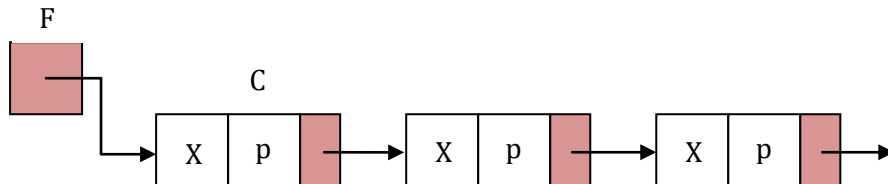


Dans tous les fichiers, le type *file* a été changé par le type *fap*, pour éviter la confusion avec le type *fichier*. Quelques fonctions ont également été ajoutées dans *fap.h*.

ALGORITHME DU PROGRAMME



TYPE :

F : adresse de tête de File, type **fap**

C : maillon de la file, de type **fap**, de cases :

- **Elem**, contenant un élément X de type **entier**
- **Priorite**, contenant une priorité p de type **entier**
- **Suivant**, contenant une adresse vers le maillon suivant, de type **fap**

PRIMITIVES :

Initialiser() : Renvoie une file vide.

EstVide (fap F) : renvoie 1 si la file F est vide, 0 sinon.

EstPleine (fap F, int nbmax) : Renvoie 1 si la file F est pleine, cad que le nombre d'éléments dans cette file a atteint nbmax (la taille max des files est décidée dans test_fap.c et simul_fap.c). Renvoie 0 sinon.

Afficher (fap F) : Affiche la file F sous cette forme : [X , p] [X , p]

Inserer (fap F, int x, int p) : Renvoie une file où le maillon [x , p] a été inséré à F. Si ce maillon a la plus grande priorité, elle est ajoutée en tête de file. Sinon, on place le maillon après le dernier ayant une plus grande priorité qu'elle. **Ici, la file est toujours triée par ordre de priorité décroissante** (Celui a qui a la plus grande priorité p est le plus prioritaire).

Extraire (fap F, int *x, int *p) : Extrait l'élément en tête de file.

Detruire(fap F) : Détruit la file. Si celle-ci n'est pas vide, détruit d'abord tous les éléments de celle-ci.

TESTS (TEST_FAP)

Test d'insertion : (on ne prête pas attention à la valeur de X ici)

- Insertion dans cet ordre : [X , 1] [X , 2] [X , 3]
Renvoie : [X , 3] [X , 2] [X , 1]
Conclusion : Effectue correctement une insertion en tête.

- Insertion dans cet ordre : [X, 1] [X, 3] [X, 2]
Renvoi : [X, 3] [X, 2] [X, 1]
Conclusion : Insère [X, 2] au bon endroit (recherche et insertion en milieu de file)
- Insertion jusqu'à atteindre NBMAX (=2)
Renvoi : « Erreur : la fap est pleine ! »
Conclusion : Détecte correctement une file pleine.

Test d'extraction :

- Extraction de : [X, 3] [X, 2] [X, 1]
Renvoi : [X, 3]
Etat de la file : [X, 2] [X, 1]
Conclusion : Effectue correctement une extraction classique.
- Extraction de : NULL
Renvoi : « Erreur : La fap est vide ! »
Conclusion : Détecte correctement une file vide.

SIMULATION

Le programme simul_fap génère aléatoirement un couple [X, p] qui sera ensuite inséré dans la File F. nb_evenement correspond ainsi au nombre d'éléments dans F. Ensuite, la file est affichée, puis on extrait un par un les éléments de la file, jusqu'à que celle-ci soit vide.

Le programme proposé ici est simple, car je n'ai pas bien compris vos attentes...

Avec un peu plus de temps, je pourrais simuler une file d'attente dynamique, où par exemple, le 1^{er} élément de la file serait extrait toutes les 2 secondes, pendant qu'un élément serait ajouté toutes les 1 secondes, avec la possibilité de changer la priorité d'un élément déjà présent dans la file d'attente.