M3101 – Principes des systèmes d'exploitation Semestre 3 DUT IUT de Blagnac	Iulian Ober Patricia Stolf

TP 3.3 (1.5h)

Dans la suite des TP nous utiliserons le langage Python (versions 3.x). La plupart des appels système Unix/POSIX sont implémentées également dans ce langage, principalement dans le module **os** *(il convient donc de démarrer vos programmes avec la clause :* **from os import *** *).*

1. Ecrire un shell simplifié en Python

Ecrire un programme Python (*myshell.py*) qui lit des lignes de commande à l'entrée standard et les lancent en exécution, à l'instar du *shell* standard Linux. Le programme ne doit pas utiliser la fonction *system,* l'exécution est lancée avec une fonction de la famille *os.exec*.

Pour simplifier le travail, nous allons faire les hypothèses suivantes :

- Une commande est formée d'une seule ligne. Seules les commandes *externes* (i.e., programmes exécutables avec arguments) seront traitées.
- Les arguments sont séparés par des espaces. Les caractères d'échappement (\), guillemets, etc. ne sont pas traités.
- Les opérateurs de combinaison de commandes (|, ||, &&, etc.) ou de redirection (<, >, etc.) ne sont pas pris en compte.
- En revanche, la ligne **peut se terminer par &** (séparé du dernier argument par un espace) ce qui produit un effet similaire au lancement en arrière-plan en *shell* standard.
- A la fin d'un processus lancé en premier plan (sans &), le shell affiche son PID et son code de retour

A noter : il convient de traiter les erreurs potentielles à chaque appel système (exception **OSError***) et afficher une description de la cause de l'erreur (cf.* **OSError.strerror** *)*