

## Ejercicio 1

Dirección Lógica	Solicitud	Dirección Física
0-430	R	670
0-150	W	Violación de Seguridad
1-15	R	2315
2-130	W	Desplazamiento fuera del rango
4-25	X	Segmento Faltante

## Ejercicio 2

- a) Para el programa A se necesitan  $\frac{4300bytes}{128bytes} = 33.59 \rightarrow 34$  páginas como mínimo. Para el programa B se necesitan al menos  $\frac{3068bytes}{128bytes} = 23.97 \rightarrow 24$  páginas.
- b) Dado que las páginas son de 128KB, el proceso A utiliza 34 paginas, es decir,  $34 * 128KB = 4354$  KB. Entonces el proceso A desperdicia 54 KB. En cuanto al proceso B, utiliza 24 páginas, es decir,  $24 * 128KB = 3072$  KB. Entonces el proceso B desperdicia 4 KB. El método por paginación no desperdicia memoria por fragmentación externa.
- c) La tabla de paginación del proceso A pesará:  $34 * 8 \text{ bytes} = 272 \text{ bytes}$ .  
La tabla de paginación del proceso B pesará:  $24 * 8 \text{ bytes} = 192 \text{ bytes}$ .

## Ejercicio 3

La instrucción LDR en arquitectura ARM permite un formato que incluye cargar un valor constante de 32-bits desde la versión ARMv6T2. Esto es posible porque la arquitectura reserva un espacio de memoria al final de cada sección de código llamado `literal pool`. Su propósito es justamente cargar valores constantes o direcciones de etiquetas, que su tamaño esta fuera del rango de las instrucciones MOV y MVN, que luego serán cargados a un registro.

Cuando se llama a esta instrucción con un valor constante de 32-bits, el ensamblador realiza lo siguiente:

- Verifica si el valor no se encuentra ya en alguna `literal pool`. Si es así, carga la dirección de la constante ya existente.
- En caso contrario, coloca el valor en la siguiente `literal pool` disponible y carga esa nueva dirección.

Sin embargo, si la siguiente `literal pool` esta fuera de rango, el ensamblador genera un mensaje de error.