Master 1 IMIS
Systèmes d'exploitation

Université d'Orléans
2018-2019

Contrôle continu

Durée 1h30. Une feuille A4 recto-verso manuscrite (cours et/ou td) autorisée.

Nom et prénom : **Groupe :**

Exercice 1 (8 points)
On dispose d'un système dont la mémoire centrale est organisée en pages de taille 4 Koctets et où l'algorithme utilisé en cas de défaut de page est l'algorithme LRU (Least Recently Used).

On propose un programme `pg` qui manipule une matrice `t` de taille fixe 4096×1024 ; les éléments de `t` sont des réels, de type double. On rappelle qu'un réel occupe 8 octets en mémoire. Le programme comporte trois phases ; dans chacune des deux premières phases, on calcule les éléments d'une partie de la matrice, ligne par ligne, par une double boucle :

```
for (i = 0; i < hauteur; i++)
   for (j = 0; j < largeur; j++)
     t[i][j] = f (i, j);
```

Dans la troisième phase, le calcul est effectué colonne par colonne :

```
  for (j = 0; j < largeur; j++)
    for (i = 0; i < hauteur; i++)
      t[i][j] = f (i, j);
```

`f` est une fonction de calcul simple, dont la nature exacte est sans importance ; `hauteur` et `largeur` sont, dans cet ordre, les arguments du programme. On mesure, pour chaque phase, trois temps : le temps total d'exécution, le temps passé par le processus en mode utilisateur, et le temps passé par le processus en mode système (on utilise pour cela l'appel système `time`). Le temps CPU est désigné par les deux derniers temps. Les temps sont fournis en *tics* d'horloge, sachant que la machine utilisée a une horloge qui en effectue 100 par seconde.

Il n'est pas utile pour la suite de vous donner le programme complet. Il est cependant conseillé de lire le sujet en entier avant de répondre aux questions.

Q1 : Expliquer la différence, pour un processus, entre mode utilisateur et mode noyau.

Voici une première exécution de ce programme, qui remplit 1000 lignes et 200 colonnes de la matrice. Chaque temps est affiché en centièmes de secondes sous la forme $t = t_1 + t_2 + t_3$, où t_i concerne la phase i :

```
MacBook-Pro-de-Wadoud:~ bousdira$ pg 1000 200
Processus 11241 : temps écoulé       305 =  201 +   49 +   55
Processus 11241 : temps utilisateur  160 =   58 +   47 +   55
Processus 11241 : temps système      119 =  119 +    0 +    0
```

Q2 : Expliquer pourquoi le processus passe un temps important en mode système pendant la première phase, et pas pendant les deux suivantes.

Q3 : Calculer l'ordre de grandeur du temps de calcul d'un élément de la matrice.

Voici une seconde exécution de ce programme, qui remplit 2000 lignes et 100 colonnes de la matrice :

```
MacBook-Pro-de-Wadoud:~ bousdira$ pg 2000 100
Processus 11242 : temps ecoule     11122 =  834 + 4825 + 5463
Processus 11242 : temps utilisateur  207 =   66 +   64 +   77
Processus 11242 : temps systeme      872 =  256 +  329 +  287
```

Q4 : Expliquer pourquoi le temps passé en mode système pendant la première phase a approximativement doublé (par rapport à la première exécution), alors que le temps passé en mode utilisateur n'a pas varié de façon significative.

Q5 : Expliquer pourquoi le processus passe maintenant un temps comparable en mode système pendant chacune des phases.

Q6 : Expliquer pourquoi le temps réel écoulé, surtout pendant les deux dernières phases, est très supérieur au temps CPU (aucun autre processus-utilisateur n'a évidemment perturbé l'exécution).

Exercice 2 (6 points)
On considère une mémoire paginée. Soit la chaîne de références suivante : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

Q1 : Calculer le nombre de défauts de pages, si on utilise un nombre de cadres de pages égal à 3, pour l'algorithme FIFO, puis l'algorithme LRU.

Algorithme FIFO :

1	2	3	4	1	2	5	1	2	3	4	5

Algorithme LRU :

1	2	3	4	1	2	5	1	2	3	4	5

Q2 : Calculer le nombre de défauts de pages, si on utilise un nombre de cadres de pages égal à 4, pour l'algorithme FIFO, puis l'algorithme LRU.

Algorithme FIFO :

1	2	3	4	1	2	5	1	2	3	4	5

Algorithme LRU :

1	2	3	4	1	2	5	1	2	3	4	5

Q3 : Discuter les résultats obtenus aux deux questions précédentes.

Exercice 3 (6 points)
Le comptage du temps sur le PC est effectué via une interruption matérielle IRQ 0 du circuit 8259 (timer). IRQ 0 est déclenchée 18 fois par seconde.

Q1 : Écrire, en langage algorithmique, une routine d'interruption de l'IRQ 0 qui permet d'afficher l'heure courante (en Heure, Minute, Seconde) sous la forme `HH:MM:SS`.
Indication : vous supposerez que toutes les variables utilisées sont correctement initialisées. Utilisez une variable pour compter le nombre d'interruptions.

Q2 : Que se passe-t-il si l'exécution de la routine d'interruption dure plus de 0,06 seconde ?

Q3 : Que peut-on proposer dans ce cas ?