

4. Dado un proceso que está en modo “ejecutándose” y pretende acceder a una dirección de memoria fuera del área asignada, lo cual sería un error en la ejecución, ¿a qué modo pasaría dicho proceso? Razone la respuesta.

- (a) Bloqueado.
- (b) No cambia de modo.
- (c) Finalizado.**
- (d) Preparado.

La correcta es la C. Si al proceso se le asigna una dirección de memoria que se encuentre fuera del área asignada este proceso muere (finaliza) debido a que dicha dirección de memoria NO EXISTE, convirtiendo dicha instrucción en un error no recuperable.

19. Suponga que tenemos 3 procesos ejecutándose concurrentemente en un determinado instante. El sistema operativo utiliza un sistema de memoria con paginación. Se dispone de una memoria física de 131072 bytes (128K). Sabemos que nuestros procesos al ser ejecutados tienen los parámetros que se muestran en la tabla.

Los datos indican el tamaño en bytes de cada uno de los segmentos que forman parte de la imagen del proceso. Sabiendo que una página no puede contener partes de dos segmentos diferentes (pila, código o datos), hemos de determinar el tamaño de página que debería utilizar nuestro sistema y se barajan dos opciones: páginas de 4096 bytes (4K) o páginas de 512 bytes (1/2K). Se pide:

- a) ¿Cuál sería la opción más apropiada, 4096 bytes o 512 bytes? Justifica totalmente la respuesta mostrando todos los cálculos que has necesitado para llegar a dicha conclusión.
- b) ¿Cuál es el formato de cada entrada de la Tabla de Páginas con el tamaño de página elegido? Justifica el tamaño de los campos con direcciones. Puedes añadir los bits que consideres necesarios para el buen funcionamiento del sistema indicando para que van a ser utilizados.
- c) ¿Cuántas Tablas de Páginas habrá en este sistema? ¿Cuántas entradas hay en cada tabla de páginas (filas)?

a)

Con 4K: dividimos todos los datos de la tabla (código, pila y datos) entre 4K y los sumamos entre sí para cada proceso (A, B, C)

PROCESO	CODIGO	PILA	DATOS	PAGINAS
A	20480/4096 (5 páginas)	14288/4096 (4 páginas)	10240/4096 (3 páginas)	12
B	16384/4096 (4 páginas)	8200/4096 (3 páginas)	8192/4096 (2 páginas)	9
C	18432/4096 (5 páginas)	13288/4096 (4 páginas)	9216/4096 (3 páginas)	12
TOTAL				33 paginas *4096=135168 bytes de memoria

Por lo que con páginas de 4K excedemos la memoria disponible, por lo que no nos es posible paginar con dicho tamaño de página.

Con 1/2K: dividimos todos los datos de la tabla (código, pila y datos) entre 1/2K y los sumamos entre sí para cada proceso (A, B, C)

PROCESO	CODIGO	PILA	DATOS	PAGINAS
A	20480/512 (40 páginas)	14288/512 (28 páginas)	10240/512 (20 páginas)	88
B	16384/512 (32 páginas)	8200/512 (17 páginas)	8192/512 (16 páginas)	65
C	18432/512 (36 páginas)	13288/512 (26 páginas)	9216/512 (18 páginas)	80
TOTAL				233 paginas *512=119296 bytes de memoria

La mejor opción sería paginar en 1/2K. Pero la única pega sería que perderíamos rendimiento.

b)

c)

Como tenemos 3 procesos necesitamos 3 páginas de tablas y como mínimo necesitamos una entrada por cada página por lo que:

Proceso A = 88 entradas

Proceso B = 65 entradas

Proceso C = 80 entradas