

Concours d'accès au Doctorat LMD en Informatique (2022/2023)

Épreuve commune : Algorithmique

(durée : 1h30 mn)

Exercice 1 (8p)

On souhaite gérer les notes des étudiants d'une section. Chaque étudiant possède les informations suivantes :

Un identifiant qui est son numéro d'étudiant appelé *clé* (chaîne de 10 caractères),

Un *nom* (chaîne de 20 caractères) et un *prénom* (chaîne de 20 caractères),

Une moyenne générale *moy* (réel),

Un ensemble de notes avec un coefficient *coeff* (entier) pour chaque *note* (réel).

- 1- Sachant que chaque étudiant peut avoir 0 ou plusieurs notes, proposer une liste chaînée *L-etud* pour sauvegarder les informations ci-dessus concernant l'ensemble des étudiants. (2p)

- 2- Donner les déclarations associées à cette liste. (2p)

- 3- En supposant que tous les champs de la liste des étudiants sont remplis ainsi que les notes des étudiants et tous les coefficients sont remplis sauf le champ *moy*, écrire l'algorithme d'une procédure *MoyEtudiants* qui parcourt la liste des étudiants, calcule et ensuite met à jour le champ *moy* de chaque étudiant. La procédure *MoyEtudiants* prend en paramètre la tête de la liste des étudiants *L-etud*. (4p)

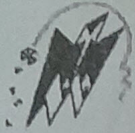
Exercice 2 (12p)

On définit le type arbre binaire de recherche (abr) comme suit :

```
Type abr = ↑ noeud
    noeud == record
        clé : entier;
        g, d : abr;
    end
```

On considère alors l'opération de partition d'un abr A selon la clé X comme une opération qui détruit l'arbre initial et renvoie deux arbres abr GA et DR, tels que GA contient les clés de A qui sont inférieures ou égales à la clé X, et DR contient les clés de A qui sont strictement supérieures à la clé X.

On ne doit pas parcourir tous les nœuds de A mais seulement les nœuds situés sur le chemin de recherche de la clé X dans A. On va devoir alors suivre ce chemin en allant de la racine jusqu'à la feuille. À chaque nœud visité, on va tester si sa clé est inférieure (resp. supérieure) à X, auquel cas elle ira dans GA (resp. DR) accompagnée de son propre sous-arbre gauche (resp. droit).



Concours d'accès au Doctorat LMD en Informatique (2022/2023)

Épreuve commune : Algorithmique

(durée : 1h30 mn)

Un exemple de partition d'un abr A en GA et DR selon la clé $X = 24$ est illustré sur la Figure ci-après. Sur l'arbre A, on montre en pointillés le chemin suivi et donc les liens supprimés. Sur les arbres GA et DR obtenus, on montre en pointillés les liens créés.

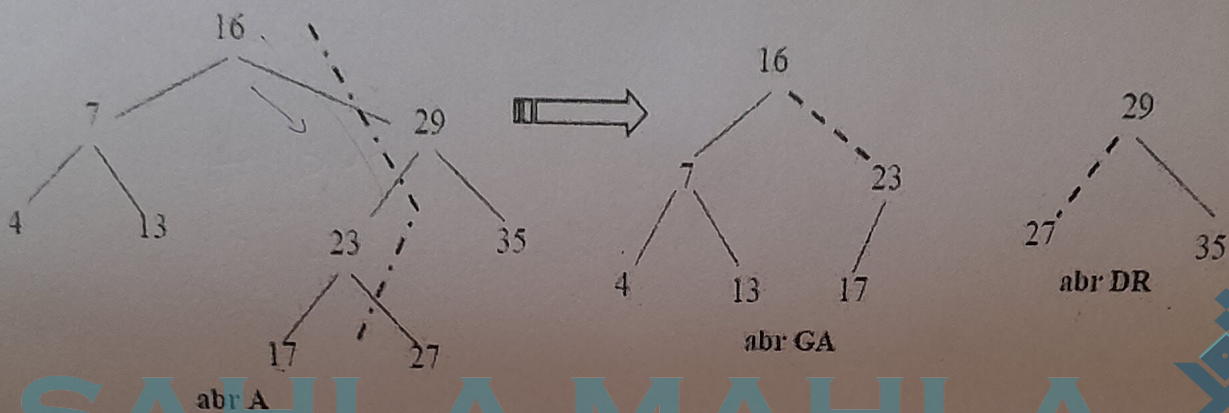


Figure: Partitionnement de A selon $X = 24$

- 1- Écrire l'algorithme d'une procédure récursive **Partition** (X : clé ; var A, GA, DR : abr) qui permet de réaliser le partitionnement de A en deux abr GA et DR suivant la clé X. (6p)
- 2- Montrer comment peut-on insérer une clé X à la racine d'un abr A tout en conservant sa propriété d'abr moyennant la procédure précédente. (1.5p)
- 3- Écrire alors l'algorithme **Ajout-Racine** (X : Clé; A : abr) réalisant l'insertion d'une clé X à la racine d'un abr A. (4.5p)



Concours d'accès au Doctorat LMD en Informatique (2022/2023)

Epreuve : Systèmes et Réseaux

(durée : 2h00 mn)

Spécialité : Systèmes Informatiques

Exercice n° 1 : (08 pts) : On souhaite effectuer le calcul de la somme des éléments d'un tableau d'entiers. Pour ce faire, le processus père crée n processus fils pour calculer les sommes des n parties égales du tableau. Plus exactement, chacun des processus fils, exécute un même code qui consiste à afficher son PID, l'indice début et l'indice fin de la partie du tableau traitée (exemple : 0..19). Ensuite il calcule la somme des éléments de cette partie du tableau, affiche cette somme et puis il l'envoie au processus père dans un tube (pipe).

Le processus père attend ses fils, récupère les sommes partielles dans le tube et les additionne. Il affiche ensuite son PID et la somme totale.

Ecrire le programme correspondant en utilisant le langage C et en supposant que :

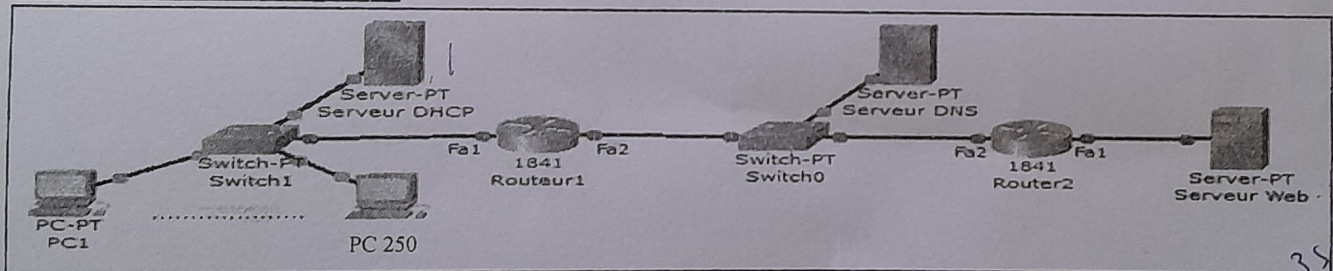
- La taille du tableau est égale à 100. Les données du tableau sont générées automatiquement à l'aide de la fonction `rand()`, exemple : `(rand()%13)+1`.
- Le nombre de processus fils est égal à 5.

Exercice n° 2 : (06 pts) : Pour mettre au point un schéma de communication entre processus, on considère une application composée d'un processus père et d'un processus fils qui communiquent à l'aide de signaux selon le protocole suivant :

À chaque fois qu'il reçoit un signal `SIGUSR1`, le processus père affiche la chaîne « Ping ! », attend une seconde, puis envoie un signal `SIGUSR2` au processus fils.

De même, à chaque fois qu'il reçoit un signal `SIGUSR2`, le processus fils affiche la chaîne « Pong ! », attend une seconde, puis envoie un signal `SIGUSR1` au processus père. Dès qu'il a mis en place le gestionnaire de signal, le fils envoie un premier signal `SIGUSR1` au père. Donnez les instructions (en langage C) nécessaires pour cette application.

Exercice n° 3 (06 pts) : Soit le réseau suivant :



Réseau de départ : 192.168.4.0/22

1) A l'aide de la technique du masque variable (VLSM), implémentez un plan d'adressage pour le réseau précédant :

	Adresse ip		MSR	Passerelle par défaut
Serveur Web				
Serveur DNS				
Routeur1 (Fa1)				
Routeur1 (Fa2)				
Routeur2 (Fa1)				
Routeur2 (Fa2)				
Serveur DHCP				
	Adresse début	Adresse fin	M SR	Option DHCP (passerelle)
Etendue DHCP				

- 2) Donner les tables de routage des routeurs.
- 3) Donner les étapes de configuration nécessaires pour qu'un client DHCP (en dehors du sous réseau du serveur DHCP) puisse obtenir une adresse à partir de l'unique serveur DHCP du réseau. Cette solution est-elle tolérante aux pannes ? justifier.

SAHLA MAHLA

المصدر الاول للطالب الجزائري

