Université de Bordj Bou Arréridj
Faculté mathématique et Informatique.
Département Informatique.
Durée 1H : 30 de 10:30 à 12:00. **Le : 24/05/2014**

Année universitaire 2013/2014
2ème Année Licence informatique.
Module : **Systèmes d'exploitation 1.**
Documents non autorisés.

Examen Rattrapage

Exercice 01 (5 points) (Questions de cours)

1- Donner les objectifs du système d'exploitation?

2- Quel est le rôle d'un PCB ? et quelles sont les trois principales catégories d'information d'un PCB ?

3- Dans un système Ms-Dos, est ce qu'un processus qui fait une requête pour une ressource non disponible serait

4- loqué dans la file d'attente de cette ressource ? Expliquer.

5- Quelles sont les états d'un processus ?

6- Donner le rôle et l'objectif de l'MMU ?

Exercice 02 (4 points) (l'édition de liens)

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

module PROGRAMME (1)	taille: 450		module ETIQUETTE (4)	taille: 957	
	liens à satisfaire:	OUVRIR LIRE FERMER EDITER		liens utilisables:	NOM 15 SOCIETE 223 ADRESSE 425 CODEPOST 912 VILLE 950
	liens utilisables:	RUN 12			
module LECTURE (2)	taille: 460		module IMPRESSION (5)	taille: 214	
	liens utilisables:	OUVRIR 2 LIRE 250 FERMER 533		liens utilisables:	IMPRIMER 121
	liens à satisfaire:	NOM SOCIETE ADRESSE CODEPOST VILLE			
module EDITION (3)	taille: 642				
	liens utilisables:	EDITER 54			
	liens à satisfaire:	NOM SOCIETE ADRESSE CODEPOST VILLE IMPRIMER			

1- Les adresses d'implantations de ces modules après la construction de programme finale?
2- La taille totale du programme résultant ?
3- La table des liens après la translation des modules?

Taille 2723

Exercice 03 (5 points) (gestion de processus)

Nous considérons un tableau qui contient les informations de quatre processus A,B,C,D :

1- Dessiner (représenter) un schéma illustrant leur exécution à l'aide de :
- l'algorithme FCFS.
- l'algorithme SRT.
- l'algorithme à tourniquet RR (quantum = 2).

2- Donner le temps de réponse moyen de ces processus dans chaque type d'ordonnancement.
3- Donner le temps d'attentes moyen de ces processus dans chaque type d'ordonnancement.

Processus	Date d'arrivée	Temps de traitement
A	0.000	4
B	2.001	7
C	3.001	2
D	3.002	2

Exercice 04 (6 points)

1_ Comment transférer l'adresse logique vers l'adresse physique dans un système paginée ?

2_ Donner le mécanisme du fonctionnement pour un système de segmentation paginée ?

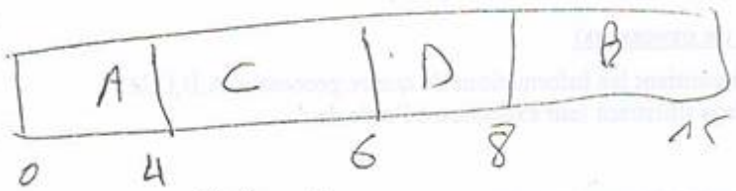
3_ Donner le nombre des processus, des instructions printf avec le schéma

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main() {
int valeur;
 int valeur1;
        valeur1 =  fork();
  printf ("printf-0 \n");
        valeur = fork();
  printf ("printf-0 \n");
if (valeur == 0) {fork() ; printf ("printf-0 \n");
                  fork() ;  printf ("printf-0 \n");}

   return 0;}
```

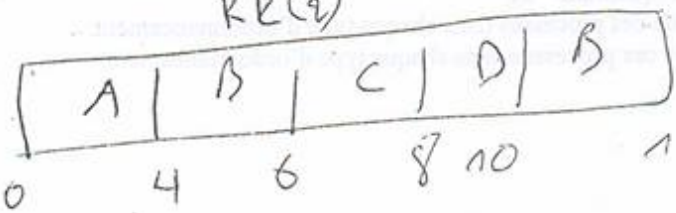
SRT



TAM = 2,5

TRM = 6,25

RR(2)



TAM = 3,5

TRM = 7,25

Universitaire de Bordj Bou Arréridj
Faculté mathématique et Informatique.
Département Informatique.
Durée 1H : 30 de 14:30 à 16:00.

Année universitaire 2013/2014
2ème Année Licence informatique.
Module : **Systèmes d'exploitation 1.**
Le : 17/06/2014
Documents non autorisés.

Examen Rattrapage

Exercice 01 (6 points) (Questions de cours)

1. Donner les objectifs du système d'exploitation?
2. Donner le rôle du vecteur d'interruption? Pourquoi les systèmes d'exploitation utilisant les niveaux d'interruption ?
3. Donner le rôle de l'ordonnanceur ?
4. Quelles sont les politiques de remplacement des pages victimes au niveau de mémoire virtuelle ?
5. Dans les deux cas où le Q<< (Q=Quantum) et Q>> quelle est votre remarque au niveau d'ordonnanceur ?

Exercice 02 (4 points) (l'édition de liens)

On dispose de 5 modules compilés, mémorisés dans les fichiers **cmpdisk.o, cmpfile.o, disk_io.o, lsbrk.o, printstr.o**. Ces 5 modules constituent la base d'un programme, qui nécessite en plus des modules de bibliothèque.

On se propose de faire l'édition des liens des 5 modules dans l'ordre suivant :
cmpdisk.o, cmpfile.o, disk_io.o, lsbrk.o, printstr.o.
1- Définir les adresses d'implantations des modules après la construction de programme final ?
2- Donner le contenu de la table des liens après la construction de programme final ?
3- la taille total de programme

cmpdisk.o:	las	bloc_transfer_dr	
	las	close_printer	
	las	compare_file	
	lu	compare_hierarch	1862
	las	exit	
	lu	exit_prog	2020
	lu	file_error	422
	las	free	
	las	get_memory	
	lu	main	2460
	las	open_drive	
	las	open_printer	
	las	print_string	
cmpfile.o:	las	bloc_transfer_dr	
	lu	compare_file	200
	las	file_error	
disk_io.o:	lu	bloc_transfer_dr	2118
	las	exit	
	las	exit_prog	
	las	get_memory	
	lu	open_drive	236
	las	print_string	
lsbrk.o:	las	exit_prog	
	lu	get_memory	0
	las	lmalloc	
	las	print_string	
printstr.o:	lu	ask_confirm	750
	lu	close_printer	1020
	las	getchar	
	lu	open_printer	996
	lu	print_string	340
	las	write	

cmpdisk.o	3376
cmpfile.o	644
disk_io.o	3862
lsbrk.o	82
printstr.o	1196

table 2. résultat de "taille"

Exercice 03 (5 points) (gestion de processus)

Soit un algorithme d'ordonnancement préemptif basé sur des priorités dynamiques, où une plus grande valeur indique une priorité plus grande. Quand un processus entre dans la file des processus prêts, sa priorité est 0. Tant qu'il reste dans cette file, sa priorité augmente graduellement en fonction du temps avec un taux α. Lorsqu'il passe à l'état d'exécution, sa priorité continue d'augmentera, mais avec un taux différentβ.

1. Indiquez quel est l'algorithme d'ordonnancement obtenu si $0<\alpha<\beta$. S'il d'agit d'un algorithme équivalent à un algorithme déjà vu en cours. Indiquez-le.

- Supposez maintenant que $\alpha>0$ et que la valeur de priorité est une valeur constante $P>0$ lorsqu'un processus passe à l'état d'exécution. Répondez aux questions suivantes :

2. Quel algorithme obtenons-nous dans ce cas ?
3. Décrivez un avantage et un problème important de cette approche.

Exercice 04 (5 points)

1. Donner le nombre des processus avec le schéma.
2. Donner le nombre des instructions printf avec le schéma.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

int main(void) {
            printf ("pere \n");
 int valeur;
 int valeur1;
      valeur = fork();
  if (valeur == 0) { printf ("p=0 \n"); }
  if (valeur > 0) { printf ("p>0 \n");fork();
}

valeur1 =fork();
valeur1 =fork();

if ( valeur1 == 0) {
int valeur1 =fork ();printf ("p1=0 \n"); }
if (valeur1 > 0) {printf ("p1>0 \n"); fork();
}

  return 0;}
```

Bon courage