Université de Gabès
Faculté des Sciences de Gabès
Département Informatique

A. U: 2023-2024

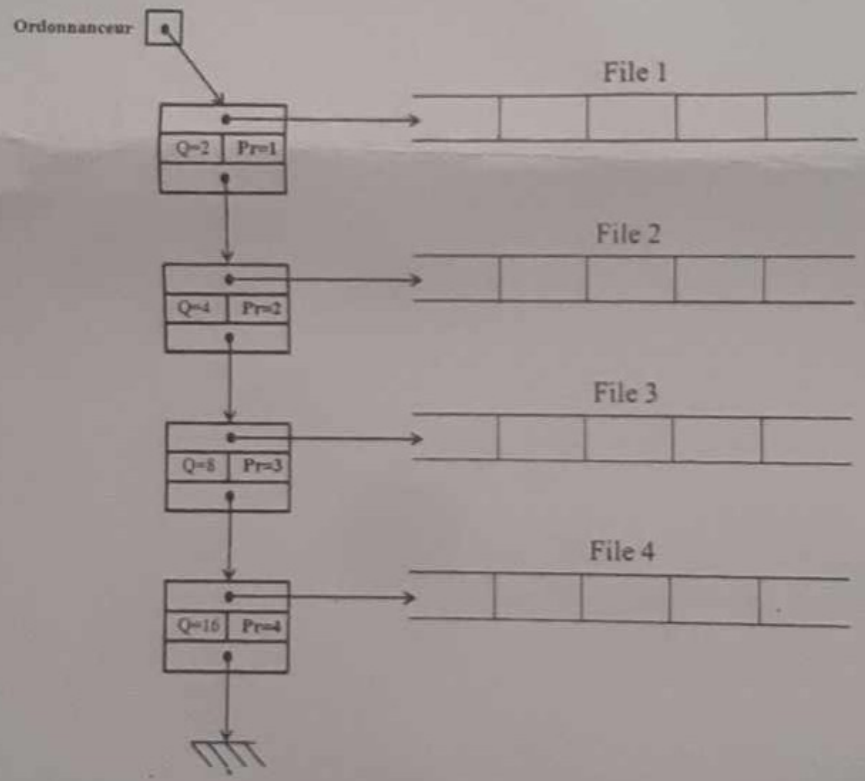
EXAMEN

Sections: LGLSI1, LIRIS1 **Épreuve d':** Algorithmique, structures de données et complexité

Nature de l'épreuve:	DC☐	DS☐	EF☑	Documents:	autorisés	☐	non autorisés ☑
Date de l'épreuve:	20/06/2024			Session:	principale	☐	contrôle ☑
Durée de l'épreuve:	1H30			Enseignant:	Fethi Mguis		

Exercise 1 : (10 pts)

On considère un ordonnanceur Multi-niveaux qui utilise une liste simplement chaînées, comme la montre la figure suivante, pour organiser les processus prêts :



- Un processus a un identifiant(PID), une priorité(Pr) et une durée d'exécution(DE). Tout les champs sont de type entier.
- A chaque niveau de priorité correspond une file de processus, dans laquelle seront ajoutés les processus prêts ayant cette priorité.
- A chaque niveau de priorité correspond un quantum Q. En effet, le processeur sera alloué à chaque processus d'une telle file pendant une durée égale à Q. Ensuite, le processus quitte le système s'il a terminé son exécution, sinon il va rejoindre la file suivante.
- Un nouveau processus doit rejoindre la file correspondante à sa priorité.
- L'ordonnanceur utilise 4 niveaux de priorités(1-4). Donc lorsque un processus atteint le dernier niveau de priorité, il réside dans ce niveau jusqu'à la fin de son exécution.

1. Définir les types nécessaires pour le fonctionnement de l'ordonnanceur.

2. Ecrire la procédure InitialiserOrdonnanceur(O :*Ordonnanceur) permettant de choisir la valeur du quantum de chaque niveau et initialiser la file correspondante à vide.
3. Ecrire la procédure ArrivéeProcessus(O :Ordonnanceur, P : Processus) qui s'exécute à l'arrivée d'un nouveau processus P.
4. Ecrire la procédure FinQuantumProcessus(O :Ordonnanceur, P : Processus) qui s'exécute lorsque le processus P libère le processeur. Cette procédure permet d'allouer le processeur au processus suivant, incrémenter la priorité du processus qui a été en exécution et l'ajouter à la file suivante lorsque c'est nécessaire.
5. Ecrire la procédure AfficherEtatSystem(O :Ordonnanceur) permettant d'afficher les processus prêts.

Exercise 2 : (4 pts)

On considère une pile contenant n entiers. On souhaite trier les éléments de la pile dans l'ordre décroissant(c'est-à-dire le plus grand élément soit placé en sommet de la pile) et ce en utilisant uniquement une autre pile intermédiaire et une variable de type entier.

1. Définir les types nécessaires pour représenter une pile d'entiers.
2. Ecrire la procédure **Tri(P : *Pile)**.
   N.B : Vous pouvez appeler les procédures/fonctions Init, EstVide, Empiler, Dépiler et Sommet sans les ré-implémenter.

Exercise 3 : (6 pts)

A est un arbre binaire de recherche dont les données sont des entiers.

1. Dessiner A après l'insertion des nombres 110, 25, 32, 140, 115, 15, 20, 40, 150, 210
2. Quelle est la hauteur de A ?
3. Afficher les éléments de A selon les 3 parcours en profondeur gauche(Préfixé, Infixé et Postfixé).
4. Afficher les éléments de A selon le parcours en largeur gauche.
5. Que devient A si on rajoute 125 ?
6. Que devient A si on supprime 25 ?

Bon Travail