

### Exercice 1 ; (7pts)

Considérant quatre travaux A, 2 ; B, 5 ; C, 3 et D, 1 avec leurs temps d'exécution soumis à un calculateur dans cet ordre, mais quasi simultanément.

Q1 : Donnez les diagrammes de : FIFO ; SJF ; SRTF ; round robin avec (Q = 2).

Q2 : Calculez pour chaque job le temps d'attente et le temps de résidence.

Q3 : Calculez les temps moyens et discutez les résultats.

### Exercice N° 2 : (7pts)

Considérez un système disposant de 16 MO de mémoire physique réservée aux processus utilisateur. La mémoire est composée de cases de taille 4 KO. L'espace logique d'un processus est composé de trois segments (le segment de code, le segment de données et le segment de pile). Chaque segment est composé d'une ou plusieurs pages. Lorsqu'un processus demande à être chargé en mémoire, le système tente d'allouer à chaque segment de l'espace logique du processus, une zone contiguë en mémoire, dans l'ordre suivant : le segment de code, le segment de données et le segment de pile.

En cas de succès, chaque segment est chargé dans la zone contiguë allouée au segment. La taille de l'espace alloué à chaque segment est exactement égale au nombre de pages qu'il y a dans le segment (mémoire à partition variable). Un processus chargé en mémoire y séjournera jusqu'à la fin de son exécution (pas de va-et-vient et pas de relocation). En cas d'échec, le chargement du processus est retardé et aucun espace n'est alloué au processus jusqu'à la prochaine tentative de chargement en mémoire. Lorsqu'un processus se termine, le processus libère son espace mémoire et le système tente de charger en mémoire d'autres processus.

### Première partie : Allocation d'espace en mémoire

Considérez les 5 processus du tableau suivant :

Instant d'arrivée	Processus	Taille	Temps
0	A	5 MO + 6 MO + 1 MO	8 ms
3	B	4 MO + 1 MO + 2 MO	10 ms
6	C	2 MO + 1 MO + 1 MO	12 ms
9	D	2 MO + 1 MO + 2 MO	16 ms
12	E	6 MO + 3 MO + 1 MO	5 ms



Par exemple, la première ligne du tableau signifie que le processus A arrive à l'instant 0, son espace d'adressage logique est composé d'un segment de code de 5 MO, d'un segment de 2 données de 6 MO et d'un segment de pile de 1 MO. Lorsque le processus A est chargé en mémoire, il y séjournera pendant 8 ms.

1/- Donnez sous forme de diagrammes de temps l'évolution de l'état de la mémoire aux différentes étapes de traitement de ces processus, sous les hypothèses suivantes :

- Le mode d'allocation des trous pour les segments est le premier ajustement (First Fit) - Le répartiteur de haut niveau fonctionne selon PAPS (Premier Arrivé, Premier Servi). Le répartiteur de haut niveau a pour rôle d'ordonnancer les demandes de chargement en mémoire.

2/- Cette stratégie d'allocation d'espace souffre-t-elle de fragmentation interne ? De fragmentation externe ? Justifiez votre réponse.

3/- Expliquez pourquoi le gestionnaire de la mémoire n'alloue aucun espace à un processus s'il n'y a pas suffisamment d'espace pour les trois segments du processus (la politique du tout ou rien).

### Deuxième partie Translation d'adresse

Supposez que l'adressage logique est sur 24 bits. Les 2 premiers bits (de poids fort) indiquent le numéro de segment. Les 10 bits suivants donnent le numéro de page. Les 12 derniers bits sont réservés au déplacement dans la page. L'adresse physique est aussi sur 24 bits. Les 12 bits de poids fort indiquent le numéro de case. Les 12 derniers bits spécifient le déplacement dans la case.

1/- Donnez le nombre de cases en mémoire physique.

2/- Expliquez comment convertir une adresse logique en une adresse physique.

3/- Considérez l'adresse logique suivante : 01 00 0000 0010 0000 0011 0001

Donnez l'adresse physique correspondant à l'adresse logique précédente, si le segment 01 est chargé dans une zone contiguë commençant à la case numéro 3 : 0000 0000 0011



4/- Est-il possible de réaliser la translation d'adresse lors du chargement (dites pourquoi) ?

### Exercice N° 2 :(7pts)

La tentative d'exécution d'une instruction qui ne figure pas dans le répertoire d'un processeur donne lieu à un déroutement pour instruction inexistante. Par exemple, on peut utiliser ce mécanisme pour réaliser, par programme, la simulation d'un processus industriel. En particulier, pour ce faire, nous disposons de trois instructions spécifiques pour le contrôle automatique du haut fourneau, celles-ci sont :

- ⚡ **ATE** : Pour augmenter la température.
- ⚡ **DTE** : Pour diminuer la température.
- ⚡ **INIT** : Pour initialiser les paramètres du haut fourneau.

L'effet de ces trois instructions de contrôle est simulé en appelant les procédures correspondantes (procédures utilisateurs) lors du traitement du déroutement. La détermination des opérandes de ces instructions, se fait en analysant le contenu de l'instruction elle-même, qui contient leurs adresses, la transmission des résultats utilisent les conventions propres de l'instruction traitée (passage dans un registre, etc.).

a./ Ecrire le programme de déroutement associé.

b./ Un ordinateur est chargé cette fois-ci de contrôler réellement le haut fourneau. Il exécute périodiquement le cycle de contrôle suivant :

1. Travail de fond
2. Opération d'augmentation de température (chauffage : déclenché chaque 50ms)
3. Travail de fond
4. Opération de diminution de température (refroidissement: déclenché chaque 50ms)

On dispose d'une horloge de période de  $10\mu s$  ( $1\mu s = 10^{-6}s$ ) et qui provoque une interruption au passage du compteur à zéro. En supposant que les temps pris par les opérations 2 et 4 sont de quelques millisecondes (ms). Et que le travail de fond ne termine jamais, on demande d'écrire les routines de ce système.

**Bonne chance.**