

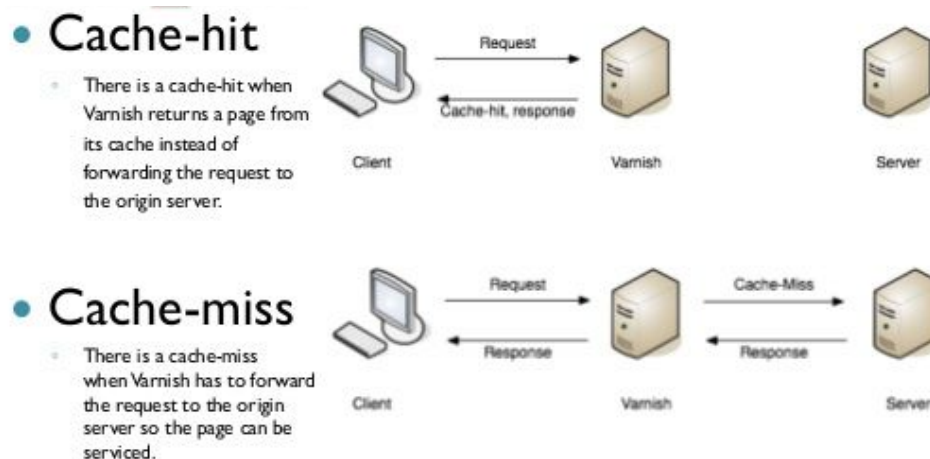
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
PROGRAMACIÓN DE SISTEMAS
Paralelo 3

Proyecto Primer Parcial 2017-II

Antecedentes

Memoria caché: Es una memoria donde se almacena información de manera temporal, con el objetivo de reducir latencia e incrementar el rendimiento de los sistemas.

Cuando un cliente desea acceder a un objeto, se consulta si éste ya reside en la memoria caché, en cuyo caso se lo accede localmente (HIT). Por el contrario, si el objeto solicitado no reside en la memoria caché (MISS), se lo obtiene del almacenamiento principal y se lo almacena en la caché, de manera que un pedido posterior al mismo objeto lo encontrará almacenado en la memoria caché.



* <https://www.slideshare.net/boyron/varnish-cache-44387846>

La memoria caché tiene una capacidad de almacenamiento limitada. Cuando la caché está llena y se desea agregar un nuevo objeto, se debe **desalojar (evict)** un objeto ya existente. La elección del objeto a ser desalojado se toma en base a una **política de desalojo (eviction policy)**. Una manera de medir la efectividad de una memoria caché, es calculado la tasa de fallas (miss rate), que se obtiene sumando todas las operaciones que provocaron un MISS y se divide para el número total de operaciones.

La política de desalojo (eviction policy) más utilizada es *LRU (Least Recently Used - Menos recientemente usado)*. Con esta política, cuando la caché está llena y se desea agregar un nuevo objeto, se busca cuál es el objeto alojado que lleva más tiempo sin ser accedido y se lo elimina. En su lugar, se agrega el nuevo objeto y se lo marca como el más recientemente accedido.

Para que LRU funcione, se debe llevar un registro del orden de los objetos basados en su acceso más reciente. Es común implementar este tipo de registros con una lista doblemente enlazada.

Adicionalmente, para acceder de manera rápida a un elemento particular de la lista doblemente enlazada, se utiliza una estructura adicional llamada “tabla hash” .

Enunciado del Proyecto

Desarrollar un simulador de memoria caché con una política de desalojo LRU (Least Recently Used), utilizando el Lenguaje de Programación C.

- Utilizar solo las claves (no se almacenará el valor de los objetos)
- Las claves estarán formadas de 6 caracteres alfanuméricos.
- Para la ejecución el listado de claves estará en un archivo de texto, una clave por cada línea, que se introducirá mediante entubamiento de entrada.
- El tamaño de la caché se ingresará como un argumento al programa
- Implementar y utilizar una lista doblemente enlazada
- Implementar y utilizar una tabla hash, utilizando la función hash explicada más adelante.
- La función de hash a utilizar con las claves (cadenas de 6 caracteres alfanuméricos) es la siguiente: Sumar los valores ASCII de cada caracter multiplicados por su posición en la cadena. Luego a la suma total hacer módulo con un número primo determinado por cada grupo (ejemplo: 2017)

Ejemplo:

```
Cadena:      aaccee
Hash:        (97*1 + 97*2 + 99*3 + 99*4 + 101*5 + 101*6)%2017 = 78
```

- El programa debe producir como salida, por cada registro ingresado, la clave del registro y si se trató de un **hit** o de un **miss**.
- Al final de la ejecución debe mostrar la tasa de misses o fallas (número total de misses / número total de registros).

Ejemplo de ejecución:

```
cat listado_prueba.txt | ./simuladorLRU -s 50
```

Donde:

- El archivo “**listado_prueba.txt**” contiene el listado de claves para comprobar el funcionamiento del simulador.
- El archivo “**simuladorLRU**” es el archivo ejecutable de su programa.
- El valor **50** es el tamaño de la memoria caché (este valor puede cambiar).