

Systèmes d'exploitation et Processus : vers le BAC

Exercice 1 :

La commande UNIX « ps » présente un cliché instantané des processus en cours d'exécution. Avec l'option « -eo pid,ppid,stat,command », cette commande affiche dans l'ordre : l'identifiant du processus PID (process identifier), son PPID, l'état (« STAT ») et le nom de la commande à l'origine du processus.

Les valeurs du champ STAT indiquent l'état des processus :

- R : processus en cours d'exécution
- S : processus endormi

Sur un ordinateur, on exécute la commande « ps -eo pid,ppid,stat,command » et on obtient un affichage dont on donne ci-dessous un extrait.

```
$ ps -eo pid,ppid,stat,command

PID  PPID  STAT  COMMAND
1    0     Ss    /sbin/init
....  ....  ..    ...
1912 1908  Ss    Bash
2014 1912  Ss    Bash
1920 1747  Sl    Gedit
2013 1912  Ss    Bash
2091 1593  Sl    /usr/lib/firefox/firefox
5437 1912  Sl    python programme1.py
5440 2013  R     python programme2.py
5450 1912  R+    ps -eo pid,ppid,stat,command
```

1. Qu'est-ce que le PPID d'un processus ?

Pour répondre aux questions suivantes, il faut se référer à l'affichage ci-dessus.

2. Quel est le nom de la première commande exécutée par le système d'exploitation lors du démarrage ? Justifier la réponse.
3. Quels sont les identifiants des processus en cours d'exécution sur cet ordinateur au moment de l'appel de la commande « ps » ? Justifier la réponse.
4. Depuis quelle application a-t-on exécuté la commande ps ? Justifier la réponse.
5. Quelles sont les autres commandes qui ont été lancées à partir de cette application ? Justifier la réponse.

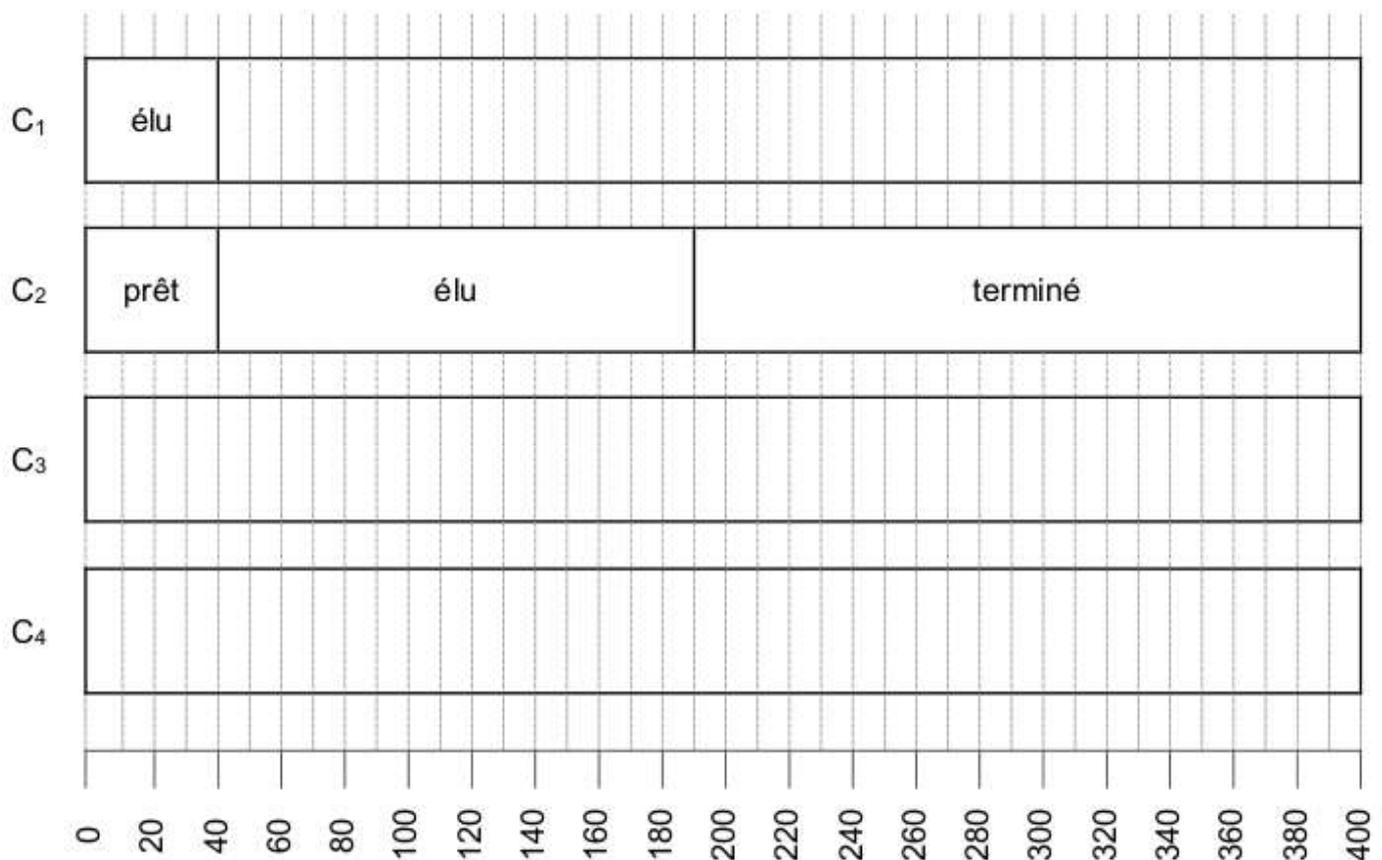
Exercice 2 :

1. Les états possibles d'un processus sont : *prêt, élu, terminé* et *bloqué*.
 - a. Expliquer à quoi correspond l'état élu.
 - b. Proposer un schéma illustrant les passages entre les différents états.
2. On suppose que quatre processus C_1 , C_2 , C_3 et C_4 arrivent dans la « file d'attente » du noyau dans cet ordre.

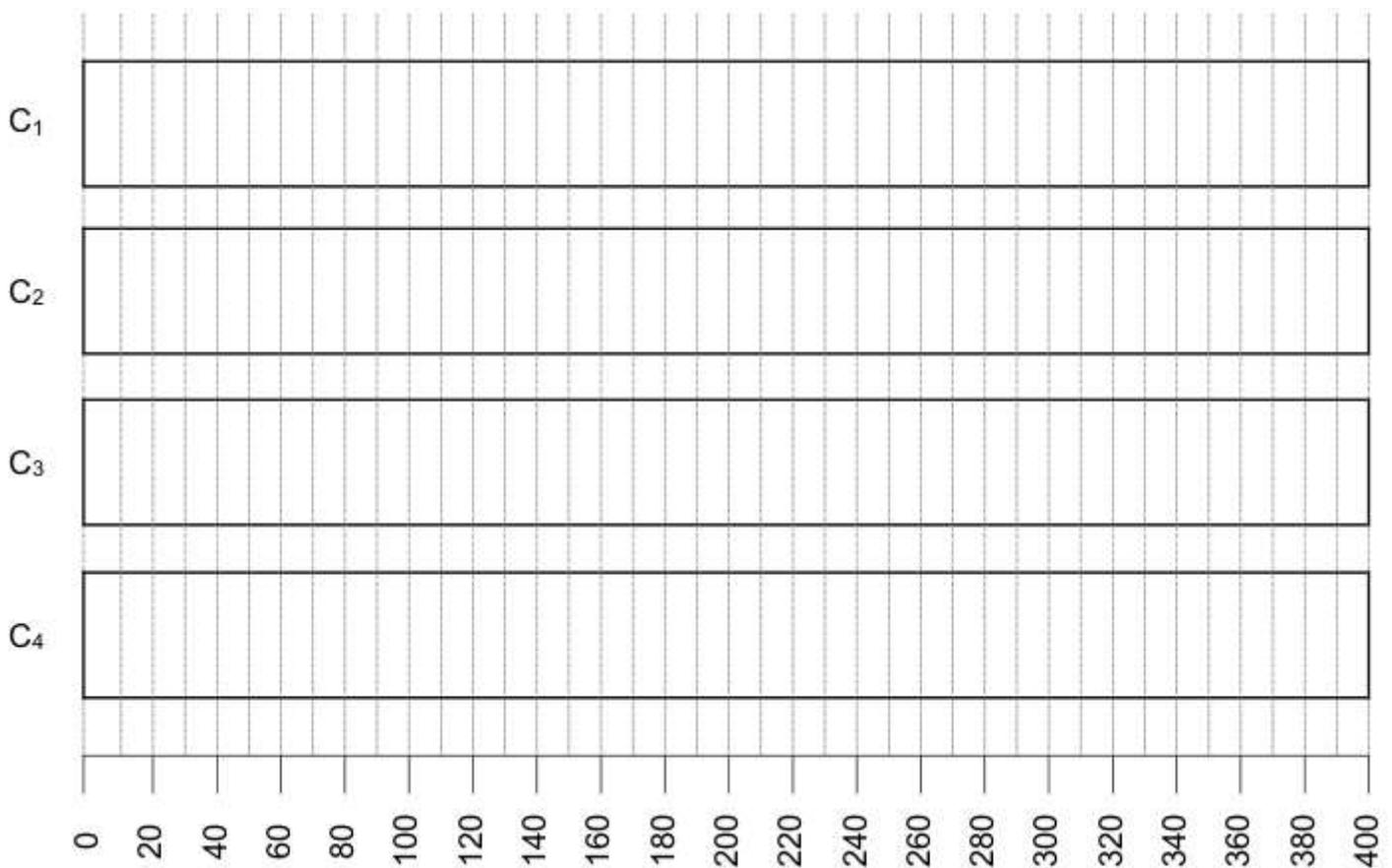
On sait que :

- Le noyau applique la politique d'ordonnancement FIFO (*First In, First Out*, c'est-à-dire premier processus arrivé, premier processus servi)
- Le processeur contient un seul cœur qui exécute un seul processus à la fois.
- Les temps d'exécution totaux de C_1 , C_2 , C_3 et C_4 sont respectivement 100 ms, 150 ms, 80 ms et 60 ms.
- Après 40 ms d'exécution, le processus C_1 demande une opération d'écriture disque, opération qui dure 200 ms. Pendant cette opération d'écriture, le processus C_1 passe à l'état bloqué.
- Après 20 ms d'exécution, le processus C_3 demande une opération d'écriture disque, opération qui dure 10 ms. Pendant cette opération d'écriture, le processus C_3 passe à l'état bloqué.

Sur la frise chronologique ci-dessous, les états du processus C_2 sont donnés. Compléter la frise avec les états des processus C_1 , C_3 et C_4 .



3. Reprendre la question 2, mais cette fois-ci, le noyau stoppe et renvoie dans la file d'attente un processus encore en cours d'exécution au bout de 30 secondes.



Exercice 3 :

Les parties A et B peuvent être traitées indépendamment.

A. Ordonnement des processus

Dans le laboratoire d'analyse médicale d'un hôpital, plusieurs processus peuvent demander l'allocation du processeur simultanément.

Le tableau ci-dessous donne les demandes d'exécution de quatre processus et indique :

- le temps d'exécution du processus (en unité de temps) ;
- l'instant d'arrivée du processus sur le processeur (en unité de temps) ;
- le numéro de priorité du processus (classé de 1 à 10).

Plus la priorité est grande plus le numéro de priorité est petit. Ainsi le processus P3, du tableau ci-dessous, est plus prioritaire que le processus P1.

L'ordonnement est de type préemptif, ce qui signifie qu'à chaque unité de temps, le processeur choisit d'exécuter le processus ayant le plus petit numéro de priorité (un seul processus à la fois). Ceci peut provoquer la suspension d'un autre processus qui reprendra lorsqu'il deviendra le plus prioritaire dans la file d'attente.

Processus	Temps d'exécution	Instant d'arrivée	Numéro de priorité
P1	3	0	4
P2	4	2	2
P3	3	3	1
P4	4	5	3

1. Reproduire le diagramme ci-dessous, sur votre copie, et indiquer dans chacune des cases le processus exécuté par le processeur entre deux unités de temps (il peut y avoir des cases vides).

P1														
0	1													

- Recopier et compléter les temps de séjour ainsi que les temps d'attente de chacun des processus (toujours en unités de temps).

$$\text{Temps de séjour} = \text{instant de terminaison} - \text{instant d'arrivée}$$

$$\text{Temps d'attente} = \text{temps de séjour} - \text{temps d'exécution}$$

Processus	Temps d'exécution	Instant d'arrivée	Temps de séjour	Temps d'attente
P1	3	0	14-0=14	14-3=11
P2	4	2		
P3	3	3		
P4	4	5		

- À quelles conditions le temps d'attente d'un processus peut-il être nul ?

B. Processus et ressources

Dans ce laboratoire d'analyse médicale de l'hôpital, le laborantin en charge du traitement des différents prélèvements (sanguins, urinaires et biopsiques) utilise simultanément quatre logiciels :

- Logiciel d'analyse d'échantillons (connecté à l'analyseur)
- Logiciel d'accès à la base de données des patients (SGBD)
- Traitement de texte
- Tableur

Le tableau ci-dessous donne l'état à un instant donné des différents processus (instances des programmes) qui peuvent soit mobiliser (M) des données (D1, D2, D3, D4 et D5), soit être en attente des données (A) ou ne pas les solliciter (-).

Une donnée ne peut être mobilisée que par un seul processus à la fois. Si un autre processus demande une donnée déjà mobilisée, il passe en attente.

Exemple : le SGBD mobilise la donnée D4 et est en attente de la donnée D5

	D1	D2	D3	D4	D5
Analyseur échantillon	M	-	-	A	-
SGBD	-	-	-	M	A
Traitement de texte	-	M	A		-
Tableur	A	-	M	-	M

- À partir du tableau ci-dessus, démontrer que, à cet instant, les processus s'attendent mutuellement.
- Comment s'appelle cette situation ?
- On suppose que l'analyseur d'échantillon libère la ressource D1. Donner un ordre possible d'exécution des processus.