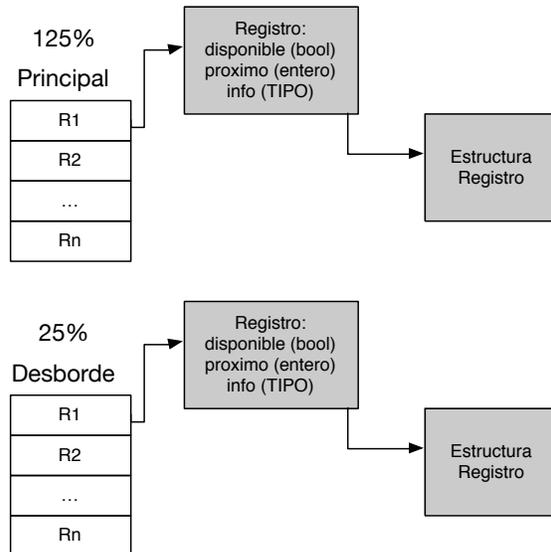


Archivos Hash con resolución de colisiones por lista en archivo desborde
Prof. Andrés Arcia-Moret
Junio – 2014



Crear: Crea un archivo principal con una cantidad preestablecida de registros. Regularmente se utiliza las proporciones $cte1 = 125\% T$ y $cte2 = 25\% T$, donde T es el tamaño del archivo estimado por el usuario. Se crea un registro vacío que sirve para iniciar todos los archivos. La escritura de registros vacíos se hace en el archivo principal y desborde con el objeto de poder manejar archivos de tamaño máximo. Adicionalmente se crea una pila donde residirán los registros disponibles.

1. TAM_MAX_P= cte1*t
2. TAM_MAX-D= cte2*t
3. TAM_ACT_P=0, TAM_ACT_D=0
4. R.hab=verdadero
5. r.sig=-1
6. Abrir PRINCIPAL
7. RP(i=0; i< TAM_MAX_P; i++)
 - 7.1 Escribir (P,i,R)
8. Abrir DESBORDE
9. RP(i=0;i< TAM_MAX_D;i++)
 - 9.1 PILA_DESBORDE.push(i)
 - 9.2 ESCRIBIR (D,I,R)
10. Cerrar PRINCIPAL, DESBORDE;

Abrir Archivo

1. TAM_MAX_P= TAM_MAX_D= TAM_ACT_P= TAM_ACT_D=0
2. Abrir PRINCIPAL
3. continuar=verdadero
4. RM (continuar=verdadero)
 - 4.1 Si (leer(P,TAM_MAX_P,R)==verdadero)
 - 4.1.1 TAM_MAX_P ++
 - 4.1.2 SI (R.hab==falso)
 - 4.1.2.1 TAM_ACT_P++
 - Sino
 - 4.1.3 continuar=falso
5. Cerrar PRINCIPAL
6. Abrir DESBORDE
7. continuar=verdadero
8. RM (continuar=verdadero)
 - 8.1 Si (leer(D, TAM_MAX_D, R) == verdadero)
 - 8.1.1 TAM_MAX_D ++
 - 8.1.2 SI (R.hab==verdadero)
 - 8.1.2.1 PILA_DESBORDE.push(TAM_MAX_D -1)
 - Sino
 - 8.1.2.2 TAM_ACT_D ++
 - Sino
 - 8.1.3 continuar=falso
9. cerrar DESBORDE

Buscar Registro

R = registro que contiene la clave a buscar

S = registro existente en una posición cualquiera

1. POS= HASH(R.clave)
2. Si (POS<0 v POS> TAM_MAX_P)
 - 2.1. Retornar ERROR
3. Leer (P,POS,S)
4. Si (S.hab==falso)
 - 4.1. Si (R.clave==S.clave)
 - 4.1.1.Cerrar PRINCIPAL
 - 4.1.2.Devolver S
 - Sino
 - 4.1.3.SI (S.sig i=-1)
 - 4.1.3.1. Abrir DESBORDE
 - 4.1.3.2. continuar = verdadero
 - 4.1.3.3. RM (continuar)
 - 4.1.3.3.1 Leer(D,S.sig,S)
 - 4.1.3.3.2 Si (S.sig===-1)
 - 4.1.3.3.2.1. continuar=falso
 - 4.1.3.3.3 Si (S.clave==R.clave)
 - 4.1.3.3.3.1.Cerrar PRINCIPAL, DESBORDE
 - 4.1.3.3.3.2.Retornar S
 - 4.1.3.4. Cerrar DESBORDE
5. Cerrar PRINCIPAL

Agregar Registro

Inserta un registro nuevo (NUEVO_REG) en el archivo principal (PRINC). Hay tres casos de importancia. Primero si el registro esta vacío, se coloca directamente el registro a insertar en el principal (se sobrescribe el registro vacío). Si el registro principal está lleno, pero no tiene desborde, se escribe el primer desborde sin olvidar reescribir el principal con diferente puntero. Por último, se busca el último de la lista de desborde (5.6.5) hasta para poder concatenarlo con el penúltimo en el desborde.

1. pos = **HASH** (NUEVO_REG.clave)
2. Si (pos < 0 v pos >= TAM_MAX_P)
 - 2.1. Retornar ERROR
3. **PREPARAR_REGISTRO** (NUEVO_REG.data,
 NUEVO_REG.hab=falso,
 NUEVO_REG.sig=-1)
4. **LEER** (PRINC, pos, REG_EXIST)
5. Si (REG_EXIST.hab == verdadero)
 - 5.1. **ESCRIBIR** (PRINC, pos, NUEVO_REG)
 - 5.2. **CERRAR** PRINC
 - 5.3. TAM_ACT_P ++
 - 5.4. Retornar ÉXITOSino
 - 5.5. **ABRIR** DESBORDE
 - 5.6. Si (REG_EXIST.sig== -1)
 - 5.6.1. REG_EXIST.sig=PILA_DESBORDE.pop()
 - 5.6.2. Si (REG_EXIST.sig== -1)
 - 5.6.2.1. **CERRAR** PRINCIPAL, DESBORDE
 - 5.6.2.2. **RETORNAR** ERROR "Archivo Lleno"
 - 5.6.3. **ESCRIBIR**(PRINC, pos, REG_EXIST)
 - 5.6.4. **CERRAR** PRINCSino
 - 5.6.5. RM (REG_EXIST.sig != -1)
 - 5.6.5.1. REG_ANTER = REG_EXIST.sig
 - 5.6.5.2. **LEER** (DESB, REG_EXIST.sig, REG_EXIST)
 - 5.6.6. REG_EXIST.sig=PILA_DESBORDE.pop()
 - 5.6.7. Si (REG_EXIST.sig== -1)
 - 5.6.7.1. **CERRAR** DESBD
 - 5.6.7.2. **RETORNAR** ERROR "Archivo Lleno"
 - 5.6.8. **Escribir**(DESB, REG_ANTER, REG_EXIST)
 // con nuevo puntero
6. TAM_ACT_D ++
7. **ESCRIBIR** (DESB, REG_EXIST.sig, NUEVO_REG)
8. **CERRAR** DESBD

Eliminación: Elimina un registro R con clave “clave” de un archivo Hash. Existen varios casos a considerar. Si el registro se encuentra en el Principal y éste a su vez se encuentra encadenado o no al Desborde. Si está de primero en el Desborde, en cuyo caso hay que actualizar el principal también. O si se encuentra simplemente en el Desborde, en cuyo caso habrá que actualizar los punteros correspondientes. Es importante recordar que debe devolverse los recursos a la entidad “policía” que se encarga de administrar los registros disponibles para nuevos ingresos.

Recuerde que, la función LEER del archivo, va a la posición **pos** y leer el registro en **REG_EXIST**.

```
1.pos = hash(k)
2.Si (pos<0 v pos > tam_max_p)
  2.1.      Retornar ERROR
3.LEER (PPAL, pos, REG_EXIST)
4.Si (REG_EXIST.disp == verdadero)
  4.1.      Cerrar(PPAL)
  4.2.      Retornar ERROR (“clave inexistente”)
5.Si (clave == REG_EXIST.clave)
  5.1.      Si (REG_EXIST.sig = -1)
    5.1.1.  REG_EXIS.sig == verdadero
    5.1.2.  ESCRIBIR (PPAL, pos, REG_EXIST)
    5.1.3.  CERRAR(PPAL)
    5.1.4.  tam_act_maestro --
    5.1.5.  Retornar (EXITO)
    SI NO
    5.1.6  Abrir (DBDE)
    5.1.7  LEER(DEBE, REG_EXIST.sig, TEMP)
    5.1.8  ESCRIBIR(PPAL, pos, TEMP)
    5.1.9  TEMP.disp = verdadero
    5.1.10 TEMP.sig = -1
    5.1.11 ESCRIBIR(DBDE, REG_EXIST.sig, TEMP)
    5.1.12 tam_act_desb --
    5.1.13 pila_desborde.push(REG_EXIST.prox)
    5.1.14 CERRAR(PPAL, DBDE)
    5.1.15 Retornar (EXITO)
```

```
SI NO (a lo mejor el registro se encuentra en el desborde)
5.2 Abrir (DBDE)
5.3 Actualizar_maestro = verdadero
5.4 Encontrado = falso
5.5 Anterior = pos
5.6 Actual = REG_EXIT.prox
5.7 continuar = verdadero
5.8 R.M. (continuar == verdadero ^ encontrado == falso)
  5.8.1. Anterior = actual
  5.8.2. Actual = REG_EXIST.prox
  5.8.3. LEER(desborde, actual, REG_EXIST)
  5.8.4. Si (S.prox = -1)
    5.8.4.1. continuar = falso
  5.8.5 Si (S.clave = clave)
    5.8.5.1. encontrado = verdadero
    SI NO
    5.8.5.2. actualizar_ppal = falso
5.9 Si (encontrado == falso)
  5.9.1. CERRAR (maestro, desborde)
  5.9.2. Retornar ERROR
5.10 Si (actualizar_principal == verdadero)
  5.10.1. LEER (maestro, pos, T)
  5.10.2. TEMP.sig = REG_EXIST.sig
  5.10.3. ESCRIBIR(PPAL, pos, TEMP)
  SI NO
  5.10.4 LEER(desborde, anterior, TEMP)
  5.10.5 TEMP.sig = REG_EXIST.sig
  5.10.6 ESCRIBIR(DBDE, Anterior, TEMP)
5.11 REG_EXIST.disp = verdadero
5.12 REG_EXIST.sig = -1
5.13 ESCRIBIR(DBDE, Actual, REG_EXIST)
5.14 tam_act_desb --
5.15 pila_desborde.push(Actual)
5.16 Cerrar (DBDE)
6. Cerrar (PPAL)
7. Retornar EXITO
```