

On dispose d'une machine monoprocesseur doté d'un seul périphérique. Pour les transitions d'état d'un processus, on considère que l'état Bloqué se divise en 2 états :

- **Etat Attente** : processus bloqué en attente de la libération du périphérique.
- **Etat E/S** : processus bloqué, en attente de fin d'exécution de l'E/S.

Les demandes d'E/S sont gérés selon le mode FIFO.

Considérons les 4 programmes suivants :

P1 : Calcul pendant 30ms
Lecture pendant 40ms
Calcul pendant 20ms
Lecture pendant 40ms
Calcul pendant 20ms

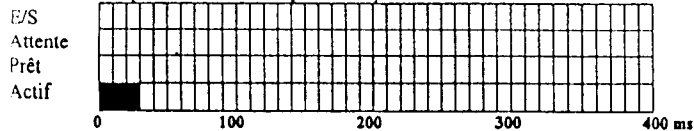
P2 : Calcul pendant 30ms
Lecture pendant 40ms
Calcul pendant 40ms
Lecture pendant 20ms
Calcul pendant 10ms

P3 : Calcul pendant 20ms
Lecture pendant 40ms
Calcul pendant 10ms

P4 : Calcul pendant 40ms

- a. Donner les temps de réponse de chaque programme dans les modes monoprogrammé et multiprogrammé, en remplissant les diagrammes ci-après. A cet effet, vous noircirez les cases correspondantes à l'état du processus.

Pi (Identificateur du processus)



On suppose que la durée d'une tâche de contrôle est négligeable.

- b. Calculer le taux d'utilisation N du processeur dans les 2 modes.

1^{ère} Epreuve de Moyenne Durée
SYS01

Durée : 1h30mn

Exercice 1 (12pts) :

Considérons un processus utilisateur désirant éditer une chaîne P de 8 caractères sur une imprimante.

- a. Le protocole d'impression s'opère comme suit : L'imprimante est reliée à une interface (coupleur de débit 100 car/seconde), disposant de 2 registres :
- Registre de données Printer_DataRegister de 8 bits.
 - Registre d'état Printer_StatusRegister de 16 bits, ayant la structure suivante :



6
└─→ { 1 : périphérique prêt
 0 : impression en cours

Le noyau du système d'exploitation copie la chaîne P dans un buffer (buf) situé dans l'espace noyau (SE), ensuite il vérifie si l'imprimante est prête pour imprimer le 1^{er} caractère dans l'interface de l'imprimante. Dans ce cas, une fois le 1^{er} caractère imprimé, le noyau enchaîne l'opération pour l'impression des autres caractères.

En utilisant l'instruction `Copy_FromUser(adr, buf, N)` qui permet de copier `N` caractères de la chaîne d'adresse `adr` vers la chaîne `buf`, écrire la procédure de sortie d'une chaîne de  caractères.

Quel est le temps d'attente (CPU) pour l'impression de cette chaîne de caractères ?

- b. Afin d'éliminer ce temps d'attente, on utilise le mécanisme d'interruption. Maintenant, le protocole d'impression est dirigé par interruption. L'impression passe toujours par la copie de la chaîne à imprimer dans le buffer système. Puis, le noyau lance l'impression du 1^{er} caractère, et appelle ensuite le gestionnaire des processus (scheduler) afin d'exécuter un autre processus.

Donner les programmes nécessaires à l'implantation de ce système en utilisant les deux primitives :

Scheduler() : appel du gestionnaire de processus (scheduler).

Réveiller() : réveil du processus bloqué en attente de fin d'impression, pour redevenir prêt.

Enable() : activation du système d'interruption.

Si le temps d'exécution d'une interruption est de 1ms, quel est alors le temps de traitement CPU pour la prise en charge de l'impression de 8 caractères ? Quel est le gain CPU par rapport à P ?

- c. Afin d'améliorer ce gain, on utilise le mécanisme DMA dont la structure de la commande est :

Code opération	adresse	Taille
----------------	---------	--------

Donner les programmes nécessaires à l'exécution de ce mode, sachant que le lancement du DMA se fait par l'instruction SetupDMA().

Quel est le gain en temps CPU en supposant que le temps d'exécution de la routine de fin d'E/S est de 1ms ?

Première épreuve de moyenne durée

Exercice 01 :

Soit une machine dont la configuration est la suivante : CPU, MC et Périphériques. On considère l'exécution de quatre programmes P1, P2, P3, P4 selon le mode multiprogrammé. A l'instant 0, tous les programmes présents se trouvent à l'état prêt dans l'ordre P1, P2, P3, P4. Le tableau suivant résume les CPU et Entrées / Sorties de chaque processus.

P1	P2	P3	P4
4 CPU	2 CPU	1 CPU	2 CPU
3 E/S	4 E/S	5 E/S	2 E/S
2 CPU	5 CPU	2 CPU	3 CPU
2 E/S			2 E/S

- Une tâche de contrôle prend une (01) unité de temps quelque soit sa nature.

1. Donner le diagramme de transition d'état des programmes.
2. Construire la synoptique temporelle d'exécution des programmes et calculer le temps d'utilisation du processeur pour l'exécution de chacun des programmes et pour le lot de programme.
3. Donner le contenu des files (associé à chaque état) à chaque transition d'état d'un processus.

Exercice 02 :

Soit une machine dont la configuration est la suivante : CPU, MC et Périphériques. Cette machine dispose de plusieurs niveaux d'interruptions relatives aux périphériques, d'une interruption horloge et d'un appel au superviseur (déroutement lié à l'accès à une adresse mémoire inexistante). Un appel au superviseur comporte plusieurs paramètres : cause de l'appel et le format du mot d'état structuré comme suit :

- | | | |
|----------------------|---------------------------|---------------------|
| - Mode d'exécution : | - Masque d'interruption : | - Compteur ordinal. |
| 0 : esclave | 0 : démasqué | |
| 1 : maître | 1 : masqué | |

LPSW(m) est l'instruction de chargement du mot d'état contenu à l'adresse m.

On s'intéresse à la mesure de la taille mémoire lors de l'initialisation du système. L'accès à une adresse mémoire inexistante provoque un déroutement ayant pour rôle de relancer à chaque fois cette opération (mesure de la taille mémoire). Si cette opération ne réussit pas après une durée de temps D alors il provoque l'arrêt du système.

Ecrire les programmes nécessaires à la mesure de la taille mémoire.

EMDI (durée 1h 30)
SYSTEME D'EXPLOITATION

Exercice 1 :

On désire réaliser un moniteur d'enchaînement sur le système DOS. Il s'agit de lancer séquentiellement une suite de programmes (10 au maximum) préalablement compilés et sauvegardés dans un répertoire « C:\programs ».

Le timer interrompe le processeur 18.2 fois à la seconde et le niveau d'interruption du timer est 0x1C.

Pour cela, on utilisera les primitives suivantes :

-Lister(rep, T) où rep est le nom d'un répertoire et T un tableau de chaînes de caractères. Cette primitive permet de charger dans le vecteur T les noms des fichiers exécutables (programmes) contenus dans le répertoire rep.

-Process(fichier) où fichier est le nom d'un programme (fichier exécutable). Cette primitive permet de charger et lancer le programme contenu dans fichier. Le programme appelant se met en attente jusqu'à la fin du programme lancé.

-Setvect(num,func) met à jour le niveau d'interruption num avec la routine func.

-Getvect(num) renvoie la routine liée au niveau num.

-Disable() désactive le système d'interruption.

-Enable() active le système d'interruption.

Exercice 2 :

Soient un vecteur T de 100 caractères et une imprimante de 80 colonnes gérée par un canal. On désire imprimer le vecteur T sur une nouvelle page (une page à part). Une commande canal possède 4 (quatre) champs selon le format suivant :

31	24 23	16 15	8 7	0
opération	adresse	indicateur	compteur	

opération : 0x01 impression d'une ligne

0x0B saut de ligne

0x 8B saut de page

0x 89 imprime une ligne et fait un saut de page

adresse : adresse de la zone mémoire à transférer

compteur : quantité (en octets) à transférer

indicateur : non utilisé

Une fois que la canal est lancé par le CPU, il retrouve l'adresse de la première commande dans un registre CAW (mot d'adresse canal) qui est à l'adresse 72. Les commandes canal sont dans des adresses successives en mémoire.

On utilise la fonction malloc(nbo) qui retourne l'adresse de la zone mémoire allouée, où nbo est le nombre d'octets à allouer.

1^{ère} Epreuve de Moyenne Durée

SYS01

Durée : 1h30mn

Exercice 1 (12pts) :

Soit la configuration d'un ordinateur constitué d'un processeur central PC, une mémoire centrale, une imprimante IMP, un disque dur DSK, et une horloge qui intervient 1 fois chaque seconde.

Le débit d'impression sur l'imprimante est de 25 entiers/100ms, tandis que le débit de lecture sur disque est de 5 entiers/5ms, et l'écriture sur disque est de 5 entiers/10ms.

Quant au processeur central, il met un temps de 1ms pour exécuter une instruction. Ainsi, exécuter la fonction **generer(i)** et 5ms pour exécuter la fonction **Somme(T)**. La fonction **generer(i)** génère un nombre aléatoire entre 1 et i, et la fonction **Somme(T)** somme les éléments du tableau T.

Les instructions **ECRIRE (PER, X, N)**, respectivement **LIRE (PER, X, N)**, permettent d'écrire ou imprimer sur (respectivement lire du) le périphérique PER, à partir de la mémoire X, N éléments.

1. On considère le mode d'exploitation multiprogrammé des 3 programmes suivants :

Pgm 1	Pgm 2	Pgm 3
Debut	Debut	Debut
Pour i ← 1 à 3	i ← generer(5);	V[1] ← generer(100);
Faire T[i] ← generer(i);	Lire(DSK, T[i], 1);	V[2] ← 0;
Ecrire(DSK, T[i], 1);	J ← generer(10);	V[3] ← 45;
Fait;	Lire(DSK, T[i], 1);	Ecrire(IMP, V, 3);
Ecrire(IMP, T, 3);	s ← T[i] * T[i];	Fin
i ← somme(T);	Ecrire(DSK, s, 1);	
Ecrire(IMP, i, 1);	Fin	
Fin		

Une tâche de contrôle dure approximativement une 1ms, et qu'à l'instant t = 0 les programmes sont présents en mémoire dans l'ordre pgm1, pgm2, pgm3 et les programmes se terminent par l'instruction SVC(fin). Donnez le diagramme d'exécution des 3 programmes, en calculant pour chaque programme le temps CPU, le temps d'entrée/sortie consommé et le temps de réponse.

2. Ecrire les programmes nécessaires à ce système multiprogrammé qui permettent de donner le temps de réponse, le temps cpu et le temps d'entrée/sortie associés à chaque programme. Il faudra tenir compte de l'arrivée dynamique des programmes, et en utiliser les procédures suivantes :

Insere(F,m) : insere l'element m dans la queue de la file F

delete(f) : supprime l'element de tete de la file f

Tete (F) : recupere l'element en tete de file F

Vide (F) : teste si la file F est vide

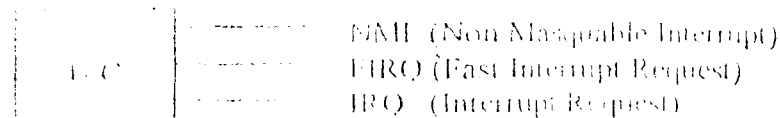
Le PSW est lancé le programme dont le mot d'état est m

Exercice 2 (8pts) :

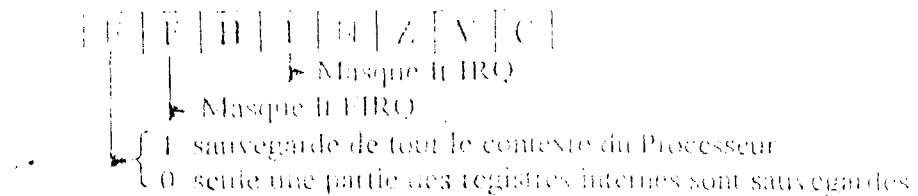
Le cœur du microprocesseur 6800 est constitué de :

- 2 registres accumulateurs A et B de 8 bits,
- 2 registres indexés X et Y de 16 bits,
- d'un registre pointeur U de la pile utilisateur,
- d'un registre pointeur S de la pile système,
- d'un registre de page DP de 8 bits,
- d'un registre de codes condition CCR de 8 bits,
- et du compteur ordinal PC de 16 bits

Il présente trois niveaux d'interruption matérielles



L'interruption de type NMI ne peut être masquée. Par contre, les interruptions FIRO et IRQ peuvent être masquées. La ligne FIRO est prioritaire sur IRQ : c'est une interruption qui doit être gérée rapidement. Pour cela, FIRO ne sauvegarde que les registres PC et CCR. Les interruptions IRQ et NMI, quant à elles, sauvegardent tous les registres (sauf S) dans la pile S. La structure du registre CCR est donnée comme suit :



Une interruption a alors lieu de la façon suivante :

1. Le compteur ordinal PC et le registre CCR sont empilés dans la pile système S.
2. Les masques associés à IRQ et FIRO sont remplis par les valeurs correspondantes.
3. La valeur I est remplie selon le niveau d'interruption activé.
4. Le microprocesseur dispose aussi des instructions suivantes :
 - RTI : retourner au programme interrompu
 - PSHS les registres : empile les registres dans la pile système S. La liste des registres est donnée dans l'ordre d'empilement.
 - PPHS les registres : dépile les registres depuis la pile système S. La liste des registres est donnée dans l'ordre d'empilement.
5. Donnez l'ordre de priorité des interruptions.
6. Expliquez comment le système utilise le bit I lors de la sauvegarde ou restauration du contexte en précisant sa valeur pour chaque niveau d'it, et quelle est la valeur du masque à chaque empilement à chaque niveau.
7. Donnez les cannes d'interruptions associées aux 3 niveaux.

Réponse à l'exercice