

Redictado FOD 2016 – Primera Fecha de Evaluación Parcial – 15/11/16

1. Dadas las siguientes definiciones, codifique un procedimiento para dar de alta un registro en el archivo. Al realizar el alta se debe comprobar que no se inserten códigos duplicados. Las altas se realizan reutilizando registros borrados, si es que hay. Para reutilizar registros borrados, se mantiene una pila de posiciones borradas con cabecera en el registro cero.

Type

tProducto = **Packed Record**

Case activo: **Boolean of** {discriminante de registros para comprobar si son válidos o no (de encadenamiento de posiciones libres)}

True: (cod: **LongWord**;

nombre: **String**[23];

stock: **Word**;

precioVta: **Single**);

False: (proxLibre: **LongWord**); {el registro es de encadenamiento de espacio libre; si proxLibre=0 no hay más posiciones libres en el archivo}

end;

tCtlProd = **Record**

pp : **File of** tProducto; // archivo de productos

p: tProducto; // variable para leer productos

cab: tProducto // variable para mantener al registro 0 en memoria

end;

Function Alta (**var** a: tCtlProd; **const** nuevop: tProducto): **Boolean**;

{Devuelve **False** si encuentra código de producto duplicado, o **True** si se hizo el alta.

El archivo a.pp está abierto para lectura y escritura y no debe cerrarse.

Al abrirse a.pp se leyó el registro 0 en a.cab.}

...

2. Dado el estado inicial del siguiente árbol B+, con nodos con capacidad máxima para tres registros en todos los nodos, y mínima 1, especifique los estados sucesivos completos al realizarle las operaciones que se especifican. Además, indique los nodos que se leen y escriben para realizar cada operación.
Nota: Los nodos internos se parten 2-1, y los *underflows* se resuelven con hermanos derechos.

2: 0 (348) 4 (629) 1 (812) 3			
0: 187, 223 4	4: 348, 521 1	1: 629, 747, 801 3	3: 933 -1

Operaciones: +700, -933, -747.

3. Dado el siguiente estado inicial de un archivo directo estático de bloques con capacidad para dos registros e indicador de desborde, especifique los estados sucesivos completos al realizarle las operaciones que se especifican, e indique los bloques que se leen y escriben para realizar cada una de ellas. Los registros en saturación se organizan con dispersión doble, con función $f_2(x) = x \bmod 3 + 1$. Para las altas, se debe verificar que no haya claves duplicadas.

0: N |

1: N | 1146, 5431

2: N |

3: N | 2948

4: N | 3659, 814

Operaciones: +644, +8101, +5003.

Resolución

Type

tProducto = **Packed Record**

Case activo: **Boolean of** {discriminante de registros para comprobar si son válidos o no (de encadenamiento de posiciones libres)}

True: (cod: **LongWord**;

 nombre: **String**[23];

 stock: **Word**;

 precioVta: **Single**);

False: (proxLibre: **LongWord**); {el registro es de encadenamiento de espacio libre; si proxLibre=0 no hay más posiciones libres en el archivo}

end;

tCtlProd = **Record**

pp : **File of** tProducto; // archivo de productos

p: tProducto; // variable para leer productos

cab: tProducto // variable para mantener al registro 0 en memoria

end;

Function Alta (**var** a: tCtlProd; **const** nuevop: tProducto): **Boolean**;

{Devuelve **False** si encuentra código de producto duplicado, o **True** si se hizo el alta.

El archivo a.pp está abierto para lectura y escritura y no debe cerrarse.

Al abrirse a.pp se leyó el registro 0 en a.cab.}

var l: **longWord**; // posición libre

begin

// Búsqueda de código duplicado:

seek(a.pp, 1);

repeat

 read(a.pp, a.p)

until (a.p.activo **and** (a.p.cod = nuevop.cod)) **or** eof(a.pp);

// Si hay duplicado se sale devolviendo False:

if a.p.activo **and** (a.p.cod = nuevop.cod) **then begin** Alta:=False; **Exit end;**

// Alta normal:

if a.cab.proxLibre <> 0

then begin // hay alguna posición libre:

 l:=a.cab.proxLibre;

 seek(a.pp, l);

 read(a.pp, a.cab);

 seek(a.pp, 0);

 write(a.pp, a.cab)

end

else l:=FileSize(a.pp); // no hay posición libre

seek(a.pp, l);

write(a.pp, a.nuevop);

Alta:=True

end;

Posibles errores:

- Asignar y/o abrir-cerrar el archivo dentro de la función cuando se indica que el archivo ya está abierto y no debe cerrarse: si se abre el archivo con la intención de reposicionarse al comienzo, es un error leve (performance). Si abren y también cierran el error es grave (análisis).

- Posicionamiento en 0 en vez de en 1 para búsqueda de duplicado: error leve (performance). Si no se posicionan en 0 o 1 error muy grave (lógica).
- No buscar duplicado: error grave (análisis).
- Buscar duplicado sin verificar validez de registro: error grave (lógica).
- Encontrar duplicado y seguir leyendo hasta fin de archivo: error moderado (lógica y performance).
- No buscar posición libre y agregar al final: error muy grave (análisis).
- Inconsistencia en el manejo de posiciones libres: muy grave (lógica-algoritmia).

No de admite ningún error muy grave. Se tolera hasta un error grave. Todos los leves son tolerables.

2: 0 (348) 4 (629) 1 (812) 3			
0: 187, 223 4	4: 348, 521 1	1: 629, 747, 801 3	3: 933 -1

Operaciones: +700, -933, -747.

+700:

7: 2 (747) 6				
2: 0 (348) 4 (629) 1			6: 5 (812) 3	
0: 187, 223 4	4: 348, 521 1	1: 629, 700 5	5: 747, 801 3	3: 933 -1

Costo: [L2]L1 E1E5 E2E6 E7

-933:

7: 2 (747) 6				
2: 0 (348) 4 (629) 1			6: 5 (801) 3	
0: 187, 223 4	4: 348, 521 1	1: 629, 700 5	5: 747 3	3: 801 -1

Costo: [L7]L6L3 L5E5E3 E6

-747:

7: 2 (629) 6			
2: 0 (348) 4		6: 1(747) 5	
0: 187, 223 4	4: 348, 521 1	1: 629, 700 5	5: 801 -1

Costo: [L7]L6L5 L3E5 L2E2E6 E7

Libres: 3

Posibles errores:

- Al nodo raíz o lo leen en todas las operaciones o no lo leen en ninguna. Cambiar el criterio sería un error leve.
- Cambiar números de nodos: gravísimo.
- No partir nodos internos 2-1 como se especifica sino 1-2: grave.
- No partir las hojas 2-2 sino 3-1 o 1-3: grave.
- Fusionar 5 y 3 liberando al 3 en la baja de 933: moderado.
- Fusionar 2 y 6 liberando al 6 en la baja de 747: grave (si no fusionó en la baja previa no debe hacerlo ahora).
- Liberar al nodo 5 y no al 3 en la baja de 747: si actualiza correctamente el nodo 1 (cambiando referencia al siguiente) con las lecturas y escrituras pertinentes, grave, si no actualiza al 1, gravísimo.
- Error en lecturas y escrituras en todas las operaciones: gravísimo.

No de admite ningún error muy grave (gravísimo). Se tolera hasta un error grave.

0: N |
 1: N | 1146, 5431
 2: N |
 3: N | 2948
 4: N | 3659, 814

Operaciones: +644, +8101, +5003.

+644:

0: N |
 1: N | 1146, 5431
 2: N | 644
 3: N | 2948
 4: S | 3659, 814

$8101 \bmod 5 = 1$; $8101 \bmod 3 + 1 = 2$, $(1 + 2) \bmod 5 = 3$

Costo: L1E1 L3E3

+5003:

$644 \bmod 5 = 4$; $644 \bmod 3 + 1 = 3$, $(4 + 3) \bmod 5 = 2$

Costo: L4E4 L2E2

+8101:

0: N |
 1: S | 1146, 5431
 2: N | 644
 3: N | 2948, 8101
 4: S | 3659, 814

0: N |
 1: S | 1146, 5431
 2: N | 644, 5003
 3: S | 2948, 8101
 4: S | 3659, 814

$5003 \bmod 5 = 3$; $5003 \bmod 3 + 1 = 3$, $(3 + 3) \bmod 5 = 1$; $(1 + 3) \bmod 5 = 4$; $(4 + 3) \bmod 5 = 2$

Costo: L3E3 L1 L4 L2E2

Posibles errores:

- Leer secuencialmente los bloques del archivo hasta llegar al de destino: gravísimo.
- Reflejar cambios correctamente en indicadores de desborde, pero no indicar escrituras correspondientes: grave.
- No usar indicadores de desborde: si no los dibujan y resuelven correctamente marcando registros borrados e indicando bien lecturas y escrituras: grave.
- Error en lecturas y escrituras en todas las operaciones: gravísimo.

No de admite ningún error muy grave (gravísimo). Se tolera hasta un error grave.