Interrogation écrite n°1 (Sujet B) : « Interclassement de deux fichiers TOF »

Soient F1 et F2 deux fichiers **TOF** ayant la même structure et le même ordre (croissant). En disposant de buffers en mémoire centrale (MC), on souhaite réaliser l'opération d'interclassement de ces deux fichiers contenant le même type d'enregistrement en créant un fichier F3 de type TOF contenant tous les enregistrements de F1 et F2.

a) Quel est le nombre minimal de buffers en MC nécessaires pour réaliser cette opération ? (1 pt)

b) Donnez la déclaration des structures des fichiers. (2 pts)

c) Donnez un algorithme qui réalise l'opération d'interclassement de F1 et F2 qui doivent être parcourus simultanément en un seul parcours et en utilisant un nombre minimal de buffers. (10 pts)

d) Donnez le coût de cette opération dans le meilleur cas et dans le pire des cas. (3 pts)

e) Comment voyez-vous l'interclassement multiple de N fichiers TOF ? Donnez un schéma représentant la MC et la MS. Quel est le nombre minimal de buffers nécessaires ? (4 pts)

---

a) Nombre minimal de buffers : 3 buffers.

b) Déclarations :

Fusion de 2 fichiers ordonnés (TOF)

On parcours les 2 fichiers en parallèle dans 2 buffers (buf1 et buf2) et on rempli le buffer (buf3) du 3e fichier en ordre croissant.

Les déclarations sont celles utilisées dans les fichier TOF standard.

c) Algorithme d'interclassement de F1 et F2 :

```
Fusion (nom1,nom2, nom3: chaine)
var
   F1 : Fichier de Tbloc Buffer buf1 Entete( entier, entier);
   F2 : Fichier de Tbloc Buffer buf2 Entete( entier, entier);
   F3 : Fichier de Tbloc Buffer buf3 Entete( entier, entier);
   i1, i2, i3 :  entier;
   j1, j2, j3 :  entier;
   continu : booleen;
   e, e1, e2 : Tenreg;

   buf : Tbloc;
   i, j, indic : entier;

Debut
        ouvrir(F1, nom1, 'A' );
        ouvrir(F2, nom2, 'A' );
        ouvrir(F3, nom3, 'N' );

        i1 ← 1; i2 ← 1; i3 ← 1;      // les num de blocs de F1, F2 et F3
        j1 ← 1; j2 ← 1; j3 ← 1;      // les num d'enreg dans buf1, buf2 et buf3

        LireDir(F1, 1, buf1)
        LireDir(F2, 1, buf2)

        continu ← vrai
```

```
TQ continu                    // tant que non fin de fichier dans F1 et F2 faire

        SI ( j1 <= buf1.NB  et  j2 <= buf2.NB )
                // choisir le plus petit enreg, dans buf1 et buf2
                e1 ← buf1.tab[j1];
                e2 ← buf2.tab[j2]
                SI ( e1.cle <= e2.cle )
                        e ← e1;  j1 ← j1 + 1;
                SINON
                        e ← e2;  j2 ← j2 + 1;
                FSI

                // et le mettre dans buf3
                SI ( j3 <= b )
                        buf3.tab[j3] ← e;  j3 ← j3 + 1
                SINON
                        buf3.NB ← j3 – 1;
                        EcrireDir(F3, i3, buf3 );
                        i3 ← i3 + 1;
                        buf3.tab[1] ← e;
                        J3 ← 2;
                FSI
```

```
        SINON           // c-a-d : non ( j1 <= buf1.NB  et  j2 <= buf2.NB )
                // si tous les enreg d'un des blocs (buf1 ou buf2) ont été traités, passer au prochain
                SI ( j1 > buf1.NB )
                        SI ( i1 < entete(F1, 1) )
                                i1 ← i1 + 1;
                                LireDir( F1, i1, buf1 )
                                j1 ← 1
                        SINON // ( donc i1 >= entete(F1, 1) )
                                continu ← faux
                                i ← i2;   // pour la suite du TQ
                                j ← j2;
                                N ← entete(F2,1)
                                buf ← buf2
                                Indic ← 2
                        FSI // ( i1 < entete(F1, 1) )

                SINON // c-a-d ( j2 > buf2.NB )
                        SI ( i2 < entete(F2, 1) )
                                i2 ← i2 + 1;
                                LireDir( F2, i2, buf2 );
                                j2 ← 1
                        SINON // ( donc i2 >= entete(F2, 1) )
                                continu ← faux;
                                i ← i1;   // pour la suite du TQ
                                j ← j1;
                                N ← entete(F1,1);
                                buf ← buf1;
                                Indic ← 1;
                        FSI // ( i2 < entete(F2, 1) )

                FSI // ( j1 > buf1.NB )

        FSI  // ( j1 <= buf1.NB  et  j2 <= buf2.NB )

FTQ
```

```
// continuer à recopier les enregistrement d'un seul fichier (i,j,buf) dans F3
continu ← vrai;
TQ continu                    // tant que non fin de fichier dans F1 ou F2 faire
        SI ( j <= buf.NB )
                SI ( j3 <= b )
                        buf3.tab[j3] ← buf.tab[j];  j3 ← j3 + 1
                SINON
                        buf3.NB ← j3 – 1;
                        EcrireDir(F3, i3, buf3 );
                        i3 ← i3 + 1;
                        buf3.tab[1] ← buf.tab[j];
                        J3 ← 2;
                FSI // ( j3 <= b )
                j ← j + 1

        SINON // c-a-d non ( j <= buf.NB )
                SI ( i <= N )
                        i ← i + 1;
                        SI Indic = 1
                                LireDir( F1, i, buf )
                        SINON
                                LireDir( F2, i, buf )
                        FSI
                        j ← 1
                SINON
                        continu ← faux
                FSI

        FSI // ( j <= buf.NB )

FTQ
```

```
        // Le dernier buffer (buf3) n'a pas encore été ecrit sur disque ...
        buf3.NB ← j3 – 1
        EcrireDir( F3 , i3, buf3 )
        Aff-entete( F3, 1, i3)                          // le nombre de blocs dand F3
        Aff-entete( F3, 2, entete(F1,1) + entete(F2,1) )  // le nombre d'enregistrements dans F3

        Fermer( F1 )
        Fermer( F2 )
        Fermer( F3 )

Fin
```

d) Coût de l'interclassement :

➔ Meilleur cas : 2 lect + 1 écriture

➔ Pire des cas : (2*Entete(F1,1) ou 2*Entete(F2,1)) + (|Entete(F1,1) – Entete(F2,1)|) lectures.
Entete(F3,1) écriture(s)

e) Fusion multiple de N fichiers avec N+1 buffers en MC.

Les variables suivantes sont supposées être déclarées globales :

• F : Tableau de variables de type Fichier de taille n+1 (organisation TOF). La 1ere caractéristique (Entete(F,1)) désigne le nombre de blocs utilisés.

• buf : Tableau de buffers de taille n+1. Chaque buffer contient un tableau (tab) d'enreg de taille b et un entier (NB).

• NumBloc : tableau d'entier de taille n+1 représentant les numéros de blocs courant pour chaque fichier.

• NumEnr : tableau d'entier de taille n+1 représentant les numéros d'enregistrements courants dans chaque buffers en MC.

• Fin : tableau de booléens de taille n indiquant pour chaque fichier en entrée, si on a atteint la fin de fichier ou non.