 Ecole Supérieure Privée d'Ingénierie et de Technologies	<div style="text-align: center;">EXAMEN PRATIQUE</div> Semestre : 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/> Session : Principale <input type="checkbox"/> Rattrapage <input checked="" type="checkbox"/>
Module : Programmation Procédurale 2 Enseignant(s) : Equipe Programmation C Classes: 1A9, 1A10, 1A11, 1A12	
Documents autorisés : OUI <input checked="" type="checkbox"/> NON <input type="checkbox"/> Nombre de pages : 02 Connexion autorisée : OUI <input type="checkbox"/> NON <input checked="" type="checkbox"/> Date : 20/06/2017 Heure : 8h 30 Durée : 1h.30	

Enoncé

On vous propose d'implémenter une solution informatique à base de l'architecture réseau **Diffserf** permettant d'assurer une certaine qualité de service lors du routage du trafic tout en différenciant les services des données. La gestion des priorités du trafic s'effectue alors selon les différents types de flux (VOIP, multimédia, etc.).

Pour ce faire, nous commençons par définir un paquet IP (version IPV4). Un paquet est une structure contenant les champs suivants :

- **Donnée** (chaîne qui contient au maximum 1542 caractères)
- **Entête** structure formée par les champs suivants :
 - Un identifiant unique (entier).
 - Une adresse IP source (chaîne de 6 caractères)
 - Une adresse IP destination (chaîne de 6 caractères)
 - Une priorité (3 : priorité haute, 2 : priorité moyenne et 1 : priorité faible).
 - Un champ TTL (Time to Live : Durée de vie strictement >0).
 - Un champ protocole (entier > 0)

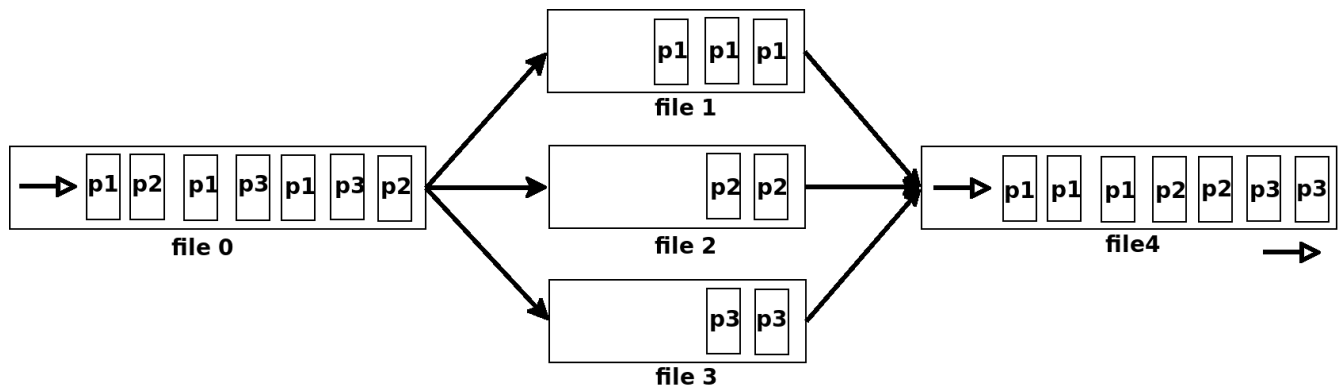


Figure 1 : Différenciation des services par priorité

Comme illustré dans la Figure 1, les données sont acheminées au début en désordre dans une file « file 0 » (modèle FIFO). Ensuite, chaque paquet est classifié dans une parmi 3 files (« file 1 », « file 2 » et « file 3 ») selon la priorité (paquet à faible, moyenne ou haute priorité).

Ensuite, tous les paquets sont redirigés vers une dernière file (« file 4 »). Pour ceci, on procède à consulter puis défiler les paquets de la file la plus prioritaire (« file 3 »), ensuite la file moyennement prioritaire (« file 2 ») et enfin la file la moins prioritaire (« file 1 »). (Voir l'illustration de cela sur la Figure 1)

Lors de cette dernière redirection (vers la file « file 4 ») et juste après avoir consulté un paquet, le champ TTL du paquet en question doit être mis à jour en le décrémentant de 1 afin d'éviter le phénomène de paquets fantôme.

Ensuite, les paquets de la dernière file « file 4 » ayant un TTL > 0 sont redirigés (défilés) vers un arbre binaire de recherche selon la valeur du protocole en question.

Enfin, à partir de l'arbre binaire de recherche, les paquets de transaction VISA dont la valeur du protocole est égale à 70 sont sauvegardés un fichier texte intitulé « HackVisa.txt ».

Travail demandé

- Déclarer les structures et les types nécessaires et implémenter la fonction **main ()** permettant d'assurer les opérations suivantes :
 - La saisie de 3 paquets de priorités différentes
 - Enfiler ces paquets dans une première file **f0**.
 - La classification de ces paquets à partir de la file **f0** dans 3 autres files **f1**, **f2** et **f3**.
 - La redirection du contenu de chaque file (selon l'ordre de priorité) vers une dernière file **f4**.
 - La redirection des paquets ayant un TTL > 0 de la file **f4** vers un arbre binaire de recherche **abr**
 - La sauvegarde des paquets de type VISA (protocole=70) dans un fichier « hackVisa.txt ».
- Implémenter une fonction qui permet de saisir un paquet.

*void saisirPaquet (paquet *paq)*

3. Implémenter une fonction qui permet de consulter une file.

paquet consulterFile (file *f)

4. Implémenter une fonction qui permet d'enfiler un paquet dans une file

void enfiler (file *f, paquet p)

5. Implémenter une fonction qui permet de défiler une file.

void defiler (file *f)

6. Implémenter une fonction qui permet de consulter d'abord la file **f0**, défiler ensuite le paquet d'entête et rediriger enfin ce paquet dans la file adéquate (**f1**, **f2** ou **f3**) selon le champ priorité.

void classifiePaquets (file *f0, file *f1, file *f2, file *f3)

7. Implémenter une fonction qui permet de vider les paquets des 3 files dans une file **f4** selon l'ordre suivant : **f3**, **f2** et **f1** (comme illustré dans la Figure 1)

void redirigerFilePriorite (file *f1, file *f2, file *f3, file *f4)

N.B. : Le champ TTL de chaque paquet doit être décrémenté avant la redirection vers la file **f4**

8. Implémenter une fonction qui permet d'insérer un paquet dans un arbre selon la valeur du protocole.

arbre insererPaquetArbre (arbre abr, paquet paq)

9. Implémenter une fonction qui permet de rediriger les paquets ayant un TTL > 0 à partir de la file **f4** vers un arbre binaire de recherche **abr** selon la valeur du protocole.

arbre file2Arbre (file f4, arbre abr)

N.B. : L'arbre **abr** ne doit contenir que les paquets à TTL > 0

10. Implémenter une fonction permettant de sauvegarder les paquets de type VISA dans un fichier texte.

void sauvPaquetVisa (arbre abr, char *nomfich)

N.B. : Les paquets de type VISA sont les paquets dont la valeur du protocole est égale à 70.

Grille d'évaluation

Nom :Prénom:

Classe : 1A...

Note :/20

Nom fonction	Algorithmique	
Déclaration des structures et des types		1
Saisir		0.5
Consulter		1
Enfiler		1
Défiler		1
Classifier		2
RedirigerPaquet		2
Insérer		1
File2Arbre		1
SauvPaquetVisa		2
Total		12.5

La Fonction main	Exécution ⁽¹⁾		Appel de fonction		Total	
<i>Menu</i>		0.5		0.5		1
<i>Saisir 3 paquets avec priorités différentes</i>		0.5		0.5		1
<i>Enfiler paquets</i>		0.5		1		1.5
<i>Classifier</i>		0.5		0.5		1
<i>Rediriger</i>		0.5		0.5		1
<i>File2Arbre</i>		0.5		0.5		1
<i>SauvPaquetVisa</i>		0.5		0.5		1
Total						7.5

Légende: Note 1 : la note d'exécution est binaire