

TP1 Prog UNIX

1.b)

Méthode simple mais pas jolie :

```
ps -o pid -o ppid -o user -o uid -o pri -o addr -o sz
```

2.a)

La commande permet d'exécuter des commandes en différée. Pour lancer l'exécution dans un fichier rajouter >file. Sinon la sortie (d'un ls par exemple) sera envoyé par mail.

```
arnaud@fwft:~$ at 9:44
```

```
warning: commands will be executed using /bin/sh
```

```
at> ls >file
```

2.b)

Réception de la terminaison de la commande par mail.

2.d)

```
arnaud@fwft:~$ batch
```

```
warning: commands will be executed using /bin/sh
```

```
at> ls >out
```

2.e)

Il est préférable de lancer batch pour ne pas obliger la machine à exécuter la tâche à l'heure précise en cas de surcharge.

3.a)

Dans tout les cas que j'ai testé le processus s'est arrêté.

4.a)

Après un test sous Debian, il semblerai que la commande (téléchargement d'un gros fichier)

```
nohup wget big-file-test
```

ne puisse pas être interrompue par le signal SIGHUP (1) mais par un signal SIGINT(2) et SIGQUIT(3)...

5.c)

Aucune différence perçue sur un téléchargement. Il est difficile de voir une différence significative quant à l'exécution de ce genre de programmes.

6.a)

De même pour une copie de fichier l'utilisation de nice semble pas significative (voir même contradictoire dans ce cas...).

```
time nice -n 19 cp file copy-file
```

```
real    0m18.874s
```

```
user    0m0.028s
```

```
sys     0m3.520s
```

```
et
```

```
time nice -n 1 cp file copy-file
```

```
real    0m19.721s
```

```
user    0m0.056s
```

```
sys     0m3.420s
```

Exemple utilisation de la commande bg :

Créer une tâche longue à effectuer (par exemple, archiver un dossier) :

```
tar -vcf test.tar TarDirectory/
```

CTRL+Z pour mettre la tâche en pause.

```
[1]+  Stopped      tar -vcf test.tar TarDirectory/
```

Passer la commande en arrière plan (pour qu'elle continue à s'exécuter)

```
bg %1
```