

info202 : Système d'exploitation
TD 2 : système d'exploitation et processus

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 22, poste : 94 22
email : Pierre.Hyvernat@univ-smb.fr
www : <http://www.lama.univ-savoie.fr/~hyvernat/>

Exercice 1 : questions de cours

Question 1. Donnez une définition succincte de “appel système”, et donnez quelques exemples. Donnez une définition succincte de “noyau”.
Donnez une définition succincte de “système d'exploitation”.

Partie 1 : processus

Question 1. Les deux processus P_1 et P_2 ont chacun besoin de 1 minute de temps de processeur pour faire ses calculs. Chacun est *bloqué* pendant 50% de son temps total d'exécution. (Cela signifie qu'à chaque instant, chaque processus a une probabilité 1/2 d'être bloqué.)

- Quelle est la durée d'exécution du processus P_1 seul ?
- Quelle est la durée d'exécution des deux processus P_1 et P_2 s'ils sont exécutés séquentiellement ? (Autrement dit, le processus P_2 commence à s'exécuter lorsque le processus P_1 est terminé.)
- Quelle est la durée d'exécution des deux processus P_1 et P_2 s'ils sont exécutés avec un ordonnanceur “multiprogrammation” qui permet de changer de processus au bout d'un court laps de temps ?

Exercice 2 : shell et composition (POSIX)

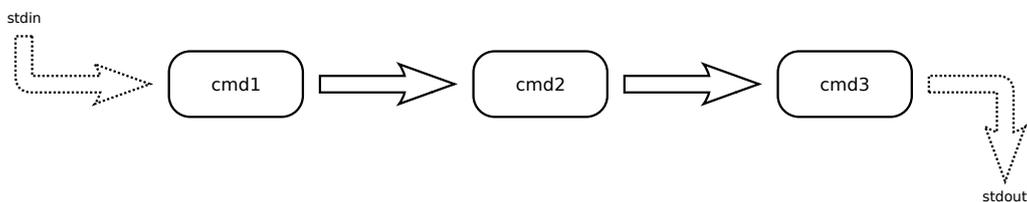
Rappels :

```
$ CMD1 | CMD2 | CMD3
```

permet

- de lancer la commande $CMD1$
- de faire agir $CMD2$ sur la sortie de $CMD1$
- de faire agir $CMD3$ sur la sortie de $CMD2$

On pourrait visualiser ceci par



- ▷ La commande “`wc -l`” permet de compter le nombre de lignes :
 - dans des fichiers si des chemins d'accès sont donnés avec la commande (“`wc -l tp1.py`” par exemple)
 - sur l'*entrée standard* si aucun chemin n'est donné.

Question 1. Le répertoire courant contient les fichiers `cours.md`, `cours.html`, `tp1.html`, `examen.pdf` et `liste_etudiants.html`.

Quel sera le résultat des commandes suivantes ?

```
$ ls | wc -l
$ ls *.html | wc -l | wc -l
$ wc -l examen.pdf | ls *.pdf
```

Question 2.

- ▷ La commande “**grep CHAINE**” permet d’afficher les lignes contenant la chaîne de caractères **CHAINE** sur la sortie standard.* Si des noms de fichiers sont donnés, la recherche se fait dans les lignes de ces fichiers ; si aucun nom de fichier n’est donné, la recherche des lignes se fait sur l’entrée standard.

Comment peut-on compter le nombre de lignes contenant la chaîne **TODO** dans tous les fichiers **.html** du répertoire courant ?

Question 3.

- ▷ La commande “**sort**” permet de trier les lignes de l’entrée standard (ou de fichiers, si des noms sont donnés à la commande) et de les afficher sur la sortie standard. L’option **-n** permet de faire un tri numérique plutôt que alphabétique. (Pour le tri alphabétique, **10** vient avant **2**!)
- ▷ La commande “**uniq**” permet de supprimer les lignes identiques *consécutives* de l’entrée standard (ou d’un fichier, si un chemin d’accès est donné avec la commande) et d’afficher le résultat sur la sortie standard. L’option **-c** permet de préfixer chaque ligne du résultat avec le nombre d’occurrences consécutives de la ligne.
- ▷ Les commandes “**head**” et “**tail**” permettent d’afficher uniquement les 10 premières ou dernières lignes de l’entrée standard (ou d’un fichier si un chemin est donné avec la commande).

Comment peut-on afficher la liste des lignes du fichier **test.txt** en supprimant *tous* les doublons (lignes identiques). L’ordre des lignes affichées n’a pas besoin d’être identique à l’ordre des lignes dans le fichier.

Comment peut-on afficher la liste des 10 lignes qui apparaissent le plus souvent dans le fichier **test.txt**. (Les lignes affichées peuvent contenir le nombre d’occurrences dans le fichier.)

Question 4.

- ▷ L’option **-l** de la commande **grep** permet de seulement afficher le nom des fichiers qui contiennent au moins une ligne contenant la chaîne donnée.
- ▷ La commande “**xargs CMD**” lit une liste de fichiers (chemins d’accès) **FICHER1 ... FICHERn** sur l’entrée standard et exécute ensuite la commande “**CMD FICHER1 ... FICHERn**”.

Comment peut-on supprimer tous les fichiers du répertoire courant qui contiennent la chaîne **version-0** ?

Question 5.

- ▷ La commande “**find -size +1k -type f**” permet d’afficher les noms de tous les fichiers (**-type f**) dont la taille est supérieure à 1Kio (**-size +1**), dans tous les dossiers, sous-dossiers, etc.

Comment peut-on, à partir de cette commande et des commandes vues précédemment,

- obtenir le nombre de fichiers dont la taille est supérieure à 1Kio,
- obtenir le nombre de lignes contenant la chaîne **TODO** dans tous ces fichiers,
- obtenir le nombre de fichiers de plus de 1Kio contenant la chaîne **TODO**.

Pour les courageux (avec leur portable) :

- obtenir la liste des 10 fichiers avec le plus de lignes contenant **TODO**,

(Il faudra regarder les options **-c** de **grep**, **-t** et **-k** de “**sort**”) ainsi que la commande “**cut**”...)

* **CHAINE** peut être une *expression régulière* qui peut contenir des symboles particuliers permettant de reconnaître des lignes de manière plus complexe.