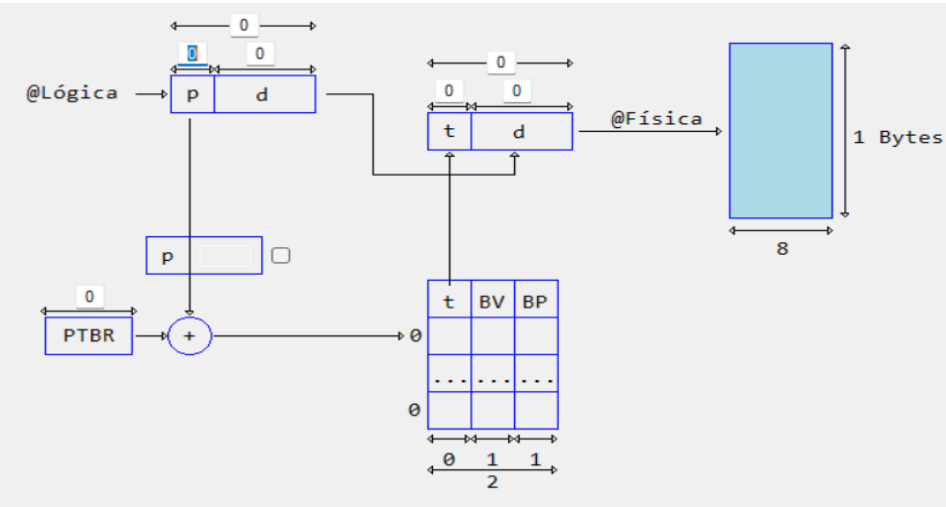


2.MEMORIA:

2.1.1-Paginacion:



2.1.1.2-TLB:

-sin identificador de proceso:

Numero de pagina	Contenido de la tabla de pagina

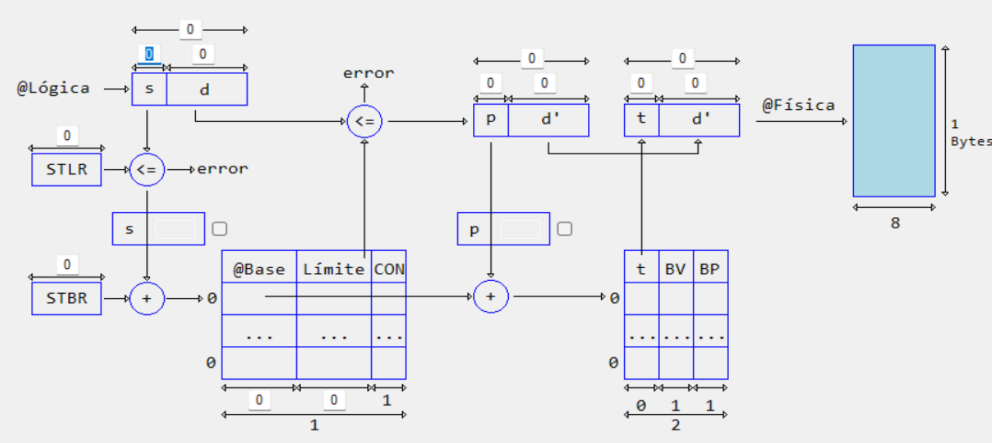
-con identificador de proceso:

Numero de pagina p	Contenido de la tabla de pagina	Identificador de proceso pid

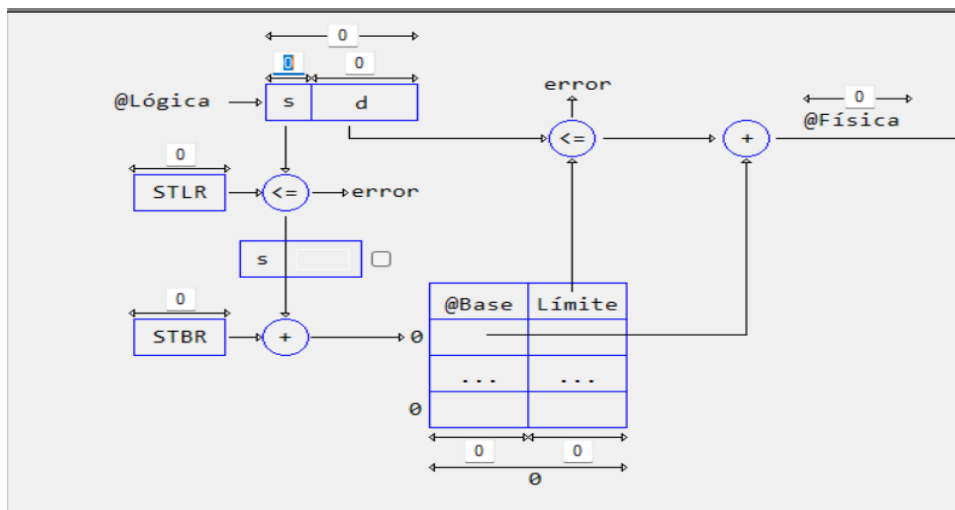
2.1.1.4-Multinivel:

@base	Contenido de la tabla sin bit de presencia

2.1.2-Segmentacion paginada:



### 2.1.3-Segmentación:



### 2.1.4-Tener en cuenta:

- 1-memoria virtual implica bit de presencia.
- 2-si no hay bit de validez en la tabla de páginas hay q añadir un PTLR para controlar el límite.
- 3-si el ancho de memoria es menor que la tabla de segmento o paginas se tiene que añadir en el cuadro zeros según el ancho de memoria.

### 2.2-FORMULÁS:

- 1-tamaño de  $STBR = @BASE = @FISICA = \text{tamaño de la memoria} / \text{el ancho de la memoria en bytes}$ .
- 2-tamaño de  $STLR = s = \text{número máximo de segmentos}$ .
- 3-tamaño de  $LIMITE = d + d' = \text{tamaño de segmento} / \text{el ancho de la memoria en bytes}$ .
- 4-tamaño de segmento  $= 2^d * \text{el ancho de la memoria en bytes}$ .
- 5-tamaño de página  $= 2^{d'} * \text{el ancho de la memoria en bytes}$ .
- 6-tamaño de la tabla de segmentos  $= \text{el ancho de la tabla de segmentos} * 2^s$ .
- 7- tamaño de la tabla de páginas  $= \text{el ancho de la tabla de páginas} * 2^p$ .
- 8-tamaño de un proceso como máximo  $= 2^{@logica} * \text{el ancho de la memoria en bytes}$ .

### 2.3-Algoritmos:

- 1-optimó: el que lleva más en referenciarse futuramente se sustituye.
- 2-FIFO: La página a reemplazar será aquella que hace más tiempo que ha sido cargada en memoria.
- 3-LRU: la página que hace más tiempo que no se ha usado se sustituye (se puede añadir contador en la tabla de página...).
- 4-NRU: hay q añadir bit de referencia(lectura) y modificada (Escritura) y se elimina aquel q tenga el valor mínimo En caso de empate se sustituye aquel q lleva más tiempo. En caso del

Working set el bit de modificada una vez q se pone a 1 ya no se cambia durante toda la ejecución.

5-AGING: hay q añadir el bit de referencia la q se reemplaza será aquella que hace más tiempo que ha sido cargada en memoria.

6-RELOJ: como fifo, pero el q entró hace más tiempo solo se reemplaza si su bit de referencia vale 0 si vale 1 se pone al fin de la cola con referencia a 0.

7-LFU: se reemplaza aquella página aquella página que tenga el contador más bajo ( el menos frecuentemente usado) se va añadiendo un contador.

8-MFU: lo contrario q la LFU se reemplaza el q tenga el contador más alto.

WORKING SET: cada  $Ws=n$  se modifica el bit de referenciada a 0.

2.4-En relación con las tablas:

@base=la dirección donde comienza la tabla de página (stbr).

Limite= (tamaño del segmento/ancho de memoria en bytes)-1. -->ya q empieza desde el 0

Tamaño de un proceso cargado=(limite+1) \*ancho de la memoria.

Número de páginas validas=tamaño de segmento/tamaño de páginas.

Número de páginas del proceso=suma del número de páginas validas de todos los segmentos

Si el tamaño de las tablas es mayor q de la memoria va saltando

Por ejemplo: si la tabla de segmentos es el doble va saltando de dos en dos

	@base	limite
500	x	y
502	x2	y2

3.-FICHEROS:

Tamaño del bloque=tamaño del disco/número de bloques.

Tamaño de disco =cantidad de sectores \* tamaño del sector.

Cantidad de sectores=número de sectores \*número de pistas \* superficies.

Tamaño de la fat = número de bloques\*(2 si es de 16 y 4 si es de 32).

$2^n \geq$  número de bloques.  $\rightarrow n$ =bloques de la fat (apuntadores en bits).

Tamaño de mapa de bits de i-nodos= $n$ .  $\rightarrow 2^n$ =número máximo de i-nodos.

Numero de i nodos ejemplo hay 64 i\_nodos desde 0 hasta 63 Son 64 bits 8bytes.número bloque de disco ejemplo 2048 bloques son 256bytes.

Tamaño de mapa de bits ocupa=264bytes.

Tamaño de la tabla de i-nodos=tamaño de i-nodos/tamaño del bloque.

tamaño de i-nodos=tamaño de (apuntadores+info) \*número de i-nodos.

Numero de apuntadores cada bloque=tamaño de bloque/tamaño de apuntador en bytes.

Tamaño de fichero máximo = tamaño de bloque \* (Numero de apuntadores cada bloque simple+dobles<sup>2</sup>+tribles<sup>3</sup>+número de bloques de apuntadores directos).

Numero de fichero=número máximo de i-nodo=tam tabla i-nodo/tam de cada i-nodo.

-----MFT-----

Bits de extensión=bits de n°bloque.

Tamaño de fichero nfts=2<sup>bits de extensión</sup> \*tamaño de bloque \*tamaño de datos de cada entrada/bits de (extensión+primer bloque).

Numero de extensiones se pueden almacenar= tamaño de datos de cada entrada/ (extensión+primer bloque)en bytes.

Numero de bloques=2<sup>bits del 1ºbloque</sup>. Tamaño de extensión=2<sup>bits de extensión</sup> \*tamaño de bloque.

Tamaño de fichero=tamaño de fichero nfts=tamaño de extensión \* número de extensiones.

Tamaño de la mft=ancho de la entrada\*número de entradas.

Numero de ficheros=tamaño índice\*8/x.-->2<sup>x</sup>=número de entradas de la mft.

4-entarda y salida:

FCFS/FIFO: el primero q llega es el q se atiende.

SSTF: accede a la pista que está más cerca donde está ahora mismo.

SCAN: dada una dirección empieza a acceder como sstf después para cambiar dirección hay q rebotar en 0 si vas abajo o la máxima pista.

LOOK: SCAN sin rebotar cambia directamente al siguiente.

SCAN N Pasos: divide la cola en subcolas de longitud N y las atiende como scan.

F-SCAN: divide las colas de manera dinámica.

C-SCAN: Sólo barre el disco en un sentido, cuando termina comienza por el principio, y en el retorno no atiende a ninguna.

C-LOOK: lo mismo q c-scan pero sin rebotar por los extremos.

tipo	Tamaño de almacenamiento	Tamaño de seguridad
RAID 0	Suma de Tamaño de todos los discos	0
RAID 1	Tamaño de un bloque	El tamaño del resto de bloques
RAID 1+0	La mitad del tamaño total	La mitad del tamaño total
RAID 0+1	La mitad del tamaño total	La mitad del tamaño total
RAID 5	El tamaño del resto de bloques	Tamaño de un bloque

Total de pistas=cantidad de sectores/número de pistas de cada sector\*cara.

1-PROCESO:

$$T_s = t_f - t_i.$$

$$T_e = t_s - t.$$

$$I = t/t_s.$$