

Examen de Système d'Exploitation II

Nature de l'épreuve : DC <input type="checkbox"/> D.S. <input type="checkbox"/> E.F. <input checked="" type="checkbox"/>	Documents : autorisés <input type="checkbox"/> non autorisés <input checked="" type="checkbox"/>
Date de l'épreuve : 26/06/2024	Calculatrice : autorisée <input type="checkbox"/> non autorisée <input checked="" type="checkbox"/>
Durée de l'épreuve : 01H30	Session : principale <input checked="" type="checkbox"/> contrôle <input type="checkbox"/>
Section : LIRS 1 – LGSSI 1	Barème approximatif : 07-08-05

NB :

- Le sujet comporte 04 pages. Les réponses sont à reproduire impérativement sur les feuilles de l'énoncé sauf indication contraire.
- Rédigez soigneusement les solutions en utilisant les mêmes conventions vues en cours. La clarté et la concision des réponses sont des éléments importants d'appreciation. Une écriture illisible ne sera ni corrigée ni notée.
- Il est strictement interdit d'écrire au crayon noir ou d'utiliser le correcteur.
- Tous les exercices sont indépendants.
- Il est conseillé de lire attentivement et entièrement chaque exercice avant de commencer la résolution.

Exercice 1 : Questions à courte réponse

[.../7 points]

1. Etudier l'exactitude des phrases suivantes. Mettre dans la troisième colonne du tableau V si la proposition est correcte et F si la proposition est fausse. Dans le cas où la proposition est fausse, soulignez la partie (généralement un mot) qui a erroné cette phrase et proposez une version correcte. Une proposition fausse non correctement corrigée n'est pas comptabilisé.

N°	Proposition	V ou F	Correction
1.	Un programme relogable est un programme dont l'exécution est dépendante de l'emplacement de son chargement.	V	
2.	La segmentation est visible au programmeur	F	la page est visible au programmeur
3.	La segmentation n'est pas visible au programmeur	(F)V	
4.	La page est une unité logique de protection et partage.	F	unité et non protéger.
5.	La segmentation requiert un matériel plus complexe pour la traduction des adresses.	F	gratuit pour la traduction
6.	La segmentation souffre de la fragmentation interne.	F	Segmentation ne pas réaliser une fragmentation interne
7.	La segmentation produit, en moyenne une fragmentation interne, d'un demi cadre par programme.	V	
8.	Les pages sont invisibles au programmeur, au compilateur, à l'assembleur, seules les adresses absolues sont employées.	EV	
9.	Un programme peut être exécuté sur différents systèmes employant des dimensions de pages différentes.	VF	des segments différents

Proposition

V ou F

Correction

10. Lorsque le nombre de pages au niveau d'un système devient important, la consultation de la table des pages peut être lente et inefficace s'il y a beaucoup de pages non chargées. On la segmente alors par processus. C'est la pagination segmentée.

V

2. Soient :

- une mémoire centrale paginée ayant m cadres
- un ensemble de n pages qui se partagent ces m cadres (avec $n > m$).
- une chaîne de p références à ces cadres (au début tous vides, $p > m$).

Donner une limite inférieure (le nombre minimal) et une limite supérieure (le nombre maximal) de défauts de pages indépendamment de l'algorithme de remplacement utilisé. Justifier votre réponse.

La limite inférieure de défaut de page : 2^{n-m}

La limite supérieure de défaut de page : $2^m - 2^n$

Exercice 2 – Segmentation & pagination

[8 points]

1) On considère la table des segments suivante pour un processus P1 :

Segment	Base	Limite
0	540	234
1	1254	128
2	54	328
3	2048	1024
4	976	200

a) Préciser les étapes permettant de transformer une adresse virtuelle en une adresse physique.

pour transformer une adresse virtuelle en une adresse physique
on ajoute au segment la base puis on l'ajoute à l'adresse physique

b) En se basant sur la réponse à la question précédente, calculer, lorsque c'est possible, les adresses réelles correspondant aux adresses virtuelles suivantes (il faut signaler éventuellement les erreurs d'adressage). Remplir pour répondre le tableau suivant :

NB : Les adresses sont données sous la forme (segment:déplacement)

Adresse virtuelle	Valide (Oui/Non)	Adresse physique
(0:128)	oui	128
(1:100)	oui	101
(2:465)	oui	468
(3:888)	non	888
(4:100)	oui	104
(4:344)	oui	344

c) L'adresse virtuelle (4,200) est-elle valide ? Justifier votre réponse.

non l'adresse (4,200) n'est pas valide.

- 2) On considère maintenant un système qui gère sa mémoire d'une manière paginée, les pages sont de taille 256 mots mémoire. La taille d'un mot mémoire est 4 octets. Le système utilise une **stratégie de remplacement locale** avec 4 cadres de la mémoire centrale accordés pour chaque processus. On considère la table des pages suivante du processus P1 :

N° de la page	0	1	2	3	4	5	6	7
Numéro du cadre (en binaire)	011	001	000	010	100	111	101	110
Présence en mémoire	oui	non	oui	non	non	non	oui	Non

- a) Quelle est la taille de **l'espace d'adressage virtuel** du processus P1 ? Justifier votre réponse.

- b) Quel est la taille de la mémoire physique de ce système ? justifier votre réponse.

- c) Préciser les étapes permettant de transformer une adresse virtuelle en une adresse physique
*pour transformer une adresse virtuelle en une adresse physique
on ajoute à ce segment le numéro de page et puis on l'ajoute au
les ajoutant cela le déplacement*

- d) En se basant sur la réponse à la question précédente, calculer, **lorsque c'est possible**, les **adresses réelles** correspondant aux adresses virtuelles suivantes (il faut signaler éventuellement les erreurs d'adressage). Remplir pour répondre le tableau suivant :

NB. Les adresses virtuelles sont exprimées en nombre de mots.

Adresse virtuelle	Numéro de la page	Déplacement dans la page	Numéro du cadre	Adresse physique
240.	0			
546				
1578				
2072				

- e) Que se passe-t-il si P1 génère l'adresse virtuelle 770 ?

*Si P1 génère l'adresse virtuelle 770 on va avoir une
fragmentation interne*

- f) On considère l'adresse virtuelle suivante (en Hexadécimale) : 0007H Sachant que les 4 bits de poids fort désigne le numéro de page et que 12 bits suivants représentent le déplacement dans la page, donnez l'adresse physique (exprimée en hexadécimale) correspondant à cette adresse.

Exercice 3 – Algorithmes de remplacement de pages

[.../5 points]

On considère un système qui gère sa mémoire d'une manière paginée, les pages sont de taille 100 mots mémoire. La taille d'un mot mémoire est 4 octets.

Soit un programme qui possède un espace virtuel de 600 mots et qui s'exécute sur ce système. On considère la suite des adresses virtuelles suivantes (exprimés en mots mémoire) :

34; 123; 145; 510; 456; 345; 412; 10; 14; 12; 234; 336; 412.

1) Donner la suite des numéros de pages référencés.

~~2, 3, 1, 2~~

2) Le programme dispose de 3 cadres en mémoire centrale, initialement vides. Calculer le nombre de défauts de page pour les algorithmes de remplacement suivants : optimal, FIFO, LRU et l'algorithme de la seconde chance.

Algorithme optimal de beladay :

34	34	34	510	510	510	412	412	412	412	412	412	412	
-	123	123	123	456	456	456	10	10	10	10	10	10	336
-	-	145	145	145	345	345	345	14	14	14	14	14	234
1	1	1	1	1	1	1	1	1	1	1	1	1	0

Nombre de défauts de pages de Algorithme optimal de beladay : 12

Algorithme FIFO :

34	34	34	510	510	510	412	412	412	412	412	412	412	
-	123	123	123	456	456	456	10	10	10	10	10	10	336
-	-	145	145	145	345	345	345	14	14	14	14	14	336
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Nombre de défauts de pages de l'algorithme FIFO : 13

Algorithme LRU :

34	34	34	510	510	510	412	412	412	412	412	412	412	
-	123	123	123	456	456	456	10	10	10	10	10	10	336
-	-	145	145	145	345	345	345	14	14	14	14	14	234
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Nombre de défauts de pages de l'Algorithme LRU : 19

Algorithme de la seconde chance:

34	34	34	510	510	510	412	412	412	412	412	412	412	
-	123	123	123	456	456	456	10	10	10	10	10	10	336
-	-	145	145	145	345	345	345	14	14	14	14	14	234
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Nombre de défauts de pages de l'Algorithme de la seconde chance : 13

Bon Travail