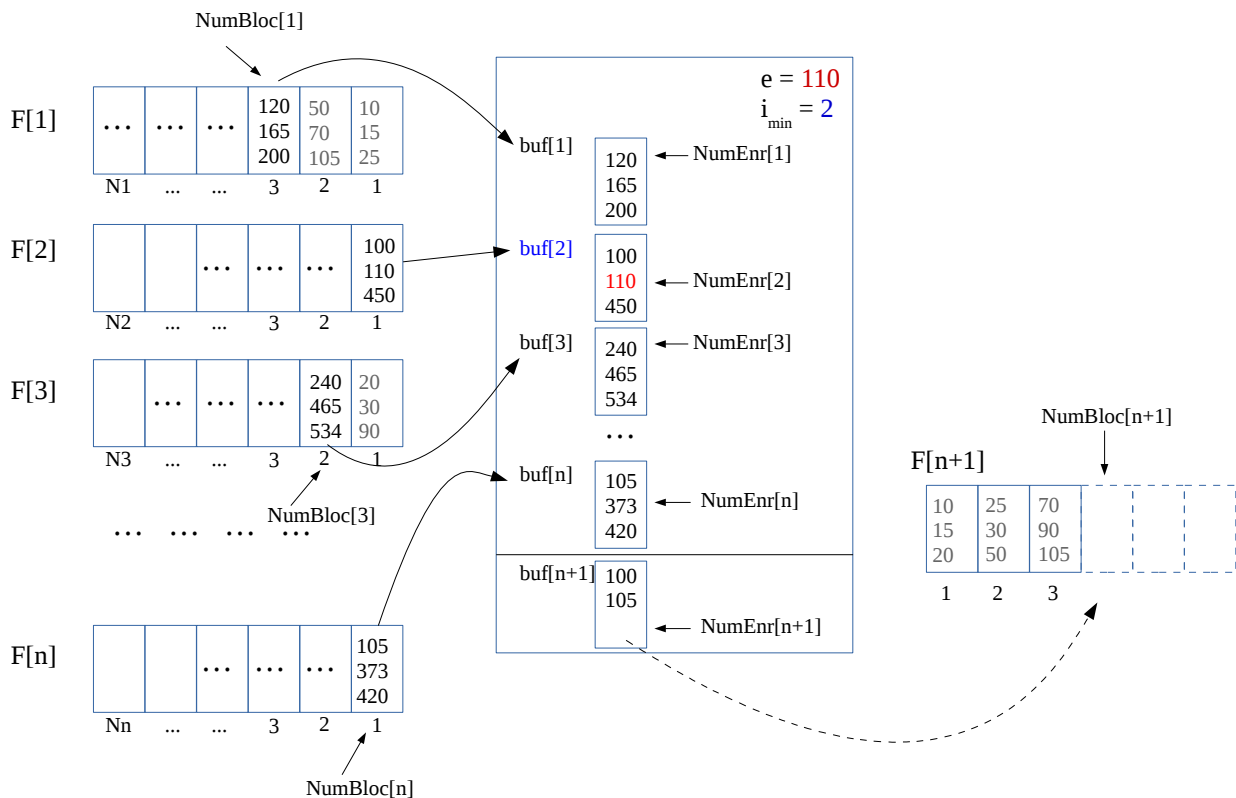


Fusion multiple de n fichiers avec n+1 buffers en MC

Les variables suivantes sont supposées être déclarées globales :

- **F** : Tableau de taille n+1, de variables de type Fichier (organisation TOF). La 1ere caractéristique (Entete(F,1)) désigne le nombre de blocs utilisés.
- **buf** : Tableau de taille n+1 de buffers. Chaque buffer contient un tableau (tab) d'enreg de taille b et un entier (NB).
- **NumBloc** : tableau de taille n+1 d'entier représentant les numéros de blocs courants pour chaque fichier.
- **NumEnr** : tableau de taille n+1 d'entier représentant les numéros d'enregistrements courants dans chaque buffers en MC.
- **Fin** : tableau de taille n de booléens indiquant pour chaque fichier en entrée, si on a atteint la fin de fichier ou non.



Procédure Fusion (n:entier, nomf:tableau de chaînes, Resultat:chaîne)

// nomf : tableau de chaînes de caractères de taille n, représentant les noms des fichiers à fusionner.

// Resultat : chaîne de caractères représentant le nom du nouveau fichier résultat de la fusion.

Var i : entier ; e : Tenreg ;

Début *// Fusion*

/ Initialisations ... */*

Pour i=1, n

Ouvrir(F[i], nomf[i], 'A') ;

NumBloc[i] ← 1 ;

NumEnr[i] ← 1 ;

SI (**Entete**(F[i], 1) <> -1) *// Fichier non vide*

LireDir(F[i], 1, buf[i]) ;

Fin[i] ← FAUX ;

SINON

Fin[i] ← VRAI ;

FSI ;

FP ;

Ouvrir(F[n+1], Resultat, 'N') ;

NumBloc[n+1] ← 1 ;

NumEnr[n+1] ← 1 ;

/ Boucle Principale ... */*

TQ (PlusPetitElem(e))

// retourne vrai s'il reste au moins un enreg non encore considéré et dans ce cas, e désigne le

// plus petit enreg parmi ceux présents en MC.

// Insertion de e dans le buffer de sortie buf[n+1] ...

SI (NumEnr[n+1] ≤ b)

buf[n+1].tab[NumEnr[n+1]] ← e ;

NumEnr[n+1] ← NumEnr[n+1] + 1 ;

SINON

buf[n+1].NB ← b ;

EcrireDir(F[n+1], NumBloc[n+1], buf[n+1]) ;

NumBloc[n+1] ← NumBloc[n+1] + 1 ;

buf[n+1].tab[1] ← e ;

NumEnr[n+1] ← 2 ;

FSI ;

FTQ ;

buf[n+1].NB ← NumEnr[n+1] - 1 ; *// Dernière écriture dans le fichier de sortie ...*

EcrireDir(F[n+1], NumBloc[n+1], buf[n+1]) ;

Aff-Entete(F[n+1], 1, NumBloc[n+1]) ; *// nombre de blocs utilisés pour le fichier de sortie*

// Fermeture des fichiers ...

Pour i=1,n+1

Fermer(F[i]) ;

FP

Fin *// Fusion.*

*/*******

Fonction PlusPetitElem(var e:Tenreg) : Booléen

// Rechercher le plus petit enregistrements non encore considéré et avancer dans le fichier associé.

Var i : entier ;

Début *// PlusPetitElem*

// Localiser le 1^{er} fichier ayant encore des enregistrements non lus ...

 i ← 1 ;

 trouv ← FAUX ;

 TQ (Non trouv ET i ≤ n)

 SI (Non Fin[i]) trouv ← VRAI SINON i ← i+1 FSI

 FTQ ;

 SI (Non trouv)

 Retourner FAUX ; *// tous les enregistrements ont déjà été considérés.*

 SINON

// Trouver le minimum parmi ceux qui restent ...

 i_{min} ← i ;

 Pour j=i+1, n

 SI (Non Fin[j]) *// si le fichier F[j] a encore des enreg non lus ...*

 SI (buf[j].tab[NumEnr[j]] < buf[i_{min}].tab[NumEnr[i_{min}]])

 i_{min} ← j ;

 FSI

 FSI

 FP ;

 e ← buf[i_{min}].tab[NumEnr[i_{min}]] ; *// le plus petit enregistrement.*

// Avancer dans le fichier F[i_{min}] ...

 NumEnr[i_{min}] ← NumEnr[i_{min}] + 1 ;

 SI (NumEnr[i_{min}] > buf[i_{min}].NB)

 NumBloc[i_{min}] ← NumBloc[i_{min}] + 1 ;

 SI (NumBloc[i_{min}] ≤ Entete(F[i_{min}], 1))

 LireDir(F[i_{min}], NumBloc[i_{min}], buf[i_{min}]) ;

 NumEnr[i_{min}] ← 1 ;

 SINON

 Fin[i_{min}] ← VRAI ;

 FSI ;

 FSI

 Retourner VRAI ;

 FSI ;

Fin *// PlusPetitElem*