VARI-Exercice Dirigé

Systèmes d'exploitation

Exercice 1 - Chaîne de production d'un programme

(d'après C.Carrez Exercices de Systèmes A)

On considère un langage d'assemblage dont les instructions sont codées de la manière suivante :

`LOAD <adresse>`	10000 + adresse	(AC) := MEMOIRE(adresse)
`ADD <constante>`	20000 + constante	(AC) := (AC) + constante
`MUL <constante>`	30000 + constante	(AC) := (AC) * constante
`STORE <adresse>`	40000+ adresse	MEMOIRE(adresse):= (AC)
`JMP <adresse>`	50000 + adresse	aller à l'instruction située à adresse

Question 1

Rappeler les différentes étapes de la chaîne de production d'un programme.

Etude de l'assembleur

Soit le petit programme suivant :

```
          NOM   ESSAI       nom du programme
A :       MOT   =25         case initialisée à 25
B :       MOT   1           case non initialisée
          LOAD  A
          MUL   10
          ADD   1
          STORE B
          FIN
```

Le résultat de l'assemblage d'un programme appelé module objet, est un fichier comprenant : le nom et la taille du programme, pour chaque instruction l'adresse relative, code machine en adressage absolu ou relatif.

Question 2

Complétez le tableau ci-dessous représentant les enregistrements du module objet.

type	nom ou adresse relative	taille ou code ou valeur	Texte
entête	ESSAI		NOM ESSAI
code absolu			A : MOT =25
code relatif			LOAD A
code absolu			MUL 10
code absolu			ADD 1
code relatif			STORE B
	FIN		

Etude de l'éditeur de liens

Le programme utilise un autre module en se débranchant à l'adresse symbolique SUITE de ce module; de même, il peut être utilisé par un autre module à l'adresse symbolique INCR. SUITE sera appelé lien à satisfaire, il sera résolu par l'édition de liens, INCR est appelé lien utilisable.

Le programme devient :

```
          NOM    ESSAI        nom du programme
          PUBLIC INCR point d'entrée dans ESSAI
          EXTERN SUITE        point d'entrée dans autre module
A :       MOT    =25          case initialisée à 25
B :       MOT    1            case non initialisée
INCR:     LOAD   A
          MUL    10
          ADD    1
          STORE  B
          JMP    SUITE
          FIN
```

Question 3

Modifier le module objet issu de l'assemblage du programme ci-dessus, en complétant le tableau :

type	nom ou adresse relative	taille ou code ou valeur	Texte
entête	ESSAI		NOM ESSAI
lien utilisable	INCR		PUBLIC
lien à satisfaire	SUITE		EXTERN
code absolu			A : MOT =25
code relatif			LOAD A
code absolu			MUL 10
code absolu			ADD 1
code relatif			STORE B
code avec lien à satisfaire			JMP SUITE
	FIN		

Lors d'une édition de liens d'un programme, le module ESSAI est placé à l'emplacement relatif 123 dans le programme exécutable. SUITE est un lien utilisable placé dans un autre module à l'emplacement relatif 8. Cet autre module est placé immédiatement derrière le module ESSAI dans le programme.

Question 4

Donnez les enregistrements fournis par l'éditeur de liens pour le module ESSAI en complétant le tableau ci dessous.

type	adresse	code ou valeur	Texte
code absolu			A : MOT =25
code relatif			LOAD A
code absolu			MUL 10
code absolu			ADD 1
code relatif			STORE B
code relatif			JMP SUITE

Etude du chargeur

Le chargeur place le programme à l'emplacement 1042 de la mémoire.

Question 5

Donnez le contenu des emplacements mémoire contenant le module ESSAI.

Exercice 2 : Exemple d'éditions de liens

(d'après C.Carrez Exercices de Systèmes A)

On dispose d'un ensemble de modules définis comme suit:

```
module PROGRAMME   taille:       332
                   liens à satisfaire:     OUVRIR
                                           LIRE
                                           FERMER
                                           EDITER
                   adresse lancement:      133
module ETIQUETTE   taille:       128
                   liens utilisables:      NOM         0
                                           SOCIETE    32
                                           ADRESSE    64
                                           CODEPOST   96
                                           VILLE     101
module LECTURE     taille:       840
                   liens utilisables:      OUVRIR      0
                                           LIRE      340
                                           FERMER    732
                   liens à satisfaire:     NOM
                                           SOCIETE
                                           ADRESSE
                                           CODEPOST
                                           VILLE
module IMPRESSION  taille:       212
                   liens utilisables:      IMPRIMER    0
module EDITION     taille:       642
                   liens utilisables:      EDITER      0
                   liens à satisfaire:     NOM
                                           SOCIETE
                                           ADRESSE
                                           CODEPOST
                                           VILLE
                                           IMPRIMER
```

On effectue l'édition de liens des modules `PROGRAMME`, `ETIQUETTE`, `LECTURE`, `IMPRESSION` et `EDITION`. Donner, en justifiant brièvement votre réponse:

- les adresses d'implantations de ces modules,
- la taille totale du programme résultant,
- la table des liens après le premier passage,
- l'adresse de lancement du programme résultant.

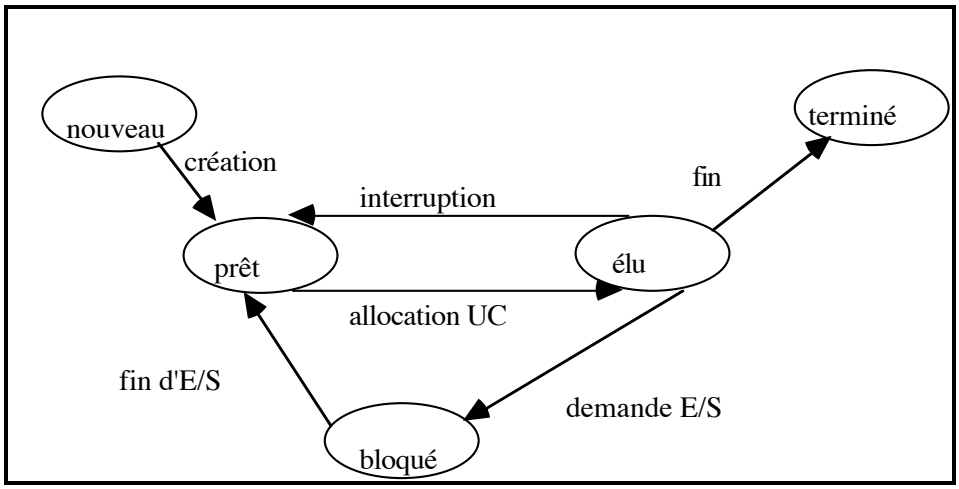
Exercice 3 : Processus en multiprogrammation.

On considère un système monoprocesseur dans lequel les processus partagent un disque comme seule ressource (autre que le processeur). Le disque ne peut être utilisé que par un seul processus à la fois : dès qu'une entrée/sortie disque s'exécute pour le compte d'un processus, elle doit se terminer avant qu'on puisse lancer une autre entrée/sortie disque.

Les entrée/sortie disque sont gérées par un dispositif d'accès direct à la mémoire(DMA).

Le système fonctionne en multiprogrammation.

Le schéma des états possibles d'un processus dans ce système est le suivant :



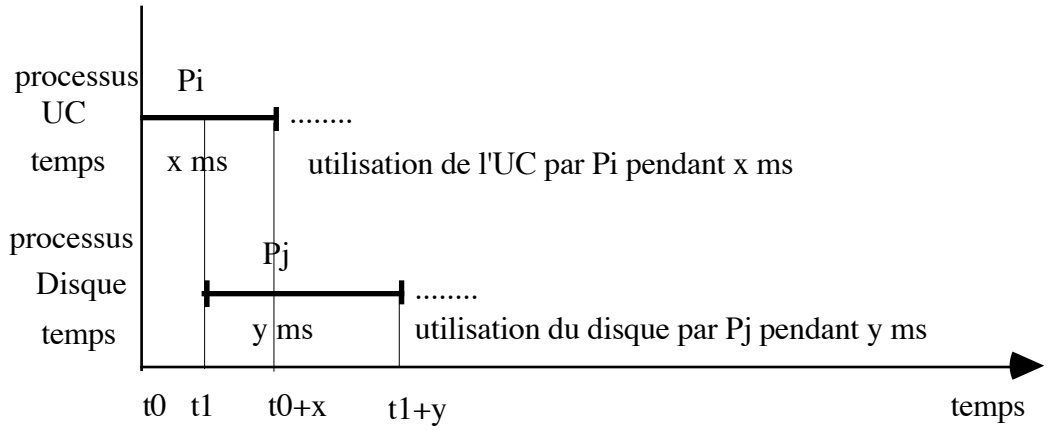
On considère les 2 processus dont le comportement est le suivant :

```
P1     calcul pendant 6 ms;
       lecture disque 10 ms;
       calcul pendant 3ms;

P2     calcul pendant 1 ms;
       lecture disque pendant 5 ms;
       calcul pendant 2ms;
       lecture disque pendant 10 ms;
```

On définit par *temps de réponse* pour un processus, l'intervalle de temps écoulé entre le lancement du processus (création) et la fin d'exécution du processus (fin), par *temps global de réponse*, le temps mis par le système pour exécuter les 2 processus.

Pour répondre aux questions, on pourra s'aider du schéma de chronogramme suivant:



Question 1

On lance et exécute P1, on lance ensuite immédiatement P2 et on l'exécute .

Quels sont les temps de réponse pour P1, pour P2 ? Quel est le temps global de réponse ?

Question 2

On lance simultanément P1 et P2 et on les exécute.

Si les deux processus sont dans l'état "prêt", le processeur sera alloué en priorité à P2.

Quels sont les temps de réponse pour P1, pour P2 ? Quel est le temps global de réponse ?

Comparez aux résultats obtenus dans la question 4 et commentez.