TD - Gestion des processus et des ressources

M. Tellene

1 Quelques rappels

1. Expliquer la différence entre un programme et un processus
2. Citer les 3 états d'un processus
3. Donner une définition et un exemple d'un interblocage

2 Ordonnancement

Ce tableau est le tableau des processus avec leur nom, leur date de création et la durée du processus. Il sera utilisé pour les 4 prochains exercices.

Processus	Date de création	Durée
P1	0	5
P2	3	8
P3	7	6
P4	9	4

Exercice 1

1. Expliquer le fonctionnement de l'algorithme **FIFO**
2. Exécuter l'algorithme **FIFO** et compléter l'annexe 1

Exercice 2

1. Expliquer le fonctionnement de l'algorithme **Shortest Job First**
2. Exécuter l'algorithme **Shortest Job First** et compléter l'annexe 2

Exercice 3

La préemption est la capacité d'un processus à prendre la place d'un autre. C'est à dire qu'un processus peut être interrompu si un autre arrive, avec une durée plus courte.

Exécuter l'algorithme **Shortest Job First avec préemption** et compléter l'annexe 3.

Exercice 4

1. Expliquer le fonctionnement de l'algorithme **Round Robin**
2. Exécuter l'algorithme **Round Robin** et compléter l'annexe 4. On fixe le temps maximum a 2 cycles

Exercice 5

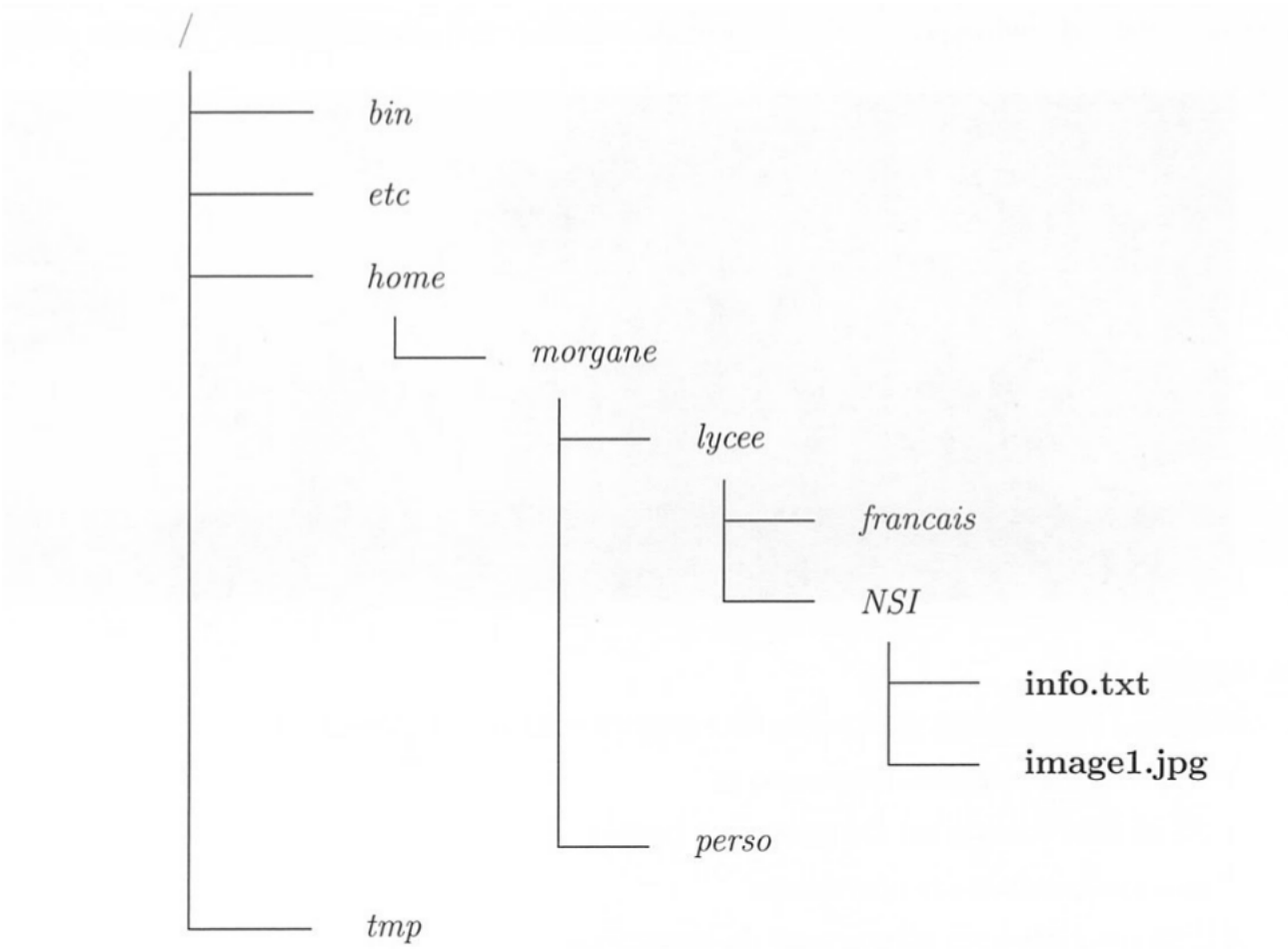
On considère 5 tâches J_1, J_2, J_3, J_4, J_5 de durées respectives 3, 5, 2, 1, 2 pouvant s'exécuter sur 3 machines. Pas plus d'une tâche ne peut être exécutée au même moment par une même machine. Une fois lancée, la tâche doit être achevée.

Comment ordonnancer les tâches de manière à toutes les terminer le plus tôt possible ?

3 Exercice type bac - Amérique du Nord (Sujet 2 - 2022)

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

1. Dans un système d'exploitation de type `UNIX`, on considère l'arborescence des fichiers suivante dans laquelle les noms de dossier sont en italique et ceux des fichiers 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 de l'utilisation de la commande `ls`.

Propostion 1 : lycee francais NSI info.txt image1.jpg perso

Propostion 2 : lycee perso

Propostion 3 : morgane

Propostion 4 : bin etc home tmp

(b) Écrire la commande qui permet, à partir de cet emplacement, d'atteindre le répertoire `lycee`.

On suppose maintenant qu'on se trouve dans le répertoire `/home/morgane/lycee/NSI`.

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

(d) Écrire la commande qui permet, à partir de cette emplacement ; de supprimer le fichier `image1.jpg`.

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 `ps` présente un cliché instantané des processus 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 ;
— le `PPID` est l'identifiant du processus parent ;
— `C` indique l'utilisation du 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 processeur par le processus ;
— `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 du processeur.

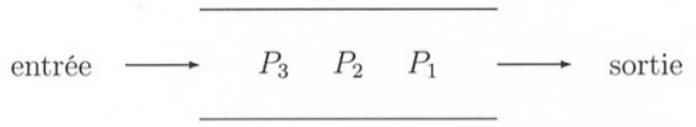
3. On considère les trois processus $P1$, $P2$ et $P3$, tous soumis à l'instant 0 dans l'ordre 1, 2, 3 :

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

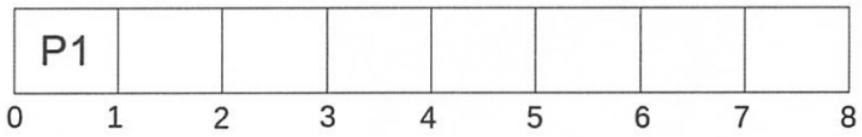
(a) Dans cette question, on considère que les processus sont exécutés de manière concurrente selon la politique du tourniquet : le temps est découpé en tranches nommés *quantums de temps*.

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 (côté entrée). Un autre processus, désormais en tête de la file (côté sortie) des processus prêts, est alors à son tour élu pour une durée égale à un quantum de temps maximum.

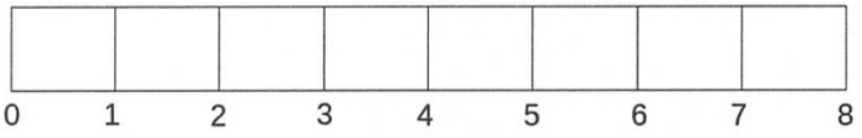


Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus exécuté à chaque cycle. Le quantum correspond à une unité de temps.



(b) Dans cette question, on considère que les processus sont exécutés en appliquant la politique du « plus court d'abord » : les processus sont exécutés complètement dans l'ordre croissant de leur 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 exécuté à chaque cycle.



4. On considère trois ressources R_1, R_2 et R_3 et trois processus P_1, P_2 et P_3 dont les files d'exécution des instructions élémentaires 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

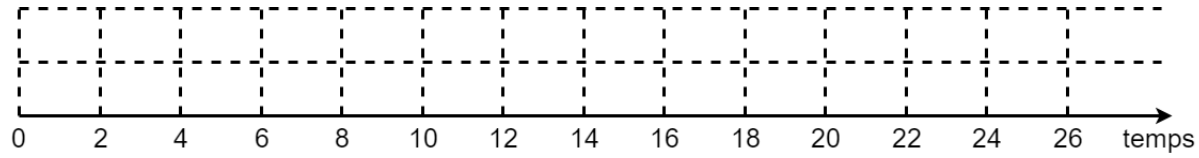
Processus P_3
Demande R_3
Demande R_1
Libère R_3
Libère R_1

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

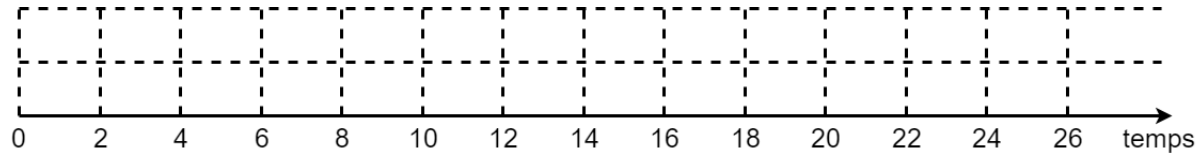
(b) Proposer un ordre d'exécution des instructions élémentaires sans interblocage.

4 Annexes

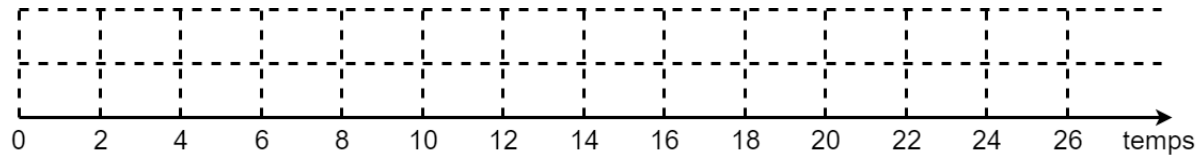
Annexe 1 - FIFO



Annexe 2 - Shortest Job First



Annexe 3 - Shortest Job First avec préemption



Annexe 4 - Round Robin