

TP AISE — Généralités sur les OS

Exercice: gestion de PTY

Réalisez un programme qui affiche la taille du terminal courant et qui supporte le redimensionnement de la fenêtre, affichant alors dynamiquement la nouvelle taille.

Voici comment on lit la taille courante du terminal:

```
#include <sys/ioctl.h>
#include <stdio.h>

int main (void)
{
    struct winsize w;
    ioctl(0, TIOCGWINSZ, &w);
    printf ("lines %d\n", w.ws_row);
    printf ("columns %d\n", w.ws_col);
    return 0;
}
```

Exercice: Fork et signaux1

Réalisez un programme qui lance un processus fils et ensuite lui envoie un signal toutes les trois secondes. Ce signal sera capturé dans le programme fils et produira sur sa sortie un compteur s'incrémentant. Enfin, dans le cas d'une interruption (SIGINT) du programme parent le programme fils sera également interrompu.

Exercice: Fork et signaux2

Réalisez un programme qui fork. Le parent enverra aléatoirement soit SIGCONT ou SIGPIPE à son fils qui devra compter le nombre de signaux reçus pour chaque type. Le parent opérera pendant 30 secondes avant de s'interrompre. Enfin, quand le parent aura fini la phase d'émission il terminera son fils qui affichera le compte des deux signaux (SIGCONT et SIGPIPE) avant de se terminer. Enfin, si le parent est interrompu avant la fin du timer, il terminera son fils de manière prématurée.

Exercice: Built-in or not Built-in

Pour chacune des commandes suivantes implémentez un programme qui effectuera (sans rentrer dans la complexité des arguments bien entendu!) la même opération. Prenez également le temps de comprendre le rôle de ce programme et de jouer avec ses arguments.

```
echo ,rm, cd, ls, date, kill, cp, mv, ln ,cat, touch, pwd, tee, (bonus) find
```

Astuce pour printf/echo utilisez la l'appel « wordexp » quel est l'avantage ?