Sujet BAC 13 : Gestion Processus

1. 2021, France sujet 1

Exo

Partie A

Cette partie est un questionnaire à choix multiples (QCM). Pour chacune des questions, une seule des quatre réponses est exacte. Le candidat indiquera sur sa copie le numéro de la question et la lettre correspondant à la réponse exacte. Aucune justification n'est demandée. Une réponse fausse ou une absence de réponse n'enlève aucun point.

Question 1

Parmi les commandes ci-dessous, laquelle permet d'afficher les processus en cours d'exécution ?

Texte

```
a. dir
b. ps
c. man
d. ls
```

Question 2

Quelle abréviation désigne l'identifiant d'un processus dans un système d'exploitation de type UNIX ?

Texte

```
a. PIX
b. SIG
c. PID
d. SID
```

Question 3

Comment s’appelle la gestion du partage du processeur entre différents processus ?

Texte

```
a. L’interblocage
b. L’ordonnancement
c. La planification
d. La priorisation
```

Question 4

Quelle commande permet d’interrompre un processus dans un système d’exploitation de type UNIX ?

Texte

```
a. stop
b. interrupt
c. end
d. kill
```

Correction

Partie B

Question 1

Un processeur choisit à chaque cycle d'exécution le processus qui doit être exécuté. Le tableau ci-dessous donne pour trois processus P1, P2, P3 :

- la durée d'exécution (en nombre de cycles),
- l'instant d'arrivée sur le processeur (exprimé en nombre de cycles à partir de 0),
- le numéro de priorité.

Le numéro de priorité est d'autant plus petit que la priorité est grande. On suppose qu'à chaque instant, c'est le processus qui a le plus petit numéro de priorité qui est exécuté, ce qui peut provoquer la suspension d'un autre processus, lequel reprendra lorsqu'il sera le plus prioritaire.

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

Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus exécuté à chaque cycle.

Correction

Question 1

On suppose maintenant que les trois processus précédents s'exécutent et utilisent une ou plusieurs ressources parmi R1, R2 et R3. Parmi les scénarios suivants, lequel provoque un interblocage ? Justifier.

Scénario 1
P1 acquiert R1
P2 acquiert R2
P3 attend R1
P2 libère R2
P2 attend R1
P1 libère R1

Scénario 2
P1 acquiert R1
P2 acquiert R3
P3 acquiert R2
P1 attend R2
P2 libère R3
P3 attend R1

Scénario 3
P1 acquiert R1
P2 acquiert R2
P3 attend R2
P1 attend R2
P2 libère R2
P3 acquiert R2

Correction

Partie C

Dans cette partie, pour une meilleure lisibilité, des espaces sont placées dans les écritures binaires des nombres. Il ne faut pas les prendre en compte dans les calculs.
Pour chiffrer un message, une méthode, dite du masque jetable, consiste à le combiner avec une chaîne de caractères de longueur comparable.
Une implémentation possible utilise l'opérateur XOR (ou exclusif) dont voici la table de vérité :

a	b	a XOR b
0	0	0
0	1	1
1	0	1
1	1	0

Dans la suite, les nombres écrits en binaire seront précédés du préfixe 0b.

Question 1

Pour chiffrer un message, on convertit chacun de ses caractères en binaire (à l'aide du format Unicode), et on réalise l'opération XOR bit à bit avec la clé.
Après conversion en binaire, et avant que l'opération XOR bit à bit avec la clé n'ait été effectuée, Alice obtient le message suivant :

```
m = 0b 0110 0011 0100 0110
```

a. Le message m correspond à deux caractères codés chacun sur 8 bits : déterminer quels sont ces caractères. On fournit pour cela la table ci-dessous qui associe à l'écriture hexadécimale d'un octet le caractère correspondant (figure 2). Exemple de lecture : le caractère correspondant à l'octet codé 4A en hexadécimal est la lettre J.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	space	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Correction

Question 1.b

Pour chiffrer le message d'Alice, on réalise l'opération XOR bit à bit avec la clé suivante :

```
k = 0b 1110 1110 1111 0000
```

Donner l'écriture binaire du message obtenu.

Correction

Question 2.a

Dresser la table de vérité de l'expression booléenne suivante :

```
(a XOR b) XOR b
```

Correction

Question 2.b

Bob connaît la chaîne de caractères utilisée par Alice pour chiffrer le message. Quelle opération doit-il réaliser pour déchiffrer son message ?

Correction

2. 2021, France sujet annulé

d'après le sujet du bac NSI 2021

Cet exercice porte sur les systèmes d'exploitation : gestion des processus et des ressources.
Les parties A et B peuvent être traitées indépendamment

Partie A

Question 1

La commande ps suivie éventuellement de diverses options permet de lister les processus actifs ou en attente sur une machine. Sur une machine équipée du système d'exploitation GNU/Linux, la commande "ps -aef" permet d'obtenir la sortie suivante (extrait) :

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	10:01	?	00:00:02	/sbin/init splash
root	4	2	0	10:01	?	00:00:00	[kworker/0:0H]
root	6	2	0	10:01	?	00:00:00	[mm_percpu_wq]
....	....	.....	....	.....	.....	.....	.....
bob	3383	1	0	10:25	?	00:00:00	sh -c /usr/bin/google-chrome-sta

a) Quelle est la particularité de l'utilisateur "root" ?

b) Quel est le processus parent du processus ayant pour PID 3383

Correction

Dans un bureau d'architectes, on dispose de certaines ressources qui ne peuvent être utilisées simultanément par plus d'un processus, comme l'imprimante, la table traçante, le modem. Chaque programme, lorsqu'il s'exécute, demande l'allocation des ressources qui lui sont nécessaires. Lorsqu'il a fini de s'exécuter, il libère ses ressources.

Programme 1	Programme 2	Programme 3
demander(table traçante) demander (modem) exécution libérer (modem) libérer (table traçante)	demander (modem) demander (imprimante) exécution libérer (imprimante) libérer (modem)	demander (imprimante) demander (table traçante) exécution libérer (table traçante) libérer (imprimante)

Question 2

On appelle p1, p2 et p3 les processus associés respectivement aux programmes 1, 2 et 3.

a) Justifier qu'une situation d'interblocage peut se produire.

b) Modifier l'ordre des instructions du programme 3 pour qu'une telle situation ne puisse pas se produire.

Correction

Question 3

Supposons que le processus p1 demande la table traçante alors qu'elle est en cours d'utilisation par le processus p3. Parmi les états suivants, quel sera l'état du processus p1 tant que la table traçante n'est pas disponible :

a) élu
b) bloqué
c) prêt
d) terminé

Correction

Partie B

Avec une ligne de commande dans un terminal sous Linux, on obtient l'affichage suivant :

```
UID        PID  PPID  C STIME TTY          TIME CMD
...
pi        6211   831  8 09:07 ?        00:01:16 /usr/lib/chromium-browser/chromium-browser-v7 --disable-quic --enable-tcp-fast-open --p
pi        6252  6211  0 09:07 ?        00:00:00 /usr/lib/chromium-browser/chromium-browser-v7 --type=zygote --ppapi-flash-path=/usr/lib
pi        6254  6252  0 09:07 ?        00:00:00 /usr/lib/chromium-browser/chromium-browser-v7 --type=zygote --ppapi-flash-path=/usr/lib
pi        6294  6211  4 09:07 ?        00:00:40 /usr/lib/chromium-browser/chromium-browser-v7 --type=gpu-process --field-trial-handle=1
pi        6300  6211  1 09:07 ?        00:00:16 /usr/lib/chromium-browser/chromium-browser-v7 --type=utility --field-trial-handle=10758
pi        6467  6254  1 09:07 ?        00:00:11 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       11267  6254  2 09:12 ?        00:00:15 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       12035   836  0 09:13 ?        00:00:00 /usr/lib/libreoffice/program/oosplash --writer file:///home/pi/Desktop/mon_fichier.odt
pi       12073 12035  2 09:13 ?        00:00:15 /usr/lib/libreoffice/program/soffice.bin --writer file:///home/pi/Desktop/mon_fichier.o
pi       12253   831  1 09:13 ?        00:00:07 /usr/bin/python3 /usr/bin/sense_emu_gui
pi       20010  6211  1 09:21 ?        00:00:00 /usr/lib/chromium-browser/chromium-browser-v7 --type=utility --field-trial-handle=10758
pi       20029  6254 56 09:21 ?        00:00:28 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       20339  6254  4 09:21 ?        00:00:01 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       20343  6254  2 09:21 ?        00:00:00 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       20464  6211 17 09:22 ?        00:00:00 /proc/self/exe --type=utility --field-trial-handle=1075863133478894917,6306120996223181
pi       20488  6254 14 09:22 ?        00:00:00 /usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi       20519   676  0 09:22 pts/0    00:00:00 ps -ef
```

La documentation Linux donne la signification des différents champs :

- UID : identifiant utilisateur effectif ;
- PID : identifiant de processus ;
- PPID : PID du processus parent ;
- C : partie entière du pourcentage d'utilisation du processeur par rapport au temps de vie des processus ;
- STIME : l'heure de lancement du processus ;

- TTY : terminal de contrôle
- TIME : temps d'exécution
- CMD : nom de la commande du processus

Question 1

Parmi les quatre commandes suivantes, laquelle a permis cet affichage ?

a) ls -l
b) ps –ef
c) cd ..
d) chmod 741 processus.txt

Correction

Question 2

Quel est l'identifiant du processus parent à l'origine de tous les processus concernant le navigateur Web (chromium-browser) ?

Correction

Question 3

Quel est l'identifiant du processus dont le temps d'exécution est le plus long ?

Correction

3. 2021, France candidat libre

Extrait sujet BAC 2021

Cet exercice porte sur la gestion des processus par un système d'exploitation.

Partie A : Processus

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), le PPID (parent process identifier), l'état STAT et le nom de la commande à l'origine du processus.

Les valeurs du champ STAT indique 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
```

À l'aide de cet affichage, répondre aux questions ci-dessous.

Question 1

Quel est le nom de la première commande exécutée par le système d'exploitation lors du démarrage ?

Correction

Question 2

Quels sont les identifiants des processus actifs sur cet ordinateur au moment de l'appel de la commande ps ? Justifier la réponse.

Correction

Question 3

Depuis quelle application a-t-on exécuté la commande ps ?
Donner les autres commandes qui ont été exécutées à partir de cette application.

Correction

Question 4

Expliquer l'ordre dans lequel les deux commandes python programme1.py et python programme2.py ont été exécutées.

Correction

Question 5

Peut-on prédire que l'une des deux commandes python programme1.py et python programme2.py finira avant l'autre ?

Correction

4. 2023, Polynésie, J1

Ordonnanceur avec liste d'attente

Un système est composé de 4 périphériques, numérotés de 0 à 3, et d'une mémoire, reliés entre eux par un bus auquel est également connecté un dispositif ordonnanceur. À l'aide d'un signal spécifique envoyé sur le bus, l'ordonnanceur sollicite à tour de rôle les périphériques pour qu'ils indiquent le type d'opération (lecture ou écriture) qu'ils souhaitent effectuer, et l'adresse mémoire concernée.

Un tour a lieu quand les 4 périphériques ont été sollicités. **Au début d'un nouveau tour, on considère que toutes les adresses sont disponibles en lecture et écriture.**

Si un périphérique demande l'écriture à une adresse mémoire **à laquelle on n'a pas encore accédé** pendant le tour, l'ordonnanceur répond `"OK"` et l'écriture a lieu. Si on a déjà demandé la lecture ou l'écriture à cette adresse, l'ordonnanceur répond `"ATT"` et l'opération n'a pas lieu.

Si un périphérique demande la lecture à une adresse à laquelle on n'a pas encore accédé en écriture pendant le tour, l'ordonnanceur répond `"OK"` et la lecture a lieu. Plusieurs lectures peuvent avoir donc lieu pendant le même tour à la même adresse.

Si un périphérique demande la lecture à une adresse à laquelle on a déjà accédé en écriture, l'ordonnanceur répond `"ATT"` et la lecture n'a pas lieu.

Ainsi, pendant un tour, une adresse peut être utilisée soit une seule fois en écriture, soit autant de fois qu'on veut en lecture, soit pas utilisée.

Si un périphérique ne peut pas effectuer une opération à une adresse, il demande la même opération à la même adresse au tour suivant.

1. Le tableau donné en annexe 1 indique, sur chaque ligne, le périphérique sélectionné, l'adresse à laquelle il souhaite accéder et l'opération à effectuer sur cette adresse. Compléter dans la dernière colonne de cette annexe, à rendre avec la copie, la réponse donnée par l'ordonnanceur pour chaque opération.

Annexe 1

N° périphérique	Adresse	Opération	Réponse de l'ordonnanceur
0	`10`	écriture	`"OK"`
1	`11`	lecture	`"OK"`
2	`10`	lecture	`"ATT"`
3	`10`	écriture	`"ATT"`
0	`12`	lecture	
1	`10`	lecture	
2	`10`	lecture	
3	`10`	écriture	

Réponse

On suppose dans toute la suite que :

- le périphérique 0 écrit systématiquement à l'adresse `10` ;
- le périphérique 1 lit systématiquement à l'adresse `10` ;
- le périphérique 2 écrit alternativement aux adresses `11` et `12` ;
- le périphérique 3 lit alternativement aux adresses `11` et `12` ;

Pour les périphériques 2 et 3, le changement d'adresse n'est effectif que lorsque l'opération est réalisée.

2. On suppose que les périphériques sont sélectionnés à chaque tour dans l'ordre 0 ; 1 ; 2 ; 3. Expliquer ce qu'il se passe pour le périphérique 1.

✓ **Réponse** ›

Les périphériques sont sollicités de la manière suivante lors de quatre tours successifs :

- au premier tour, ils sont sollicités dans l'ordre 0 ; 1 ; 2 ; 3 ;
- au deuxième tour, dans l'ordre 1 ; 2 ; 3 ; 0 ;
- au troisième tour, 2 ; 3 ; 0 ; 1 ;
- puis 3 ; 0 ; 1 ; 2 au dernier tour.
- Et on recommence...

3.a. Préciser pour chacun de ces tours si le périphérique 0 peut écrire et si le périphérique 1 peut lire.

✓ **Réponse** ›

3.b. En déduire la proportion des valeurs écrites par le périphérique 0 qui sont effectivement lues par le périphérique 1.

✓ **Réponse** ›

On change la méthode d'ordonnancement : on détermine l'ordre des périphériques au cours d'un tour à l'aide de deux listes d'attente `ATT_L` et `ATT_E` établies au tour précédent.

Au cours d'un tour, on place dans la liste `ATT_L` toutes les opérations de lecture mises en attente, et dans la liste d'attente `ATT_E` toutes les opérations d'écriture mises en attente.

Au début du tour suivant, on établit l'ordre d'interrogation des périphériques en procédant ainsi :

- on interroge ceux présents dans la liste `ATT_L`, par ordre croissant d'adresse,
- on interroge ensuite ceux présents dans la liste `ATT_E`, par ordre croissant d'adresse,
- puis on interroge les périphériques restants, par ordre croissant d'adresse.

4. Compléter et rendre avec la copie le tableau fourni en annexe 2, en utilisant l'ordonnancement décrit ci-dessus, sur 3 tours.

Annexe 2

Tour	N° périphérique	Adresse	Opération	Réponse ordonnanceur	`ATT_L`	`ATT_E`
1	0	`10`	écriture	`"OK"`	vide	vide
1	1	`10`	lecture	`"ATT"`	`(1, 10)`	vide
1	2	`11`	écriture			
1	3	`11`	lecture			
2	1	`10`	lecture			vide
2						
2						
2						
3	0	`10`	écriture		vide	vide
3	1	`10`	lecture			vide
3	2	`11`	écriture	`"OK"`	`(1, 10)`	vide
3	3	`12`	lecture			

Réponse

Les colonnes **e0** et **e1** du tableau suivant recensent les deux chiffres de l'écriture binaire de l'entier **n** de la première colonne.

nombre n	écriture binaire de n sur deux bits	e1	e0
0	00	0	0
1	01	0	1
2	10	1	0

nombre n	écriture binaire de n sur deux bits	e1	e0
3	11	1	1

L'ordonnanceur attribue à deux signaux sur le bus de données les valeurs de **e0** et **e1** associées au numéro du circuit qu'il veut sélectionner. On souhaite construire à l'aide des portes ET, OU et NON un circuit pour chaque périphérique. Chacun des quatre circuits à construire prend en entrée deux signaux **e0** et **e1**, le signal de sortie **s** valant 1 uniquement lorsque les niveaux de **e0** et **e1** correspondent aux bits de l'écriture en binaire du numéro du périphérique correspondant.

Par exemple, le circuit ci-dessous réalise la sélection du périphérique 3. En effet, le signal **s** vaut 1 si et seulement si **e0** et **e1** valent tous les deux 1.

```
flowchart LR
    e0---ET(ET)
    e1---ET
    ET---s
    style e0 stroke-width:0px,opacity:0
    style e1 stroke-width:0px,opacity:0
    style s stroke-width:0px,opacity:0
```

5.a. Recopier sur la copie et indiquer dans le circuit ci-dessous les entrées **e0** et **e1** de façon que ce circuit sélectionne le périphérique 1.

```
flowchart LR
    ea( )---NON(NON)---ET(ET)
    eb( )----ET
    ET---s
    style ea stroke-width:0px,opacity:0
    style eb stroke-width:0px,opacity:0
    style s stroke-width:0px,opacity:0
```

✓ **Réponse** ›

5.b. Dessiner un circuit constitué d'une porte ET et d'une porte NON, qui sélectionne le périphérique 2.

✓ **Réponse** ›

5.c. Dessiner un circuit permettant de sélectionner le périphérique 0.

✓ **Réponse** ›

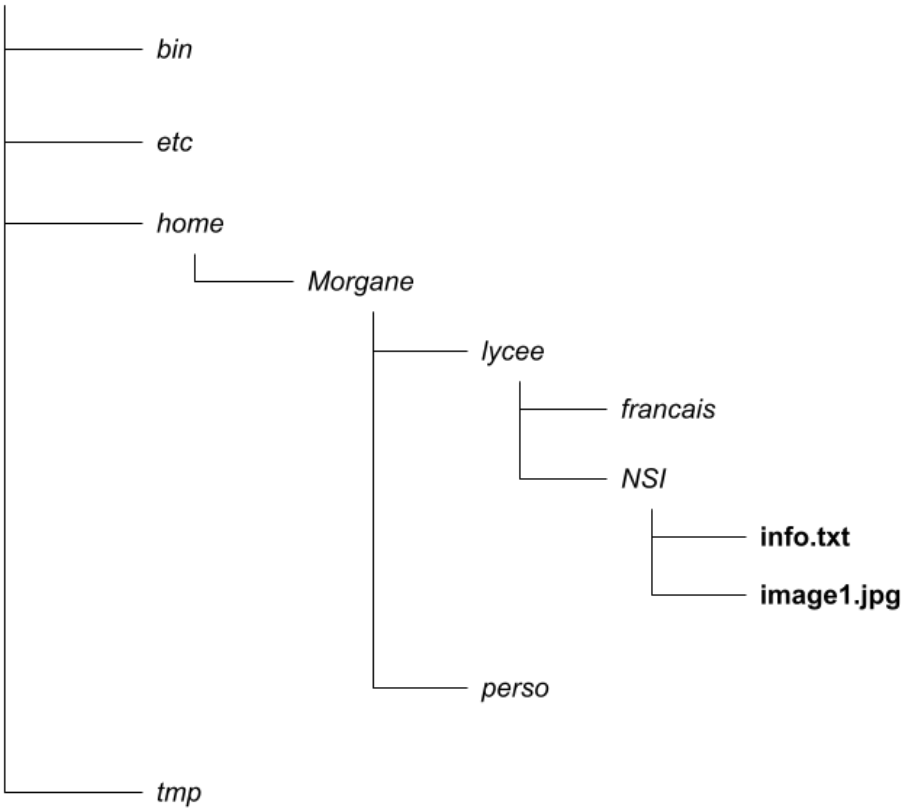
5. 2022, Amérique du Nord, J2

Exo

Cet exercice pourra utiliser des commandes de système d'exploitation de type UNIX telles que `cd, ls, mkdir, rm, rmd, mv, cat`.

Question 1

Dans un système d'exploitation de type UNIX, on considère l'arborescence des fichiers suivante dans laquelle les noms de dossiers sont en italique et ceux des fichiers qui sont en gras :



On souhaite, grâce à l'utilisation du terminal de commande, explorer et modifier les répertoires et fichiers présents.
On suppose qu'on se trouve actuellement à l'emplacement */home/morgane*

(a) Parmi les quatre propositions suivantes, donner celle correspondant à l'affichage obtenu lors se l'utilisation de la commande `ls`.

- **Proposition 1 :** lycee francais NSI info.txt image1.jpg perso
- **Proposition 2 :** lycee perso
- **Proposition 3 :** morgane
- **Proposition 4 :** bin etc home tmp

(b) Ecrire la commande qui permet, à partir de cet emplacement, d'attendre le répertoire **lycee**.
On suppose maintenant qu'on se trouve dans le repertoire `/home/morgane/lycee/NSI`

(c) Ecrire la commande qui permet de créer à cet emplacement un répertoire nommé `algorithmique`.

(d) Ecrire la commande qui permet, à partir de cet emplacement, de supprimer le fichier `image1.jpg`.

Quesiton 2

On rappelle qu'un processus est une instance d'application. Un processus peut être démarré par l'utilisateur, par un périphérique ou par un autre processus appelé parent.
La commande UNIX `pa` présente un cliché instantané des procesus en cours d'exécution.
On a exécuté la commande `ps` (avec quelques options qu'il n'est pas nécessaire de connaître pour la réussite de cet exercice). Un extrait du résultat de la commande est présenté ci-dessous :

```
UID          PID  PPID  C STIME TTY          TIME CMD
test         900   739  0 10:51 ?        00:00:00 /usr/lib/gvfs/gvfs-udisks2-vol
test         907   838  0 10:51 ?        00:00:00 /usr/lib/gvfs/gvfsd-trash --sp
test         913   739  0 10:51 ?        00:00:00 /usr/lib/gvfs/gvfsd-metadata
test         918   823  0 10:51 ?        00:00:00 /usr/lib/x86_64-linux-gnu/xfce
test         919   823  0 10:51 ?        00:00:00 /usr/lib/x86_64-linux-gnu/xfce
test         923     1  0 10:51 ?        00:00:02 xfce4-terminal
test         927   923  0 10:51 pts/0    00:00:00 bash
test        1036     1  0 11:18 ?        00:00:02 mousepad /home/test/Documents/
test        1058   923  0 11:22 pts/1    00:00:00 bash
root        1132     2  0 11:37 ?        00:00:00 [kworker/0:0-ata_sff]
root        1134     2  0 11:43 ?        00:00:00 [kworker/0:2-ata_sff]
test        1140   739  0 11:43 ?        00:00:00 /usr/lib/x86_64-linux-gnu/tumb
root        1149     2  0 11:43 ?        00:00:00 [kworker/u2:0-events_unbound]
test        1153   927  0 11:44 pts/0    00:00:00 vi
test        1154   927  0 11:44 pts/0    00:00:00 python3 prog.py
test        1155  1058  0 11:44 pts/1    00:00:00 ps -aef
```

On rappelle que :

- l'UID est l'identifiant de l'utilisateur propriétaire du processus;
- le PID est l'identifiant du processus parent;
- le PPID est l'identifiant du provessus parent;
- C indique l'utilisation processeur;
- STIME est l'heure de démarrage du processus;
- TTy est le nom du terminal de commande auquel le processus est attaché;
- TIME est la durée d'utilisation du processus par le processeur;
- CMD le nom de commande utilisé pour démarrer le processus.

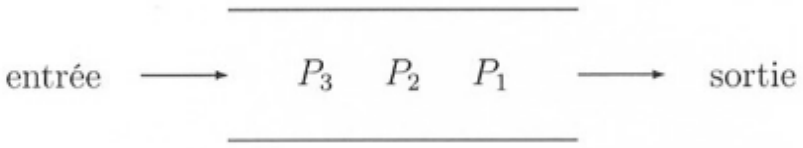
(a) Donner le PID du parent du processus démarré par la commande `vi`.
(b) Donner le PID d'un processus enfant du processus démarré par la commande `xfce4-terminal`.
(c) Citer le PID de deux processus qui ont le même parent.
(d) Parmi tous les processus affichés, citer le PID des deux qui ont consommé le plus de temps.

Question 3

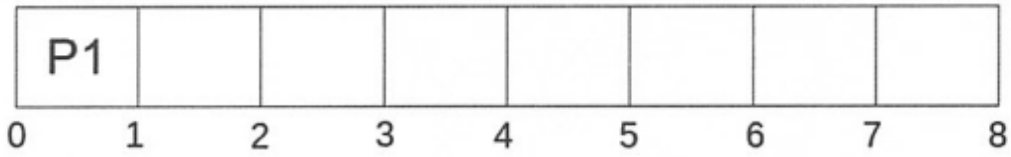
On considère les trois processus P1, P2, P3, tous soumis à l'instant 0 dans l'ordre 1, 2, 3 :

Nom du processus	Durée d'exécution en unité de temps	Ordre de soumission
P1	3	1
P2	1	2
P3	4	3

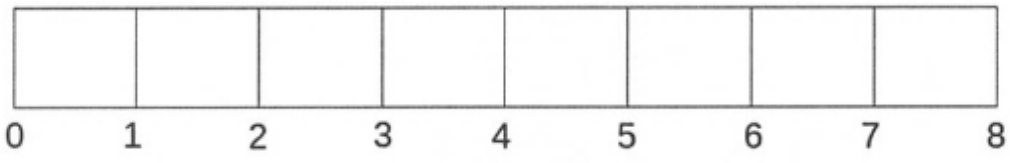
(a) Dans cette question, on considère que les processus sont exécutés de manière concurrente selon la politique du tourniquet : le temps est découpé en tranches nommées *quantums de temps*.
Les processus prêts à être exécutés sont placés dans une file d'attente selon leur ordre de soumission. Lorsqu'un processus est élu, il s'exécute au plus durant un quantum de temps. Si le processus n'a pas terminé son exécution à l'issue du quantum de temps, il réintègre la file des processus prêts (coté entrée). Un autre processus, désormais en tête de la file (coté sortie) des processus prêts, est alors à son tour élu pour une durée égale à un quantum de temps maximum.



Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus exécuté à chaque cycle. Le quantum correspond à une unité de temps.



(b) Dans cette question, on considère que les processus sont exécutés en appliquant la politique du plus court d'abord : les processus sont exécutés complétement dans l'ordre croissant de leurs temps d'exécution, le plus court étant exécuté en premier.
Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus excécuté à chaque cycle.



Question 4

On considère trois ressources R1, R2 et R3 et trois processus P1,P2 et P3 dont les files d'exécution des instructions élèmentaires sont indiquées ci-dessous :

Processus P_1
Demande R_1
Demande R_2
Libère R_1
Libère R_2

Processus P_2
Demande R_2
Demande R_3
Libère R_2
Libère R_3

Processus P_3
Demande R_3
Demande R_1
Libère R_3
Libère R_1

(a) Rappeler les différens états d'un processus et expliquer pourquoi il y a ici risque d'interblocage, en proposant un ordre d'exécution des instructions élèmentaires le provoquant.
(b) Proposer un ordre d'exécution des instructions élèmentaires sans interblocage.