

Exercice 2 (3 points)

*Cet exercice porte sur les processus, l'ordonnancement, la logique booléenne.
Il est constitué de trois parties indépendantes.*

Partie A : processus

La ligne de commande `ps` tapée dans un terminal permet d'avoir la liste des processus d'un système.

La commande `ps -eo user,pid,ppid,time,cmd` permet d'afficher pour tous les processus du système les colonnes suivantes :

USER : le nom de l'utilisateur qui exécute le processus

PID : l'identifiant du processus

PPID : l'identifiant du processus parent

TIME : le temps d'utilisation du processeur par le processus

CMD : la commande ou l'application, à l'origine de la création du processus

Voici un extrait de l'écran d'un terminal après exécution de la commande :

```
ps -eo user, pid, ppid, time, cmd
```

```
USER      PID    PPID  TIME      CMD
root       1       0  00:00:02  /sbin/init
kernoops  1567    1  00:00:00  /usr/sbin/kerneloops
user01    1611    1  00:00:00  /lib/systemd/systemd --user
user01    1752   1611  00:00:00  /usr/libexec/gnome-session-binary --systemd-
user01    1766   1752  00:00:00  /usr/libexec/at-spi-bus-launcher --launch-im
user01    1770   1611  00:00:51  /usr/bin/gnome-shell
user01    5151   1632  00:00:00  /usr/libexec/gvfsd-network --spawner :1.2 /o
user01    5168   1632  00:00:00  /usr/libexec/gvfsd-dnssd --sp
user01    5348   1770  00:00:01  /snap/vlc/2344/usr/bin/vlc
user01    5468   1770  00:00:01  /snap/gimp/393/usr/bin/gimp
user01    5564   1611  00:00:00  /usr/bin/python3 /usr/bin/gnome-terminal --w
user01    5566   5564  00:00:00  /usr/bin/gnome-terminal.real --wait
user01    5571   1611  00:00:00  /usr/libexec/gnome-terminal-server
user01    5589   5571  00:00:00  bash
user01    5596   5589  00:00:01  /usr/bin/python3 /home/user01/.local/bin/tho
user01    5603   5596  00:00:00  /usr/bin/python3 -u -B -m thonny.plugins.cpy
user01    5615   1611  00:00:00  /usr/bin/python3 /usr/bin/gnome-terminal --w
user01    5617   5615  00:00:00  /usr/bin/gnome-terminal.real --wait
user01    5622   5571  00:00:00  bash
user01    5629   5622  00:00:00  ps -eo user,pid,ppid,time,cmd
```

En utilisant les données de l'extrait ci-avant, répondre aux questions suivantes :

1. Déterminer le nom de l'application associée au processus dont l'identifiant est 5468.
2. Déterminer l'identifiant du processus qui a sollicité le processeur le plus longtemps.
3. Déterminer l'identifiant du processus qui a le plus d'enfants.
4. Déterminer le nom du programme dont est issue la commande `ps`.
5. Déterminer la succession des identifiants des processus qui ont permis de générer le processus associé à la ligne de commande : `ps -eo user,pid,ppid,time,cmd` en partant du processus initial dont le PID est 1.

Partie B : ordonnancement

Dans cet exercice, étant donné un instant initial noté $t_0 = 0$, on dit qu'un processus est caractérisé par :

- sa durée d'exécution, exprimée en unités de temps ;
- son instant d'arrivée, défini à partir de t_0 , correspondant à l'instant où le processus est créé par le système d'exploitation.

6. Les trois états principaux d'un processus sont « prêt », « bloqué » et « élu ». Donner la définition de chacun de ces trois états.

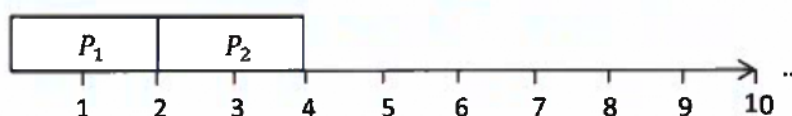
On considère un système d'exploitation qui utilise un ordonnancement par tourniquet pour gérer les processus. Dans cet ordonnancement, le processus élu dispose alors d'un temps donné prédéfini appelé quantum, et s'exécute :

- soit jusqu'à ce qu'il soit terminé (durée d'exécution restante inférieure ou égale au quantum),
- soit pendant la durée du quantum (il retourne ensuite à l'état « prêt » et réintègre la file d'attente de l'ordonnanceur en dernière position),
- soit jusqu'à ce qu'il se bloque de lui-même en raison d'une ressource indisponible.

7. On s'intéresse à la situation suivante mettant en jeu trois processus avec un quantum fixé à 2 unités de temps.

Nom du processus	Temps d'exécution	Instant d'arrivée
P_1	5	0
P_2	3	1
P_3	4	5

Recopier et compléter le chronogramme d'exécution des processus.



Partie C : logique booléenne

On donne ci-dessous les tables de vérité des opérateurs NON, ET et OU, dans lesquelles les variables a et b booléennes prennent les valeurs 0 pour FAUX et 1 pour VRAI.

NON	
a	NON a
0	1
1	0

ET		
a	b	a ET b
0	0	0
0	1	0
1	0	0
1	1	1

OU		
a	b	a OU b
0	0	0
0	1	1
1	0	1
1	1	1

8. Déterminer la table de vérité de l'expression booléenne $\text{NON}(a \text{ ET } b)$.

9. À l'aide d'une deuxième table de vérité, justifier que :

$$\text{NON}(a \text{ ET } b) = (\text{NON } a) \text{ OU } (\text{NON } b)$$