

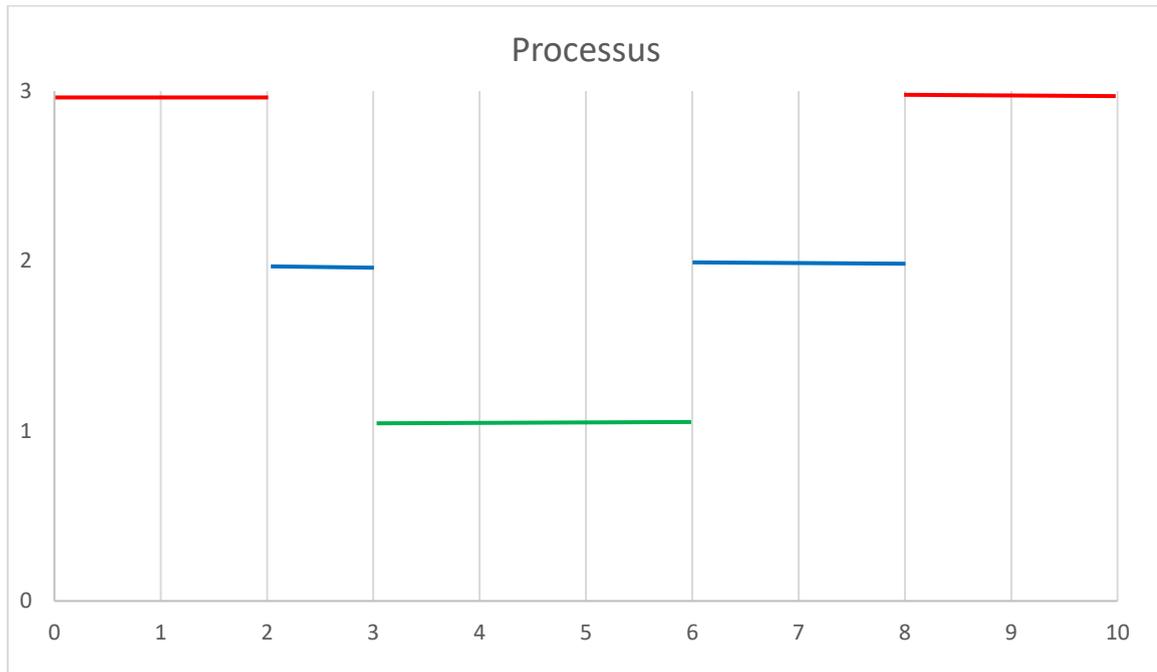
Exercice 1

PARTIE A

- 1b en système Unix (Linux, ici) on utilise *ps*
- 2c **Ordonnement**
- 3b en système Linux : **PID**
- 4d en système Unix (Linux, ici) on utilise *kill*

PARTIE B

1 Les processus **P1** , **P2** et **P3** son notés **1**, **2** et **3** sur le graphique



P3	P3	P2	P1	P1	P1	P2	P2	P3	P3	
0	1	2	3	4	5	6	7	8	9	10

2

Il s'agit du **scénario 2** car nous nous retrouvons dans la situation où **P1 possède R1** et **attend R2** avant de pouvoir continuer et **P3 possède R2** et **attend R1** avant de pouvoir continuer.

Exercice 2

Partie A

1

Pour passer à l'exécution, le **Programme 1 a besoin du modem**, mais le modem est **déjà utilisé** par le *Programme 2*. Le processus **p1 est donc bloqué**.

Pour passer à l'exécution, le **Programme 2 a besoin de l'imprimante**, mais l'imprimante est **déjà utilisée** par le *Programme 3*. Le processus **p2 est donc bloqué**.

Pour passer à l'exécution, le **Programme 3 a besoin de la table traçante**, mais la table traçante est **déjà utilisée** par le *Programme 1*. Le processus **p3 est donc bloqué**.

Les 3 processus sont bloqués et ne pourront pas libérer les ressources attendues par les autres processus (la libération des ressources se faisant après l'exécution), nous sommes bien en **situation d'interblocage**.

2

<u>Programme 3</u>
demander (table traçante)
demander (imprimante)
exécution
libérer (imprimante)
libérer (table traçante)

3

b. bloqué

Partie B

1

ps -ef (permet l'affichage des processus sous Linux)

2

L'identifiant du processus parent à l'origine de tous les processus concernant le navigateur Web est 831

3

L'identifiant du processus dont le temps d'exécution est le plus long est 6211 (00:01:16)

Exercice 3

1

Des problèmes peuvent survenir, lorsque les processus obtiennent des accès exclusifs aux ressources.

Par exemple, un processus P1 détient une ressource R1 et attend une autre ressource R2 qui est utilisée par un autre processus P2.

Le processus P2 détient la ressource R2 et attend la ressource R1.

On a une **situation d'interblocage** (deadlock en anglais) car P1 attend P2 et P2 attend P1.

Les deux processus vont attendre indéfiniment.

En général, un ensemble de processus est en interblocage si chaque processus attend la libération d'une ressource qui est allouée à un autre processus de l'ensemble. Comme tous les processus sont en attente, aucun ne pourra s'exécuter et donc libérer les ressources demandées par les autres.

Ils attendront tous indéfiniment.

2

- a) Enlever l'exclusion mutuelle : par exemple des opérations algorithmes "sans blocages" (utilisation d'opérations "lecture-modification-écriture")
- b) Enlever "hold and wait" : Exemple, s'imposer de demander au plus 1 ressource à la fois. Eviter de bloquer plusieurs ressources en même temps.
- c) Contre l'attente circulaire. Ne pas faire de cycle dans la demande de ressources (dans la mesure du possible)