Corrigé ED 4

Exercice 2 :Adressage dans une mémoire virtuelle paginée

Question 1

a) Taille de la mémoire réelle : 1 Mo soit 2**20 octets, taille d'une page : 512 octets soit 2**9 octets.

Taille de l'espace adressable : 2**24, nombre de bits d'une adresse virtuelle : 24 bits, nombre de bits du déplacement : 9 bits, nombre de bits du numéro de page : 15.

Nombre de bits d'une adresse réelle : 24 bits, nombre de bits du déplacement : 9 bits, nombre de bits du numéro de case : 15. A noter que les 4 premiers bits du numéro de case sont à 0 puisque la taille de la mémoire est 2**20, il n'y a donc que 2**11 cases. Le nombre d'entrées dans la table des pages est 2**15.

b) La taille du programme est 2000 octets (chaine) + 2 octets (i) + 20 octets (instructions) soit 2022 octets. On doit allouer 4 cases, soit 2048 octets, il y a donc une fragmentation interne de 26 octets.

c) référence	adresse virtuelle	adresse réelle
chaine(10)	0 10	9 10
chaine(515)	1 3	17 3
chaine(1026)	2 2	65 2
chaine(1999)	3 463	33 463

Exercice 3

Mécanisme de gestion d'une mémoire virtuelle paginée

Question 1

bit V sert à générer un défaut de page si la page n'est pas présente en mémoire

bit M sert lors d'un remplacement de page, à éviter la recopie sur disque, d'une page qui n'a pas été modifiée

bit A sert à l'algorithme de remplacement de page type algorithme de l'horloge ou seconde chance.

Question 2

Voir également le cours

Structure des informations

```
type adresse : article
        npage : entier;
        depl : entier;
    fin article;
type defpage : article
        V : booleen;
        M : booleen;
        A : booleen;
        -- P :  protection
        NC : entier; numero de case
    fin article;
type instruction : (ecriture, autre);
type Tabpage : array (0..N-1) de defpage;  -- N taille d'une table de pages dans l'exemple 2**24
procedure decodage (AR : out adresse; AV : in adresse; I : in instruction)
P : defpage;
debut
    P:= Tabpage(AV.npage);
    si non P.V alors activer le processus remplacement de page; attendre; fin si;
    P.A:=1; si I=ecriture alors P.M:=vrai; fin si;
    AR.depl := AV.depl; AR.npage:=P.NC;
fin decodage;
```

Traitement d'un défaut de page

1 Recherche de l'adresse sur disque de la page en défaut

2 Trouver une case libre

a Si case libre alors l'utiliser

b Sinon, utiliser un algorithme de remplacement de page pour sélectionner une page victime

c Ecrire la page victime sur disque; modifier les tables de pages et de cases. Amélioration : ne ré-écrire la page que si elle a été modifiée depuis son chargement.

3 Charger en mémoire la page en défaut dans la case libre.

Question 3

La taille de la table des pages par processus est a priori 2**24 prohibitif !

Quelques solutions pour réduire la table des pages :

- Pagination de la table des pages

- Séparer l'espace virtuel de l'utilisateur en espace propre et espace système : toutes les procédures système utilisées par le processus sont dans l'espace du système et partagées. Enfin, la taille de la table des pages peut être calculée, en fin d'édition de liens par exemple, et on n'alloue que la place nécessaire...

- Utiliser une table des pages inverses voir cours

- Enfin, dans tous les cas on peut utiliser une mémoire associative.

On pourra éventuellement étudier un compromis entre taille d'une page et nombre de pages.

Question 4

Politique FIFO

	1	2	3	4	5	6	7	8	9	10	11
1	**3**	3	3	3	**3**	**9**	9	9	9	9	10
2		**5**	5	5	5	5	5	**12**	12	12	12
3			**6**	6	6	6	**6**	6	**3**	3	3
4				**8**	8	8	8	8	8	**6**	6
	x	x	x	x		x		x	x	x	x

Nombre de défauts de pages : 9 dont 4 dûs à la pagination à la demande

Politique LRU

	1	2	3	4	5	6	7	8	9	10	11
1	**3**	3	3	3	**3**	3	3	3	**3**	3	3

2		**5**	**5**	**5**	**5**	**9**	**9**	**9**	**9**	**9**	**10**
3			**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**	**6**
4				**8**	**8**	**8**	**8**	**12**	**12**	**12**	**12**
	x	**x**	**x**	**x**		**x**		**x**			**x**

Nombre de défauts de pages : 7 dont 4 dûs à la pagination à la demande