



INSTITUT D'ELECTRONIQUE  
ET D'INFORMATIQUE  
GASPARD-MONGE

## Examen SYSTÈME - L3

Mai 2016

*S. Cherrier*

### 1 Questions de cours (11pt)

#### 1.1 Processus

1. Pour quelles raisons a-t-on inventé la notion de processus ? Quels avantages cela offre t'il ?
2. Qui organise la gestion des processus ? Décrivez son mode de fonctionnement.
3. Gestion des processus : Donnez les 4 primitives systèmes utilisées pour les processus (création, etc..). Pour chacune d'elles, expliquez le fonctionnement.

4. Est-ce que les processus peuvent communiquer entre eux ? Décrivez les différentes solutions.

## **1.2 Connaissances générales**

1. Décrire le terme suivant : système temps réel. Votre système est-il temps réel ? Si non, expliquez les différences.

2. Qu'est ce que le "changement de contexte" dans un processeur ? Pourquoi et dans quel cas est-ce utilisé dans le cadre de ce cours ?

### **1.3 Mémoire**

1. Dans un système multitâches, pourquoi doit on gérer la mémoire ? quelles problèmes peuvent se poser ?
2. Qu'est ce qu'un défaut de page ?
3. Décrire le système idéal de gestion des pages mémoire, puis un algorithme que vous choisirez parmi ceux existant.

## **1.4 Entrées-Sorties**

1. Décrire la notion de "bloquant". Donner des exemples de commandes bloquantes.
2. Outre les entrées sorties, comment 'bloquer' l'exécution d'un programme ?
3. Décrire le rôle et le fonctionnement des commandes dup et dup2.

4. Quelles sont les caractéristiques d'un tube (pipe). En décrire l'utilisation.

## 2 Programmation en C Système (9 pt)

*Pour rappel, la commande `system()` est évidemment interdite. De même, pensez à commenter, et à gérer proprement l'ensemble des cas de figure. En cas de doute sur une fonction et notamment la syntaxe, décrivez en commentaire les éléments que vous comptez fournir, plutôt que de ne rien écrire.*

### 2.1 Le plus grand d'entre nous

Écrire un programme déterminant quel est le plus grand numéro de PID possible pour un processus (PS : créez vraiment les processus, et surveiller les résultats. Quand la limite est atteinte, le système recherche le premier numéro de PID disponible en repartant de 2)

### 2.2 Le multi-executor

Écrire le programme complet qui permet de lancer l'exécution de la commande (et de ses arguments) passée en argument sur la ligne de commande. Votre "executor" doit indiquer "RUNNING XXXX / START" avant l'exécution du programme demandé. Il attend une seconde, puis relance la même exécution. Au bout de 5 fois, il s'arrête et écrit "STOP" (XXXX étant le nom du programme qui va être lancé, et strictement ce nom de programme comme dans l'exemple ci-dessous).

*Bonus : Les erreurs de la commande ne doivent en aucun cas être affichées, vous penserez donc à les éliminer.*

**./executor ls -al /tmp**

RUNNING ls

START

execution 1 :

```
drwxrwxrwt 11 root      root      12288 juin  30 20:41 .
drwxr-xr-x  24 root      root      4096 juin  26 08:58 ..
-rw-rw-r--   1 sylvain   sylvain 660347 juin  30 19:08 Android-DLite.apk
```

execution 2 :

```
drwxrwxrwt 11 root      root      12288 juin  30 20:41 .
drwxr-xr-x  24 root      root      4096 juin  26 08:58 ..
-rw-rw-r--   1 sylvain   sylvain 660347 juin  30 19:08 Android-DLite.apk
```

execution 3 :

```
drwxrwxrwt 11 root      root      12288 juin  30 20:41 .
drwxr-xr-x  24 root      root      4096 juin  26 08:58 ..
-rw-rw-r--   1 sylvain   sylvain 660347 juin  30 19:08 Android-DLite.apk
drwx-----   2 sylvain   sylvain  4096 juin  30 20:08 evince-4109
```

//etc etc jusqu'à la 5...

STOP

NOTA : Vous remarquerez que le ls a donné un résultat différent à la troisième itération, puisque la réponse de la commande a évolué au cours du temps.