

Évaluation DIU-EIL bloc3 - Partie Système (30 min)

Fabien Rico

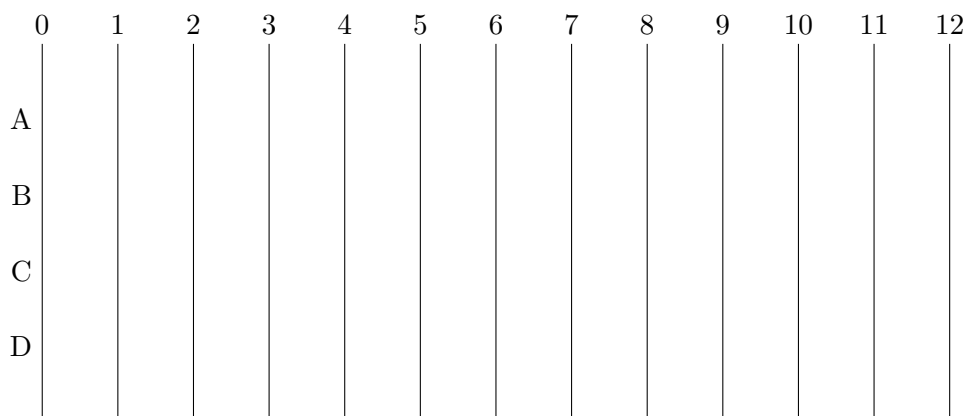
26 juin 2020

1 Ordonnement avec liste de priorité

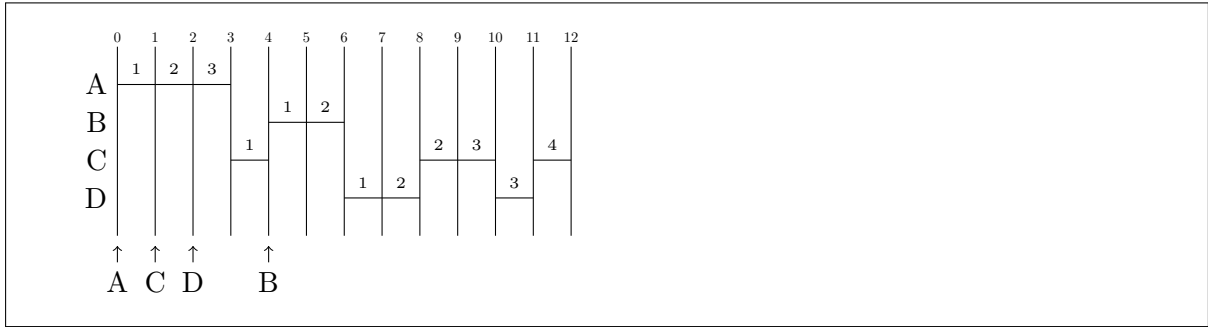
- Le système gère des listes de priorités.
- On se place en mode préemptif c'est à dire que lorsqu'une tâche plus prioritaire arrive, le système lui donne immédiatement accès au processeur.
- Pour des tâches de même priorité, le système utilise le **Round Robin** avec un quantum de 2.
- Les priorités sont croissante, plus la valeur est grande, plus la tâche est prioritaire.
- *Précision* : Pour Round Robin, une tâche qui perd la main perd son tour même si son quantum d'exécution n'était pas terminé.

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	0	10	3	
B	4	5	2	
C	1	4	4	
D	2	4	3	

1. Dans la figure suivante, faites l'ordonnement sur 12 unités de temps.



Solution:



2 Arborescence de fichiers et droits

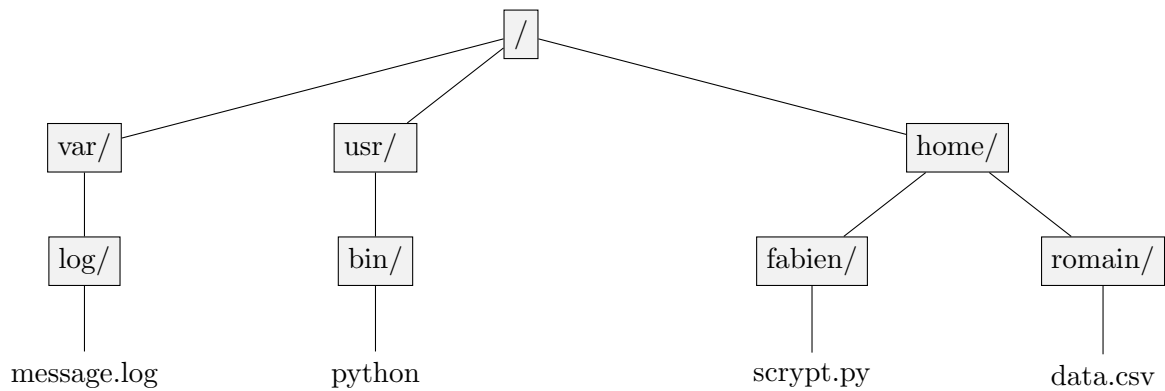


FIGURE 1 – Exemple de hiérarchie

```

dr-xr-xr-x 20    root  root  /
drwxr-xr-x 13    root  root  /usr/
drwxr-xr-x  2    root  root  /usr/bin/
-rwsr-xr--  1    root  diu   /usr/bin/python
drwxr-xr-x 30    root  root  /var/
drwxr-xr-- 10    root  log   /var/log/
-rw-rw-r--  1    root  log   /var/log/message.log
drwxr-xr-x 40    root  user  /home/
drwxr-xr-x 10    fabien diu   /home/fabien/
-rw-r--r--  1    fabien diu   /home/fabien/script.py
drwxr-xr-x  1    romain user  /home/romain/
-rw-r-----  1    romain diu   /home/romain/data.csv

```

La figure 1 représente une partie de la hiérarchie d'un système Linux. Le programme `/usr/bin/python` est un interpréteur Python qui permet donc d'exécuter un code écrit dans ce langage.

Il y a 3 utilisateurs :

- fabien qui fait partie des groupes `user`, `diu` et `log` ;
- romain qui fait partie des groupes `user` et `diu` ;
- nicolas qui fait partie du groupe `user`.

1. Représentez les possibilités d'accès des 3 utilisateurs aux fichiers `python`, `message.log`, `script.py` et `data.csv` dans le tableau 2

2. Si vous vous trouvez dans le répertoire `fabien`, donnez 2 manières d'atteindre le fichier `python` en utilisant un chemin absolu et un chemin relatif.
3. Y a-t-il un moyen pour `romain` de voir le contenu du fichier `message.log`.

	<code>python</code>	<code>message.log</code>	<code>script.py</code>	<code>data.csv</code>
<code>fabien</code>				
<code>romain</code>				
<code>nicolas</code>				

FIGURE 2 – Tableau des droits

Solution:

question ??

	<code>python</code>	<code>message.log</code>	<code>script.py</code>	<code>data.csv</code>
<code>fabien</code>	<code>rx</code>	<code>rw</code>	<code>rw</code>	<code>r</code>
<code>romain</code>	<code>rx</code>	<code>-</code>	<code>r</code>	<code>rw</code>
<code>nicolas</code>	<code>-</code>	<code>-</code>	<code>r</code>	<code>-</code>

question 2

On peut définir un chemin absolu en partant de la racine /

`/usr/bin/python`

Et un chemin relatif, pour cela il faut remonter dans l'arbre puis aller sur le noeud correspondant au fichier

`../../usr/bin/python`

question 3

En utilisant l'interpreteur python qui a le `setuid` bit. En effet, ce fichier appartient à `root`, tout utilisateur du groupe `diu` peut l'exécuter. Mais ce fichier à le `setuid` bit (i.e. le `s` placé à la place du `x` pour l'utilisateur propriétaire). Cela signifie que si `fabien` ou `romain` exécute ce fichier, le système donnera au processus les même droit que celui de `root` et permettra de faire tout ce que le programme exécuté est capable de faire.

`python` étant un interpreteur python, il permet entre autre d'ouvrir des fichiers et de les lire ce que pourra donc faire `romain`. Il est à noter que seul les membres du groupe `diu` ont cette possibilité. `nicolas`, par exemple peut lire le contenu du fichier, il peut donc le copier, mais le bit `setuid` est attaché à ce fichier particulier, de plus le nouveau fichier n'appartiendra plus à `root`, la propriété n'est donc pas transférable lors d'une copie.

Réponse q 2

Réponse q 3

FIGURE 3 – Réponses