

# Fundamentos de Organización de Datos

*Hashing*

# Dispersión de Archivos

- ✓ Técnica para generar una dirección base única para una clave dada.
- ✓ Convierte la clave en un número aleatorio, que luego sirve para determinar dónde se almacena la clave.
- ✓ Utiliza una función de dispersión para mapear cada clave con una dirección física de almacenamiento.
- ✓ Utilizada cuando se requiere acceso rápido por clave.

# Tipos de Dispersión

## **Direccionamiento estático**

El espacio disponible para dispersar los registros del archivo está fijado previamente.

## **Direccionamiento dinámico**

El espacio disponible para dispersar los registros del archivo aumenta o disminuye en función de las necesidades.

# Parámetros a considerar

Parámetros que influyen sobre el desempeño del ambiente de dispersión:

- ✓ Capacidad de almacenamiento de cada dirección
- ✓ Densidad de empaquetamiento
- ✓ Función de hash
- ✓ Método de tratamiento de desbordes

# Dispersión de Archivos

## **Función de dispersión**

Caja negra que a partir de una clave genera la dirección física donde debe almacenarse el registro.

## **Colisión**

Situación en la que un registro es asignado, por función de dispersión, a una dirección que ya posee uno o más registros.

# Dispersión de Archivos

## Desborde

Situación en la cual una clave carece de lugar en la dirección asignada por la función de dispersión.

## Densidad de empaquetamiento

Relación entre el espacio disponible para el archivo de datos y la cantidad de registros que integran el mismo.

$$DE = \text{número de registros} / \text{espacio Total}$$

# Dispersión de Archivos

Aunque la función de dispersión sea eficiente y la densidad de empaquetamiento sea baja, es probable que ocurran **desbordes**.

Métodos aplicables para resolver colisiones con desborde en *dispersión estática*:

- ✓ ***Saturación progresiva***
- ✓ ***Saturación progresiva encadenada***
- ✓ ***Saturación progresiva con área de desborde por separado***
- ✓ ***Dispersión doble***

## Ejemplo

Técnica de resolución de colisiones:

### **Saturación Progresiva**

Memoria de 11 direcciones (0..10)

Capacidad = 2 registros por dirección

$$f(x) = x \bmod N$$

Se van a dispersar 15 claves en total.



## Tabla inicial

Claves a dispersar:

- 58 →  $f(58) = 3$
- 78 →  $f(78) = 1$
- 60 →  $f(60) = 5$
- 85 →  $f(85) = 8$
- 91 →  $f(91) = 3$
- 27 →  $f(27) = 5$

Dirección	Registro	Registro
0		
1	78	
2		
3	58	91
4		
5	60	27
6		
7		
8	85	
9		
10		

Siguientes claves  
a dispersar:

- 25 →  $f(25) = 3$
- 38 →  $f(38) = 5$
- 81 →  $f(81) = 4$
- 14 →  $f(14) = 3$

En rojo las *claves intrusas*

Dirección	Registro	Registro
0		
1	78	
2		
3	58	91
4	25	81
5	60	27
6	38	14
7		
8	85	
9		
10		

- 22 →  $f(22) = 0$
- 23 →  $f(23) = 1$
- 56 →  $f(56) = 1$
- 76 →  $f(76) =$
- 10 →

¿Cuál es la  $f(21) = 10$

$$DE = \frac{15}{11*2} = \frac{15}{22} = 68\%$$

Dirección	Registro	Registro
0	22	
1	78	23
2	56	
3	58	91
4	25	81
5	60	27
6	38	14
7		
8	85	
9		
10	76	21

## Eliminar las siguientes claves

- -22 →  $f(22) = 0$
- -58 →  $f(58) = 3$
- -81 →  $f(81) = 4$
- -14 →  $f(14) = 3$

Se usa una marca (####) cuando se elimina un dato y en la próxima cubeta hay otra marca u otro dato (Ej 58, 81)

Dirección	Registro	Registro
0	22	
1	78	23
2	56	
3	<del>58</del>	91
4	25	<del>81</del>
5	60	27
6	38	14
7		
8	85	
9		
10	76	21