

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement supérieur et de la recherche scientifique
Université de Jijel
Faculté des Sciences exactes et Informatique
Département d'informatique



Travaux dirigés

SYSTÈMES D'EXPLOITATION

Pour 2^{ème} Année Licence - informatique

Par Mme Bouainah Madiha

Série TD N°1

(Introduction aux systèmes d'exploitation)

Exercice 1 : Choisir la ou les bonnes réponses .

1. A la première génération d'ordinateurs, le système d'exploitation était :
(a) un robot (b) un bonhomme à moustache (c) il n'y avait pas de SE.
2. A la 2nd génération, sont apparues les cartes perforées. Elles servaient :
(a) A la lecture des programmes et des données, donc en entrée de l'ordinateur
(b) A l'écriture des résultats, donc à la sortie de l'ordinateur
(c) Entre les deux.
3. La configuration "Lecteur de carte perforées → ordinateur → imprimante" a posé un problème :
(a) Pendant la lecture des données, le processeur restait inoccupé.
(b) Pendant l'impression des résultats, l'unité centrale restait inoccupée.
(c) L'ordinateur était trop lent par rapport au lecteur et à l'imprimante.
4. Pour résoudre ce problème :
(a) On remplace les cartes perforées par les bandes magnétiques.
(b) On remplace l'imprimante par les bandes magnétiques.
(c) On interpose des lecteurs de bandes magnétiques entre le lecteur de cartes et l'ordinateur et entre celui-ci et l'imprimante.
5. Le problème qui persistait était :
(a) La rapidité des E/S par rapport au processeur.
(b) la lenteur des périphériques d'E/S par rapport au processeur.
(c) Aucun. Tout était parfait.
6. Le premier SE est réellement apparu avec le traitement par lots (Batch processing). A cette époque, un programme s'appelait :
(a) un travail (b) un lot (c) un job
7. Pourquoi a-t-on eu recours au batch processing ?
(a) Pour maximiser le rendement du processeur
(b) Pour minimiser le temps d'attente du processeur entre la fin d'un job et le début du suivant
(c) Pour décharger l'opérateur qui était responsable du chargement manuel des jobs
8. A cette époque, régnait toujours la monoprogrammation qui veut dire :
(a) que le processeur exécute un seul programme à la fois
(b) qu'à un moment donné, il y avait un seul programme en mémoire centrale
(c) que le processeur exécute une seule instruction à la fois
9. A la 3^{ème} génération est apparue la multiprogrammation qui veut dire :
(a) que le processeur exécute plusieurs programmes à la fois
(b) qu'à un moment donné, il y a plusieurs programmes en mémoire centrale
(c) que le processeur exécute plusieurs instructions à la fois
10. Un système temps partagé suppose que :
(a) il y a plusieurs programmes en mémoire centrale

- (b) il y a plusieurs utilisateurs connectés au même ordinateur
- (c) il y a plusieurs ordinateurs.

11. Un système temps partagé est caractérisé par:

- (a) L'allocation de ressources selon un quantum de temps
- (b) L'allocation du CPU selon un quantum de temps .
- (c) Un quantum de temps est alloué à chaque usager.

12. Quelle est la séquence de boot d'un PC

- (a) BIOS, chargement du SE, Post
- (b) Post, BIOS, chargement du SE
- (c) BIOS, Post, chargement du SE

13. Un système multiprogrammé est :

- (a) un système temps partagé
- (b) un système à traitement par lot
- (c) un système dont l'exécution des programmes et les opérations d'E/S se font en parallèle

14. Les OS actuels sont :

- (a) monoprogrammés
- (b) Multiprogrammés
- (c) les deux ?

15. Le lancement du SE est initié par un programme appelé :

- (a) I/O.sys
- (b) BIOS
- (c) Bootstrap

16. Quelle entreprise finance le système d'exploitation open source mobile Android ?

- 1. Microsoft 2. Yahoo 3. Google

17. Une de ces distributions n'est pas une distribution Linux. Laquelle ?

- 1. Solaris 2. Mandriva 3. Ubuntu

18. Un système d'exploitation permet de:

- 1. Transformer le matériel en une machine virtuelle
- 2. Exploiter les ressources CPU au maximum
- 3. Optimiser l'utilisation des ressources (matérielles et logicielles)

19. Unix est un système :

- 1. Multitâche 2. Monotâche 3. Multi-utilisateur 4. Mono-utilisateur
- 5. À temps partagé

20. Quels sont parmi les systèmes suivants des systèmes d'exploitation mobiles :

- 1. Android
- 2. iOS
- 3. Windows NT
- 4. Windows Phone
- 5. Symbian OS
- 6. Mac OS

Série TD N°2

(Historique des systèmes d'exploitation)

Exercice 1 :

Le but de cet exercice est de mettre en évidence, sur un système simplifié à l'extrême, l'influence de l'évolution historique des systèmes d'exploitation sur quelques grandeurs caractéristiques de leurs performances.

On considère un ordinateur dont les organes périphériques sont un lecteur de cartes (1000 cartes / minutes) et une imprimante (1000 lignes / minutes). Un travail moyen est ainsi défini :

- lire 300 cartes,
- utiliser le processeur pendant une minute,
- imprimer 500 lignes.

On suppose que tous les travaux soumis par les usagers ont des caractéristiques identiques à celles de ce travail moyen. On définit deux mesures des performances du système :

- le débit moyen D des travaux : nombre de travaux effectués en une heure.
- le rendement R de l'unité centrale : fraction du temps total d'utilisation de l'unité centrale pendant laquelle exécute du travail utile (autre que la gestion des périphériques).

On suppose d'abord que les périphériques sont gérés par l'unité centrale. calculer R et D dans les hypothèses de fonctionnement suivantes :

- a. Le système est exploité en porte ouverte ; la durée d'une session est limitée à 15 mn. On suppose qu'un usager a besoin de 4 mn pour corriger son programme au vu des résultats, et faire une nouvelle soumission.
- b. Le système est exploité avec un moniteur d'enchaînement séquentiel des travaux.
- c. On suppose maintenant que les périphériques sont gérés par un ordinateur séparé, qui constitue une bande magnétique d'entrée à partir des cartes et liste sur imprimante le contenu d'une bande magnétique de sortie. L'ordinateur est alimenté par la bande magnétique d'entrée et produit la bande de sortie ; on néglige la durée de lecture et d'écriture des bandes. Le temps de transfert des bandes d'un ordinateur à l'autre est de 5 mn dans chaque sens ; on suppose qu'une bande regroupe une fournée de 50 travaux
 1. On suppose que le rythme de soumission des travaux est suffisant pour occuper l'ordinateur central à plein temps. Calculer les valeurs de r et d
 2. Etablir la planification de la construction des trains de travaux et calculer le temps d'attente moyen d'un usager (temps entre la soumission du travail et la réception des résultats. On admettra que les travaux arrivent à un rythme régulier, que le temps de construction d'une fournée (préparation du train de

cartes) est de 10 mn et que le temps de distribution des résultats d'une tournée (découpage et tri des listings) est de 10 mn également.

Exercice 2 :

Sur un système temps partagés, S représente le temps nécessaire pour réaliser une commutation de processus (changement de programme), Q le quantum de temps pour ce système, et R le temps moyen d'exécution d'un processus (programme). Donnez une formule pour connaître l'efficacité du CPU en fonction de ce qui suit :

- $q = \infty$
- $q > r$
- $s < q < r$
- $s = q < R$
- q est proche de 0

Exercice 3 :

On s'intéresse à l'exécution sur une configuration monoprocesseur (processeur, MC, un canal d'E/S) des trois programmes P1, P2, P3 suivants qui arrivent dans l'ordre P1, P2, P3

P1	P2	P3
5 unités de temps CPU 2 unités de temps E/S 3 unités de temps CPU	1 unité de temps CPU 4 unités de temps E/S	1 unité de temps E/S 4 unités de temps CPU

1. en supposant qu'une tâche de contrôle du système s'effectue en une unité de temps quelque soit sa nature, donner le diagramme d'exécution des programmes dans les trois modes d'exploitation mono programme, multiprogrammé et temps partagé avec un quantum de temps égal 2 unités
2. calculer les temps de réponse individuels et moyens dans chaque mode ?

Série TD N°3

(Les interruptions)

Résumé :

Le mot d'état du processeur (mep) est représenté par un enregistrement de la forme : < activité, mode, masquage, compteur ordinal >, où

- activité = attente ou active
- mode = maître (système) ou esclave (utilisateur)
- masquage = masqué (interruption bloquée) ou démasqué
- Si mep désigne un mot d'état, ses champs sont désignés par mep.act, mep.mode, mep.masq et mep.CO.
- La notation Mp[adr] désigne le contenu de la mémoire à l'adresse adr.

Lors d'une commutation de contexte, le mot d'état courant est placé dans un endroit appelé **ancien_mep**, puis sa nouvelle valeur est chargée depuis l'emplacement **nouveau_mep**. Ce dernier contient notamment l'adresse du programme traitant de l'interruption. On dispose de plus d'une instruction **charger_mep(M)** dont l'effet est de charger le mot d'état M.

La notation **adr OBJ** désigne l'adresse de l'objet OBJ.

- ❖ **Sauvegarde dans des emplacements fixes** : à chaque cause ou type d'interruption sont attachés de manière fixe deux emplacements de mémoire : ancien- mep, nouveau- mep . La commutation s'effectue comme suit :

Ancien –mep : < le mot d'état de processeur>(psw)

<mot d'état de processeur> := nouveau-mep

- ❖ **Sauvegarde sur une pile** : ce schéma est identique au précédent pour ce qui concerne le nouveau mot d'état. L'ancien mot d'état, en revanche, n'est plus rangé dans un emplacement fixe, mais sur une pile réservée à cet effet :

Ptr := ptr+1

Pile [ptr]:< mot d'état du processeur>

<mot d'état de processeur> :=nouveau_mep

Exercice 01 : Mesure de la taille d'une mémoire par déroutement.

Un système d'exploitation est en général destiné à être utilisé sur diverses configurations d'une même machine. Le système doit être adapté à la configuration utilisée. La création d'une telle version spécifique est appelée génération du système. Pour réduire le nombre de générations, certains paramètres de la configuration, comme par exemple la quantité de mémoire présente dans l'ordinateur, sont automatiquement déterminés lors de l'initialisation du système. On suppose que la mémoire est constituée de blocs de p mots, numérotés à partir de 0, et que l'accès à une adresse mémoire inexistante provoque un déroutement.

Question : Écrire un algorithme déterminant la taille mémoire d'un système (c'est-à-dire le nombre de blocs).

Exercice 02 :

Un ensemble de programme, ou travaux, doit être exécuté sur un ordinateur, en utilisant le processeur en temps partagés : le processeur est successivement alloué aux travaux, dans un ordre d'arrivée et suivant le quantum Q .

Les travaux sont numérotés de 0 à $n-1$ avec une table de priorité des processus. On définit pour chaque travail i une zone de rangement $mep[i]$ pour son mot d'état et une zone de rangement $reg[i]$ pour ses registres. Ces zones servent à sauvegarder le contexte du processus chaque fois qu'un travail est interrompu, pour permettre la reprise ultérieure de ce travail. Chaque processus entre au système avec un appel système. À la fin de chaque processus, le passage au programme suivant fait en utilisant un mot d'état noté (changement)

Question : Proposez un programme de traitement d'interruption ?

Exercice 03 : *Simulation d'instructions manquantes.*

On souhaite simuler un jeu d'instructions arithmétiques en virgule flottante sur une machine sans coprocesseur mathématique. Sans ce coprocesseur, l'exécution d'une instruction en virgule flottante (div, mult) par la machine provoque un déroutement pour instruction inexistante et l'arrêt du programme avec un message d'erreur.

Question : Écrire les procédures permettant de mettre en place une simulation logicielle de ces instructions en virgule flottante : lorsqu'un programme exécutera une telle instruction, une procédure simulant son exécution (rendant le même résultat) sera automatiquement appelée.

Série TD N°4

(Les Processus)

Exercice 1 : Ecrire un programme qui permet de créer un processus fils en utilisant la fonction fork(). Ajouter la fonction exit(0) pour sortir en cas d'erreur. Ajouter des instructions qui permettent d'afficher l'identifiant de chaque processus.

Exercice 2 : Que fait chacun des programmes suivants :

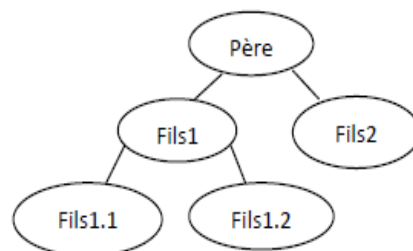
```
int main( )
{
    int p=1 ;
    while(p>0) p=fork() ;
    execlp("prog", "prog", NULL) ;
    return 0 ;
}
```

```
int main ( )
{ int p=1 ;
  for(int i=0 ; i<=4 ; i++)
  if (p>0) p=fork( ) ;
  if(p !=-1) execlp("prog", "prog",
  NULL) ;
  else exit(1) ;
  while(wait(NULL) !=-1) ;
  return 0 ;
}
```

```
int i=2 ;
int main ( )
{
    j=10;
    int p ;
    while(i-- && p = fork())
    if(p<0) exit(1) ;

    j += 2;
    if (p == 0)
    { i *= 3;
      j *= 3;
    }
    else
    {
        i *= 2;
        j *= 2;
    }
    printf(« i=%d, j=%d », i,j) ;
    return 0 ;
}
```

Exercice 3 : Réaliser le programme correspondant à l'arborescence suivante : chaque processus devra afficher son pid et le pid de son parent.



Série TD N°5

(Gestion des processus)

Exercice 1 :

On considère trois (3) processus A, T, S :

1. A : processus d'affichage
2. T : processus de transmission
3. S : processus de sortie

• **Sachant que :**

- ✓ Le processus Afficheur (A) fait une entrée/sortie après chaque 3 unités de temps (de calcul), les autres n'en font aucune.
- ✓ Chaque entrée/sortie dure une unité de temps.

Processus	Temps d'arrivée	Temps de traitement
A	0	15
T	2	6
S	8	7

- Voici Un diagramme de Gantt représente une politique d'ordonnancement des processus. Chaque case du diagramme correspond à une unité de temps.

A	A	A	T	T	T	A	A	A	T	T	T	S	S	S	A	A	A	S	S	S	A	A	A	S	A	A	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Questions :

1. Identifiez la politique d'ordonnancement représenté par le digramme de Gantt ?
2. Donnez le diagramme de Gantt de l'exécution de ces processus suivant la politique FCFS ?
3. Donnez le diagramme de Gantt de l'exécution de ces processus suivant la politique SRTF (SJF avec réquisition)
4. Calculez pour tous ces algorithmes (inclut la politique non définit) : le temps d'attente, le temps de réponse, le temps de séjour (d'exécution) ?
5. Quel est l'algorithme qui donne un meilleur temps d'attente ?

Exercice 2 :

Processus	Exécution	Arrivée
A	3	0
B	6	1
C	4	4
D	2	6
E	1	7

1- Donnez l'ordre d'exécution des processus (diagramme de Gantt)

2- Calculez le temps moyen de séjour ,le temps d'attente moyen.

Pour les politiques d'ordonnancement suivantes :

1) FIFO (First In First Out),2) SJF (Short Job First),3) SRTF (Shortest Remaining Time First)

4) Ordonnancement circulaire avec : Quantum=1

Exercice 3 :

Cinq travaux A, B, C, D et E arrivent pratiquement en même temps dans un centre de calcul. Leur temps d'exécution respectif est estimé à 10, 6, 2, 4 et 8 secondes.

Tracez le diagramme de Gantt et déterminez le temps moyen de rotation pour chacun des algorithmes d'ordonnancement suivants. Ne tenez pas compte du temps perdu lors de la commutation des processus.

- Premier arrivé, premier servi FCFS (exécution dans l'ordre 10, 6, 2, 4, 8) ;
- Plus court d'abord SJF ;
- Tourniquet (quantum q = 4 s).

Exercice 4 : On considère l'ensemble des processus suivants :

N° Processus	Date d'arrivée	Temps CPU	Priorité
1	7h 00	10 mn	2
2	7h 00	15 mn	3
3	7h 03	8 mn	4
4	7h 10	18 mn	5

A- On suppose qu'on utilise un algorithme d'ordonnancement basé sur la priorité (les priorités sont croissantes : 5 est le plus prioritaire). Donnez le diagramme de Gantt pour les priorités données dans le tableau.

B- On voudrait que la priorité des processus soit dynamique au cours du temps. Ainsi, pour calculer la priorité d'un processus, on utilise la formule suivante :

$$\text{Priorité} = \frac{\text{Temps d'attente} + \text{Temps CPU Restant}}{\text{Temps CPU}}$$

Remarque : Lors des calculs, on arrondira suivant l'exemple suivant : 3.5 ou 3.6 ->4, 3.1 ou 3.4 ->3.

1. Donnez le diagramme de Gantt sachant que la priorité est recalculée toutes les 5 minutes.
2. Calculez le temps d'attente moyen ainsi que le temps de rotation moyen.
3. Comparez les résultats obtenus par rapport à ceux obtenus avec l'algorithme de priorité classique.

Série TD N°6

(Gestion de la mémoire)

Exercice 1 :

Dans un système paginé, les pages font 256 mots mémoire et on autorise chaque processus à utiliser au plus 4 cadres de la mémoire centrale. On considère la table des pages suivante du processus P1 :

page	cadre
0	3
1	i
2	0
3	i
4	i
5	i
6	5
7	1

i signifiant que la page n'est pas chargée.

- Quelle est la taille de l'espace d'adressage du processus P1 (à quelques mots près) ?
- Calculer les adresses réelles correspondant aux adresses virtuelles suivantes émises par P1 au cours de son exécution : 240, 546, 1578, 2072
- Supposons que P1 continue son exécution en générant comme adresse virtuelle : 770. Que se passera-t-il ?

Partie B :

Un programme a un espace virtuel de 600 mots. On considère la suite des adresses virtuelles : 34, 123, 145, 510, 456, 345, 412, 10, 14, 12, 234, 336, 412.

- Donner la suite des numéros de pages référencés, sachant qu'elles comportent 100 mots.
- Le programme dispose de 300 mots en mémoire centrale. Calculer le taux de défauts de page(en supposant la mémoire initialement vide) pour les algorithmes FIFO (First Page In, First Page Out), LRU (Least RecentlyUsed),

Exercice 02 : (segmentation)

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

segment	base	longueur
0	540	234
1	1254	128
2	54	328
3	2048	1024
4	976	200

Calculer les adresses réelles correspondant aux adresses virtuelles suivantes, ou signaler les éventuelles erreurs d'adressage qui engendreront un déroutement puis le postage à P1 d'un signal du type *segmentation fault*:

(0, 128), (1, 100), (2, 465), (3, 888), (4, 100), (4, 344)

Exercice 3 : On considère un système dont l'espace mémoire usager compte 1MO. On choisit la multiprogrammation à partitions variables pour ce système. On suppose la chronologie suivante pour notre système

Instant t	Événement
$t = 0$	A (300, 55)
$t = 10$	B (400, 35)
$t = 30$	C (500, 35)
$t = 40$	D (300, 105)
$t = 50$	E (200, 35)
$t = 60$	F (100, 55)
$t = 70$	G (400, 35)
$t = 90$	H (700, 35)
$t = 110$	I (200, 25)
$t = 120$	J (400, 45)

Un processus qui ne peut pas être chargé en mémoire est placé sur une file d'attente gérée par une politique FIFO. La première ligne du tableau signifie que: le processus A arrive à l'instant 0, la taille de son espace logique est de 300 K et lorsque le processus A est chargé en mémoire centrale, il y séjournera (en mémoire centrale) pendant exactement 55 unités de temps.

Donnez les états successifs d'occupation et de libération de la mémoire si le mode d'allocation utilise l'algorithme premier ajustement (*First Fit*).

Exercice 4 : Soit un système à pagination ayant des pages de 100 octets

- Donner la suite des pages correspondant aux adresses suivantes : 34 , 145 , 10 , 236
- Proposer une formule de calcul