

# Systèmes d'exploitation

Sans documents – Durée 1h30  
Justifiez les réponses

## 1 Questions de cours

1. Décrivez en quelques lignes ce qui se passe lors d'un changement de contexte.
2. En quoi consiste la virtualisation ? Quel(s) intérêt(s) y voyez-vous ?

## 2 RAID

On dispose de 4 disques durs de 500 Go chacun et on souhaite agréger ces disques pour faire du RAID.

1 ) Quelle sera la capacité de stockage si ces disques sont assemblés en RAID0, en RAID1, ou en RAID5 ?

2 ) Quels sont les avantages de ses différentes techniques ?

3 ) On suppose qu'un fichier occupe les 100 premiers blocs de l'espace disque virtuel. Pour chaque solution, combien faut-il de temps pour lire ces 100 blocs sachant que le temps d'accès en lecture (ou en écriture) d'un bloc est de 5 ms ? Justifiez pour chacun des cas. Même question pour accéder aux blocs en écriture.

## 3 Ordonnancement

On considère une machine sur laquelle vont tourner les 5 processus A, B, C, D et E dont les durées d'exécution et les dates d'arrivée sont les suivantes :

processus	Date d'arrivée	Durée
A	0	10
B	2	4
C	1	7
D	4	12
E	3	5

On appelle temps de service (TS) d'un processus le temps écoulé entre son arrivée dans le système et sa terminaison.

Représentez le déroulement de l'exécution des processus par un diagramme et calculez le temps de service moyen pour les algorithmes d'ordonnancement suivant :

1. Ordonnancement FIFO sans quantum de temps
2. Ordonnancement suivant le plus court temps d'exécution (PCTE), en tenant compte des dates d'arrivée
3. Ordonnancement suivant la règle dite du "Tourniquet" (Round Robin) avec quantum de temps  $q=2$  en tenant compte des dates d'arrivée.

## 4 Principe de localité

Considérons le programme suivant, qui calcule la somme de deux tableaux de 100 éléments :

```
pour i = 1 jusqu'à 100 faire
    C[i] = A[i] + B[i];
fin;
```

Supposons que ce programme soit exécuté avec des pages de taille 1000 mots et que son équivalent en langage machine soit rangé en mémoire virtuelle à l'adresse 2000 (R1, R2 et R3 désignent les registres du processeur) :

```
2000  charger 1      dans R1
2001  charger 100 dans R2
2002  comparer R1 et R2
2003  si R1 plus grand sauter en 2009
2004  charger A[R1] dans R3
2005  ajouter B[R1] à R3
2006  ranger (stocker) R3 dans C[R1]
2007  incrémenter R1
2008  sauter en 2002
2009  (suite)
```

Supposons que les adresses des variables soient les suivantes :

```
6000-6099 : tableau A
7000-7099 : tableau B
8000-8099 : tableau C
9000      : constante 1
9001      : constante 100
```

Le programme est situé à l'adresse 2000 et nécessite le chargement en mémoire de la page **2**. L'exécution de la première instruction (à l'adresse 2000) nécessite le chargement de la page **9** (constante 1). L'instruction suivante, située sur la page 2 (adresse 2001) utilise la page 9 (constante 100). etc.

1. Compléter la formule des pages référencées produite par l'exécution de ce programme :  
 $2, 9, 2, 9, 2, (2, 2, 6, 2, 7\dots)^{100}$
2. Quelle est la page la plus référencée ? On suppose que la machine est très chargée et a tendance à “*swapper*” ; comment s'y prendre pour accélérer considérablement l'exécution ?

## 5 Mémoire partitionnée

Soient des partitions “libres” de mémoire de tailles 100K, 500K, 200K, 300K et 600K (dans cet ordre).

Comment placer des processus de tailles 212K, 417K, 112K et 426K (dans cet ordre) avec l'algorithme :

- First-fit (première zone libre) ?
- Best-fit (gaspillage minimum) ?
- Worst-fit (gaspillage maximum) ?

Quel algorithme fournit ici le meilleur usage de la mémoire ?