

DISEÑO DE SISTEMAS OPERATIVOS. Junio de 2003

Ejercicio 1

Sea un sistema con memoria virtual, que usa un algoritmo de reemplazo de páginas FIFO, en el que las direcciones lógicas (virtuales) que tienen un 0 en el bit de mayor peso son de usuario y las que tienen un 1 son de sistema. Este sistema usa un algoritmo de planificación de procesos expulsivo basado en prioridad. Además, el S.O. es no expulsivo (sin llamadas concurrentes), usando para ello el mecanismo de interrupción software. Se pide responder razonadamente a las siguientes cuestiones sobre la rutina de fallo de página de este sistema:

a) [Puntuación: 2'5/10] Considérese que el sistema operativo no tiene ningún error de programación y que se produce un fallo de página tal que la dirección que lo causa no pertenece a ninguna región del proceso. Dependiendo de qué código estuviera ejecutando el proceso cuando ocurrió el fallo y del tipo de dirección, se plantean los siguientes casos, cada uno de los cuales se deberá analizar para ver si es posible esa situación y, en caso afirmativo, explicar qué tratamiento requiere analizando si influye en el mismo el valor del puntero de pila en el momento del fallo:

- a1)** Proceso estaba ejecutando en modo usuario y la dirección de fallo comienza con un 0.
- a2)** Proceso estaba ejecutando en modo usuario y la dirección de fallo comienza con un 1.
- a3)** Proceso estaba ejecutando en modo sistema y la dirección de fallo comienza con un 0. Distinga entre el caso de que el fallo se produzca durante una llamada al sistema o en una rutina de interrupción.
- a4)** Proceso estaba ejecutando en modo sistema y la dirección de fallo comienza con un 1. Distinga entre el caso de que el fallo se produzca durante una llamada al sistema o en una rutina de interrupción.

b) [Puntuación: 2'5/10] Supóngase que, no habiendo ningún marco libre y siendo la página que lleva más tiempo residente una que se ha modificado por primera vez y que corresponde con la región de datos con valor inicial, se produce un fallo de página que conlleva una expansión de la pila en una página. Analice qué operaciones de las especificadas a continuación se llevan a cabo dentro de la rutina de tratamiento de ese fallo.

- Operaciones vinculadas con el *swap* (reserva, lectura y escritura) y con el sistema de ficheros (lectura y escritura). Responda primero suponiendo que se trata de un sistema con preasignación de *swap* y luego considerando que no tiene preasignación.
- Modificaciones de los campos de las entradas de las tablas de páginas: bit de validez, bit de modificado, bit de referencia, número de marco y protección.
- Cambios de contexto, especificando si son voluntarios o involuntarios.

c) Supóngase que se produce un fallo de página sobre una página de código que, al no haber marcos libres, causa la expulsión de una página modificada de una región que corresponde con un fichero proyectado en modo compartido. Especifique para cada uno de los dos escenarios planteados a continuación qué activaciones del S.O. se producen desde que ocurre el fallo hasta que el proceso implicado vuelve a ejecutar, especificando para cada activación si hay cambios de estado y/o de contexto, distinguiendo si son voluntarios o involuntarios.

- c1) [Puntuación: 2'5/10]** El proceso que causa el fallo tiene alta prioridad. Existe otro proceso de baja prioridad que la mayor parte del tiempo está realizando llamadas al sistema no bloqueantes (por ejemplo, `getpid`).
- c2) [Puntuación: 2'5/10]** El proceso que causa el fallo tiene baja prioridad. Existe otro proceso de alta prioridad que alterna rachas de ejecución en modo usuario con llamadas al sistema para dormirse un cierto tiempo (un intervalo de tiempo pequeño, inferior a la duración de un acceso a disco).

Solución

a1) Si la dirección de fallo corresponde con una dirección lógica de usuario y el proceso estaba en modo usuario cuando causó el fallo, hay que comprobar si se trata de una expansión de pila o de un error:

- Si la dirección está incluida en el rango que va desde el valor del puntero de pila en el momento de producirse el fallo y el principio de una región de tipo pila (en Linux está marcada con el indicador `MAP_GROWSDOWN`), hay una expansión de la pila, que conlleva ajustar la región, comprobando previamente que no se solapa con otras regiones.
- Si no, se trata de un error del programa, se aborta el proceso (en UNIX se manda la señal `SEGV`).

a2) Si la dirección de fallo corresponde con una dirección lógica de sistema y el proceso estaba en modo usuario cuando causó el fallo, se trata de un acceso inválido. Por tanto, es una situación posible y su tratamiento será abortar el proceso (en UNIX mandar la señal `SEGV`).

a3) Si la dirección de fallo corresponde con una dirección lógica de usuario y el proceso estaba en modo sistema

cuando causó el fallo, hay que distinguir si se estaba ejecutando una llamada al sistema o una interrupción:

- En una llamada al sistema. Desde el código de la llamada se ha accedido al mapa de usuario del proceso para leer/escribir algún parámetro de la llamada. Como en el primer apartado, pueden producirse dos situaciones:
 - Si la dirección está incluida en el rango que va desde el valor del puntero de pila en el momento de producirse el fallo y el principio de una región de tipo pila, hay una expansión de la pila.
 - Si no, se trata de un error del programa, se puede devolver un error en la llamada (del tipo EINVAL en UNIX) o abortar el proceso.
- En una interrupción. Si el S.O. no tiene errores de programación, como plantea el enunciado, esta situación no puede ocurrir. Téngase en cuenta que una interrupción tiene un carácter asíncrono y no está directamente vinculada con el proceso en ejecución, por lo que la rutina de tratamiento no tiene que acceder al mapa de usuario del proceso.

a4) El caso de que la dirección de fallo corresponda con una dirección lógica de sistema y el proceso estaba en modo sistema cuando causó el fallo no es posible, ni desde una llamada ni desde una rutina de interrupción, ya que el S.O. no tiene errores de programación.

b) A continuación, se analizan qué operaciones de cada uno de los tres grupos se llevan a cabo en la rutina de tratamiento del fallo de página planteado:

- Operaciones vinculadas con el swap y con el sistema de archivos. Dependiendo de si hay preasignación de swap:
 - Si hay preasignación, en primer lugar, habrá que escribir en el bloque de swap ya reservado previamente la página expulsada, puesto que se trata de una página de una región privada. Por lo que se refiere a la página que causó el fallo, que corresponde con una expansión de pila, no hay que leerla del disco, puesto que pertenece a una región anónima (se rellena con ceros por confidencialidad). Sin embargo, dado que hay preasignación, es necesario realizar una reserva en el swap para la zona expandida de la pila.
 - Si no hay preasignación, dado que es la primera vez que se expulsa esa página estando modificada, hay que reservar espacio en swap para ella. A continuación, se escribe la página expulsada en dicho bloque. En cuanto a la página correspondiente a la expansión, no implica ninguna operación sobre el swap.
- Modificaciones de los campos de las entradas de las tablas de páginas. Hay que distinguir entre la página que se expulsa y la que causó el fallo:
 - Página expulsada:
 - bit de validez: se desactiva puesto que ya no está residente.
 - bit de modificado: se desactiva ya que después de escribirla a swap, ya está actualizada.
 - No hace falta modificar los otros campos: el número de marco, ya que no es significativo al estar marcada como inválida, el bit de referencia, puesto que no se usa al utilizarse un algoritmo FIFO, y la protección, dado que no cambia por ser expulsada.
 - Página que causó el fallo:
 - bit de validez: se activa puesto que ahora está residente.
 - bit de modificado: se inicia como desactivado.
 - número de marco: se especifica el número de marco liberado, el que contenía la página expulsada.
 - protección: se inicia con el valor correspondiente a la región (al tratarse de la pila, permisos de lectura y escritura).
 - bit de referencia: no se modifica ya que no se usa.
- Cambios de contexto, especificando si son voluntarios o involuntarios. Dentro de la rutina de fallo, habrá un cambio de contexto voluntario debido a que el proceso debe bloquearse mientras se escribe a swap la página expulsada.

c1) A continuación, se describen las activaciones del S.O. que se producen en la situación planteada:

- El proceso de alta prioridad provoca un fallo de página y pasa a ejecutar la rutina de fallo en modo sistema. Como la página expulsada pertenece a una proyección compartida de un archivo, se debe escribir en el bloque correspondiente del archivo. Se programa la operación de escritura por DMA y se bloquea el proceso, produciéndose un cambio de contexto voluntario al proceso de baja prioridad.
- Se produce la interrupción del disco que activa la rutina de tratamiento correspondiente. Dentro de esta rutina se desbloquea el proceso bloqueado y, al detectarse que tiene mayor prioridad que el que está ejecutando, se requiere un cambio de contexto involuntario, que se difiere activando una interrupción software.
- Cuando acaba la rutina de interrupción y todas las rutinas del S.O. que pudieran estar anidadas (por ejemplo, una

llamada al sistema del proceso de baja prioridad), se activa la rutina de tratamiento de la interrupción software, que realiza el cambio de contexto involuntario al proceso de mayor prioridad.

- El proceso de alta prioridad continúa la ejecución de la rutina de fallo de página desde el punto donde se quedó bloqueado. A continuación, hay que leer la página de código. Una vez programada la operación por DMA, se bloquea nuevamente el proceso, produciéndose un cambio de contexto voluntario al proceso de baja prioridad.
- Como antes, se produce la interrupción del disco que activa la rutina de tratamiento correspondiente. Dentro de esta rutina se desbloquea el proceso bloqueado y, al detectarse que tiene mayor prioridad que el que está ejecutando, se requiere un cambio de contexto involuntario, que se difiere activando una interrupción software.
- Cuando acaba la rutina de interrupción y todas las rutinas del S.O. que pudieran estar anidadas, se activa la rutina de tratamiento de la interrupción software, que realiza el cambio de contexto involuntario al proceso de mayor prioridad.
- El proceso de alta prioridad continúa la ejecución de la rutina de fallo de página, la termina y vuelve a modo usuario ejecutando de nuevo la instrucción que causó el fallo, que esta vez no lo provocará.

c2) A continuación, se describen las activaciones del S.O. que se producen en la situación planteada:

- Estando el proceso de alta prioridad bloqueado esperando un plazo de tiempo, ejecuta el proceso de baja prioridad que provoca un fallo de página y pasa a ejecutar la rutina de fallo en modo sistema. Como la página expulsada pertenece a una proyección compartida de un archivo, se debe escribir en el bloque correspondiente del archivo. Se programa la operación de escritura por DMA y se bloquea el proceso, produciéndose un cambio de contexto voluntario al proceso nulo, suponiendo que el de alta prioridad sigue dormido.
- Se produce una interrupción de reloj que desbloquea al proceso dormido que, al detectarse que tiene mayor prioridad que el que está ejecutando (el proceso nulo), requiere un cambio de contexto involuntario, que se difiere activando una interrupción software.
- Se activa la rutina de tratamiento de la interrupción software, que realiza el cambio de contexto involuntario al proceso de mayor prioridad.
- Se produce la interrupción del disco que activa la rutina de tratamiento correspondiente. Dentro de esta rutina se desbloquea el proceso bloqueado pero, al detectarse que tiene menor prioridad que el que está ejecutando, no se lleva a cabo un cambio de contexto.
- Cuando el proceso de alta prioridad realiza una llamada al sistema para dormirse un plazo de tiempo, se produce un cambio de contexto voluntario al proceso de baja prioridad, que continúa la ejecución de la rutina de fallo de página desde el punto donde se quedó bloqueado, por lo que tiene que leer la página de código. Una vez programada la operación por DMA, se bloquea nuevamente el proceso, produciéndose un cambio de contexto voluntario al proceso nulo, suponiendo que el de alta prioridad sigue dormido.
- Se produce una interrupción de reloj que desbloquea al proceso dormido activándose la interrupción software para llevar a cabo un cambio de contexto involuntario diferido.
- Se activa la rutina de tratamiento de la interrupción software, que realiza el cambio de contexto involuntario al proceso de mayor prioridad.
- Se produce la interrupción del disco que activa la rutina de tratamiento correspondiente. Dentro de esta rutina se desbloquea el proceso bloqueado pero, al detectarse que tiene menor prioridad que el que está ejecutando, no se lleva a cabo un cambio de contexto.
- Cuando el proceso de alta prioridad realiza una llamada al sistema para dormirse nuevamente un plazo de tiempo, se produce un cambio de contexto voluntario al proceso de baja prioridad, que continúa la ejecución de la rutina de fallo de página desde el punto donde se quedó, la termina y vuelve a modo usuario ejecutando de nuevo la instrucción que causó el fallo, que esta vez no lo provocará.