Complemento para ejercicio 4 (paginación):

Para dar un poco más de utilidad al ejercicio vamos a agregar la capacidad de detectar una página no presente y asignarle una dirección física válida. De esta forma estamos accediendo a direcciones "no identity mapping"

1) Modificar el ejercicio de modo que en el loop del final se genere un valor aleatorio acotado entre dos límites que corresponderá al rango de direcciones virtuales que quiero acceder. Para ello escribir el código ensamblador correspondiente al algoritmo generador de números aleatorios descripto abajo.
2) Este número aleatorio lo utilizaremos para acceder a memoria (lectura/escritura)
   En caso que la dirección esté paginada, no sucederá nada fuera de lo normal.
   Si la dirección no está paginada se generará una excepción de DATA ABORT:
3) Escribir el handler de DATA ABORT contemplando los siguientes pasos:
   Dividiremos la dirección virtual en sus tres campos respectivos. Accedemos a la tabla de nivel 1 en busca de la entrada correspondiente.
   Si ésta existe, accedemos a la tabla de nivel 2 que corresponda y llenamos la entrada con la dirección de la nueva página asignada.
   Si la entrada de nivel 1 no existe, debemos crear una nueva tabla de nivel 2, llenar la entrada de la tabla de nivel 1, y cargar en la de nivel 2 la dirección de la nueva página asignada
4) Definiremos un espacio de memoria donde residirán las tablas de paginación. Un puntero (ptrT) nos servirá para ubicar las nuevas tablas de nivel 2
5) Definiremos un espacio de memoria "USR MEM" que será asignada a los accesos antes mencionados. Un puntero (ptrM) nos servirá para asignar las nuevas páginas creadas.

Generación de número pseudo-aleatorios: Método de Congruencias lineales

Semilla: T (32 bits)

$T_{n+1} = 0x314159265 * T_n + 0x1234567$ (se ignora la parte alta)