SR02 : TD 1

TD Exercices sur deux séances de 2h

Exercice 1. (Multiprogrammation et temps partagé)

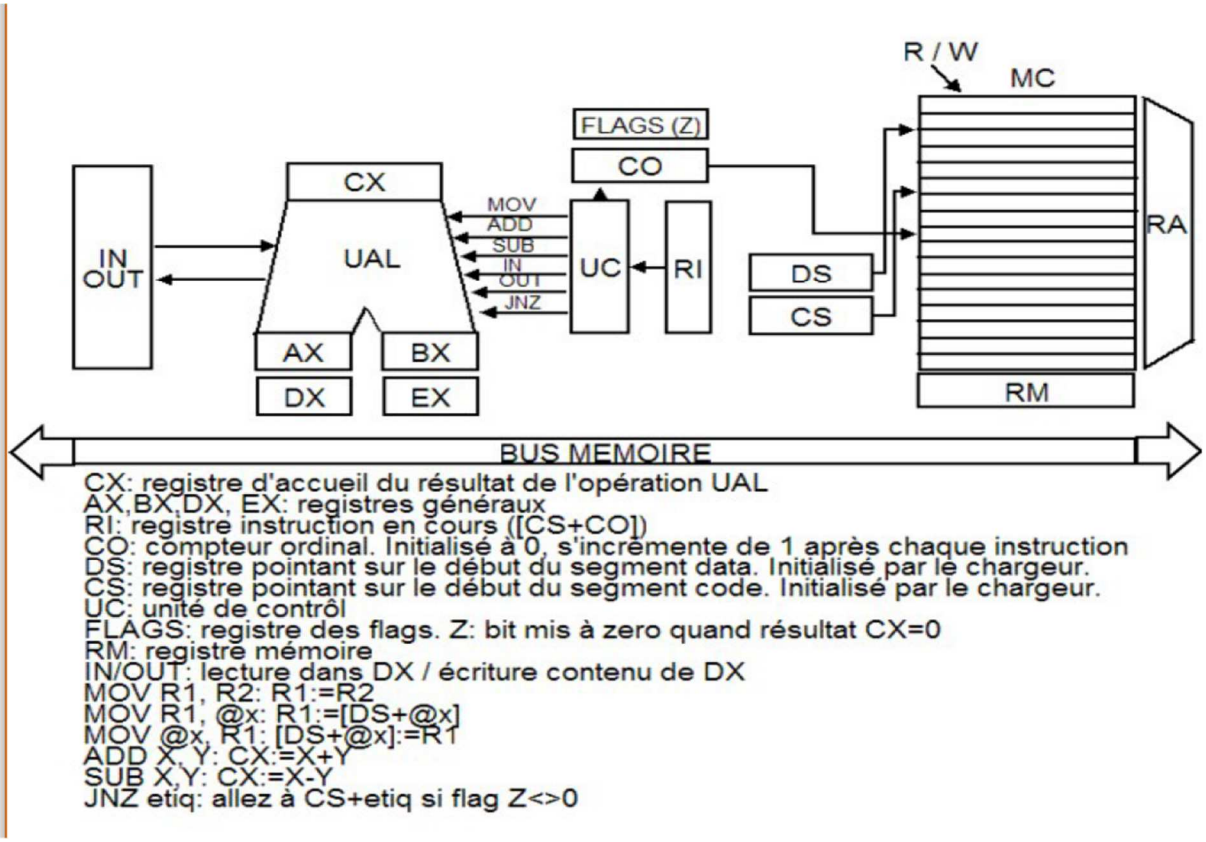
Les actions de deux programmes sont définies comme suit :

P1 : Arrivée à l'instant 0 Calcul pendant 3 unités Impression sur imprimante pendant 2 unités Calcul pendant 2 unités Affichage à l'écran pendant 1 unité Calcul pendant 3 unités FIN.	P2 : Arrivée à l'instant 1 Calcul pendant 2 unités Affichage à l'écran pendant 2 unités Calcul pendant 1 unité Impression sur imprimante pendant 2 unités FIN.

1) Elaborer les diagrammes d'exécution dans les cas suivants :
   - Système monoprogrammé
   - Système multiprogrammé
   - Système à temps partagé
2) Calculer dans chaque cas le taux d'occupation du CPU.

Exercice 2. (Cheminement d'un programme)

Soit la micro-architecture d'un calculateur illustrée à la figure suivante :



Supposons que chaque instruction occupe un mot mémoire. La durée d'exécution d'une instruction est de 1 unité de temps. Une opération d'entrée sortie dure 3 unités de temps. Plusieurs programmes peuvent être logés simultanément en mémoire centrale. Supposons que le système d'exploitation alloue le CPU à tour de rôle aux processus prêts: quantum=2 unités de temps. Le système doit favoriser les processus qui font plus d'entrées sorties. Lorsqu'une entrée/sortie est terminée, le matériel envoie un signal qui sera intercepté par le système d'exploitation. A la réception d'un tel signal, le système débloque le processus en attente de la fin de cette opération d'entrée / sortie et le met dans la file des processus prêts.

1) Quels sont les états possibles d'un processus dans ce système.
2) Quelles sont les files d'attente nécessaires pour gérer les processus de ce système.
3) Y'a-t-il une priorité entre les processus de ce système. Si oui, comment la calculer ?
4) Définir la structure du PCB qui permettra au système de reprendre l'exécution d'un processus après lui avoir réquisitionné le CPU.
5) Soient les deux programmes définis comme suit:

Program 1

Var j

*j:=3*2*

Ecrire j

END.

Program 2

Var i,k

k:=3

Lire i

k:=k+i

Ecrire k

END.

a) Compiler les deux programmes pour avoir le code assembleur correspondant.
b) Charger les deux programmes en mémoire centrale.
c) Dérouler l'exécution des deux programmes en illustrant l'évolution des files d'attentes de PCB, et les contenus des PCB.

Exercice 3. (Routine d'interruption)

Un calculateur reconnaît 6 causes d'interruption, qui sont:

- EX: action faite par un organe directement connecté
- A: fin d'opération sur le canal d'E/S
- B: fin d'opération sur le canal d'E/S
- C: fin d'opération sur le canal d'E/S
- D: fin d'opération sur le canal d'E/S
- H: déclenchement de l'horloge

Pour que l'un de ces événements conduise à une interruption, deux conditions sont nécessaires:

- le système d'interruption doit être actif

- la condition doit être armée, ce qui est réalisé en plaçant un "1" dans la position correspondante d'un registre masque appelé RM. Celui-ci comporte 6 bits qui correspondent de gauche à droite à EX, A,B, C, D et H.

L'interruption a alors lieu de la façon suivante:

- le compteur ordinal est stocké en mémoire à l'adresse 0
- le système d'interruption est désactivé
- la valeur I (1(EX), 2(A), 3(B), 4(C), 5(D), 6(H)) est placée dans le compteur ordinal.

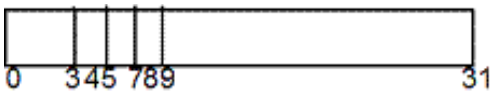
On désire programmer la gestion des interruptions de telle sorte que les priorités soient décroissantes de EX à H. Cette machine dispose entre autres des instructions suivantes :

- MOV x,y : transfert de y vers x
- STM x : stocke RM dans les 6 bits de droite de x
- MSK x : garnit RM avec les 6 bits de droite de x
- ACT : active le système d'interruption
- DOR : désactive le système d'interruption
- CLA x : charge le contenu de x dans l'accumulateur
- STO x : range le contenu de l'accumulateur dans x
- TRA x : branchement vers l'instruction d'adresse x
- TRA @x : branchement indirect vers l'adresse contenu dans x

1) Ecrire le début et la fin de la routine d'interruption i, en indiquant notamment les valeurs des constantes pour chaque cause.
2) Comment le système se branche vers la routine d'interruption ?
3) En examinant les trois dernières instructions de la routine d'interruption, indiquer pourquoi ce mécanisme comporte une faille ?
4) Quel doit être la valeur de RM à l'initialisation ?

Exercice 4. (Interruption et Appel au superviseur)

On s'intéresse à l'étude des interruptions et déroutements sur une machine X. La machine comporte un seul niveau d'interruption (déclenché par le passage à 0 de l'horloge), un seul déroutement et un seul appel au superviseur. L'horloge proprement dite est un compteur dont le contenu est décrémenté de 1 toutes les 5µs. Un appel au superviseur comporte plusieurs paramètres dont le premier indique la cause de l'appel. Le format du mot d'état du processeur est donné dans la figure suivante.



4 : mode (maître 1/esclave 0)
8 : masque d'it (masqué 1/ démasqué 0)
9-31: compteur ordinal

LPSW(m) : cette instruction permet de charger le mot d'état du processeur par la valeur du mot d'état rangé à l'adresse contenue dans m.

PSW

Relevé périodique de mesures

L'ordinateur est chargé de relever, périodiquement, des mesures sur une installation industrielle. La prise de mesures doit être déclenchée toutes les 100 ms. Comme la durée de la prise de mesures est très inférieure à la durée de cet intervalle, l'ordinateur est occupé, le reste du temps, à l'exécution d'un travail de fond qui est donc, périodiquement interrompu.

- Donner les différents programmes nécessaires à la réalisation de ce système.

Réalisation d'un moniteur d'enchaînement

Un moniteur d'enchaînement lit en séquence et fait exécuter des travaux qui lui sont soumis. A chaque travail sont associés un délai de garde (temps maximal alloué pour l'exécution de ce travail) et une adresse de début d'exécution. Si le délai de garde est écoulé avant la fin du travail, le moniteur l'arrête et lance l'exécution du travail suivant. Les programmes sont chargés à une adresse fixe qui n'a pas donc à être spécifiée comme paramètre. Chaque travail doit obligatoirement se terminer par un appel au superviseur SVC(fin), qui marque la fin du travail (cet appel est automatiquement inséré par le compilateur dans le texte du programme de l'utilisateur). L'appel au moniteur d'enchaînement est déclenché soit par la fin normale du travail (SVC(fin)), soit par l'écoulement du délai de garde.

- Écrire les procédures nécessaires à la réalisation de ce moniteur.