para sus tareas, es decir las listas de procesos en cada uno de los estados.

b) La gestión de memoria requiere que se mantengan las estructuras correspondientes a las páginas de memoria virtual:

- **Se guardarán** las tablas de páginas (los dos niveles).
- La tabla de marcos de página que indica la ocupación de la memoria física también **deberían guardarse**.

c) En el caso (i) toda la memoria física (RAM) del sistema puede ser volcada al área de salvaguarda, ésta es sin duda la solución más sencilla. Esta operación no se puede realizar en el caso (ii), aquí se dispone de un área de salvaguarda menor. Para poder realizar el proceso de hibernación hay que reducir la memoria ocupada por debajo de esos 2MB.

En el sistema operativo existen regiones de memorias que no son paginables, tales como las asociadas a determinadas estructuras de control del kernel, esas regiones no podrían liberarse, pero los marcos de páginas de los procesos podrían:

- Eliminadas de memoria páginas de código que se pueden recuperar de archivo.
- Mandar a swap páginas de datos o pila.

Lo más cómodo consistiría en marcarlas como no residentes y mandarlas a swap/eliminarlas como si hubiesen sido expulsadas. De esta forma el sistema mantendría la consistencia al ser reactivado de nuevo y dichas páginas se recuperar como si se tratasen de páginas no residentes cualquiera.

d) Hasta ahora no hemos considerado la condición de “*ociosidad*” necesaria para disparar el proceso de hibernación. Tampoco, resultan imprescindibles las condiciones bajo las cuales se hace, de todas formas **se podría** se consideraría ocioso el sistema si no está ejecutando ninguna tarea de usuario (no procesos de kernel), es decir que todas están bloqueadas.

Ahora hay que ver cómo se podría detectar esta condición. Una opción es que sea un proceso el que determine que este estado en el sistema, pero un proceso no sabe de la ejecución de otros procesos del sistema operativo y de si el sistema está cargado o no. Quien dispone de esa información es el propio kernel.

Una posible alternativa (no la única) consistiría en tener un contador de ticks ociosos, inicializado a los correspondientes a 3 minutos. De esta forma, cada vez que se active la interrupción de reloj si existe un proceso en ejecución se le descuenta una rodaja de tiempo, si no hay ninguno se descuenta un tick al contador de ocioso. En el momento en el que se active la interrupción software para un cambio de contexto, si en la cola de listos hay procesos disponibles entonces se pone el contador de ticks ociosos a su valor inicial. En la rutina de tratamiento de interrupción de reloj, si el contador llega a cero entonces se pone el sistema en estado de hibernable y se dispara la interrupción software para que realice esta tarea.

e)

```
ocioso=TICKS_OCIOSO;

int_reloj()
{
    if(proceso_en_ejecución==no)
    {
        if(--osioso)
        {
            hibernar=si;
            activar_int_SW();
        }
    }
    else if(--rodaja)
    {
        replanificar=si;
        activar_int_SW();
    }
}

int_SW()
{
    if(hibernar==si)
    {
        hibernar=no;
        fijar_nivel_int(NIVEL_MAX); /* Esto nos asegura que no se
                                     interrumpe el proceso */
        /* COMENZAR EL PROCESO DE HIBERNACIÓN */
    }
    else if(replanificar==si)
    {
        replanificar=no;
    }
}
```

Se pretende adaptar un sistema operativo multiproceso a un pequeño dispositivo portátil

```

    proceso_en_ejecucion=si;
    fijar_nivel_int(NIVEL_MAX);
    proc=planificador();
    if(es_de_proceso_de_kernel(proc)==no)
        ocioso=TICKS_OCIOSO;

    /* CAMBIO DE CONTEXTO */
}
}
```

f) De tratarse de un multiprocesador habría que aplicar el mecanismo análogo pero considerando que puede ser que uno de los procesadores sea el que está ocioso y no el otro. Habría que usar semáforos en los casos en los cuales se manipulan los contadores. Para este tipo de procesadores, por lo general la interrupción de reloj sólo llegaría a uno de ellos, en caso de que no sea así habría que defender con semáforos las condiciones de evaluación, exactamente igual que se protegen a la hora de un cambio de contexto o de manipular la lista de BCPs.