

## 1516 L2 SEPS TP 2 - Systèmes d'Exploitation et Programmation Système

### Multiprocessus Windows

Vous devez rendre un rapport et vos programmes à la fin du cours sur Espadon. Assurez-vous que votre rapport porte un nom du type "1516\_L2\_SEPS\_TDTP\_2\_MultiProcWin\_votreNom". Mettez dans le rapport des copies d'écran des résultats d'exécutions. Rendez le tout dans un fichier Zip de même nom que précédemment. Pour tous les TP, ne validez votre dépôt sur Espadon qu'une fois le TP totalement terminé; à la fin de chaque séance, déposez un zip de vos travaux avec votre rapport en l'état, mettez-le à jour à chaque séance, et une fois le TP fini, validez.

Le TD/TP s'effectue dans Dev-C++. Les entités C++ dont vous avez besoin pour lancer un processus à partir d'un programme sont les suivantes (cherchez sur le net leur utilité et expliquez les brièvement dans votre rapport!):

- `#include <windows.h>`
- `STARTUPINFO` si
- `PROCESS_INFORMATION` pi
- `ZeroMemory`
- `CreateProcess("programToRun", "commandLine", NULL, NULL, FALSE, 0, NULL, NULL, &si, &pi)`
- `WaitForSingleObject(pi.hProcess, INFINITE)`
- `CloseHandle(pi.hProcess)`
- `CloseHandle(pi.hThread)`
- `Sleep(t_En_Milli_Secondes) (c++)`
- `sleep(t_En_Milli_Secondes) (c)`
- `WaitForMultipleObjects`

## 1) "Mec, elle est ou ma caisse ?"

- Faites un petit programme console en C appelé "DisrupteurDimensionnel", qui effectue un décompte temporisé avant explosion.

- Faites un petit programme console en C++ appelé "JessieEtChester" qui crée un processus (ce qui revient à le lancer) à partir de l'exécutable "DisrupteurDimensionnel.exe" et attend qu'il ait fini en le suppliant de s'arrêter.

Les exemples de sorties des programmes sont donnés ci-contre:

```
D:\Octopussy\2015 - 01 - Janvier\Cours Système\Programmation\C\HelloWorld.exe

Hello World started !!!

Screw the universe !!!

-> Hello World : count down : 10
-> Hello World : count down : 9
-> Hello World : count down : 8
-> Hello World : count down : 7
-> Hello World : count down : 6
-> Hello World : count down : 5
-> Hello World : count down : 4
-> Hello World : count down : 3
-> Hello World : count down : 2
-> Hello World : count down : 1

Bang ! <the big one !>
Appuyez sur une touche pour continuer... _
```

```
Beginning
Test: please, let me leave !!!!

Hello World started !!!

Screw the universe !!!

-> Hello World : count down : 10
Test: please, let me leave !!!!
Test: please, let me leave !!!!
-> Hello World : count down : 9
Test: please, let me leave !!!!
Test: please, let me leave !!!!
Test: please, let me leave !!!!
-> Hello World : count down : 8
Test: please, let me leave !!!!
Test: I beg you !!!!
-> Hello World : count down : 7
Test: I beg you !!!!
Test: I beg you !!!!
Test: I beg you !!!!
Test: I beg you !!!!
-> Hello World : count down : 6
Test: I have kids !!!!
Test: I have kids !!!!
Test: I have kids !!!!

Goodbye, cruel world...
-> Hello World : count down : 5
-> Hello World : count down : 4
-> Hello World : count down : 3
-> Hello World : count down : 2
-> Hello World : count down : 1

Bang ! <the big one !>
```

Vous pouvez changer les supplications...

## 2) Pierre-Paul-Jacques

- Faites un petit programme console en C appelé "Fils", qui écrit une ligne à l'écran en comptant et en affichant son premier argument reçu en ligne de commande.
- Faites un petit programme console en C++ appelé "Pere" qui crée dans une boucle trois processus de noms *Pierre*, *Paul* et *Jacques* (stockés dans un tableau, par exemple), en leur passant à chacun leur nom par le paramètre de ligne de commande de *CreateProcess*, et attend qu'ils aient tous les trois fini leur exécution.

**Hint:** pensez à sauvegarder les "handles" dans des tableaux pour pouvoir fermer les processus proprement.

Les exemples de sortie du programme sont donnés ci-dessous:

```
H:\2015 - 01 - Janvier\Cours Système\Programmation\C++\MultiProcesses\Son\Son.exe
Executing process named Pierre
I am Pierre <0> and I love to do stuff ?
I am Pierre <1> and I love to do stuff ?
I am Pierre <2> and I love to do stuff ?
I am Pierre <3> and I love to do stuff ?
I am Pierre <4> and I love to do stuff ?
```

```
H:\2015 - 01 - Janvier\Cours Système\Programmation\C++\MultiProcesses\Father\Father.exe
Beginning
Task 'Pierre' created
Executing process named Pierre
Task 'Paul' created
Executing process named Paul
Task 'Jacques' created
Waiting for executions
Executing process named Jacques
I am Pierre <0> and I love to do stuff ?
I am Paul <0> and I love to do stuff ?
I am Jacques <0> and I love to do stuff ?
I am Pierre <1> and I love to do stuff ?
I am Paul <1> and I love to do stuff ?
I am Jacques <1> and I love to do stuff ?
I am Pierre <2> and I love to do stuff ?
I am Paul <2> and I love to do stuff ?
I am Jacques <2> and I love to do stuff ?
I am Pierre <3> and I love to do stuff ?
I am Paul <3> and I love to do stuff ?
I am Jacques <3> and I love to do stuff ?
I am Pierre <4> and I love to do stuff ?
I am Paul <4> and I love to do stuff ?
I am Jacques <4> and I love to do stuff ?

Processes finished, closing
Processes closed
Appuyez sur une touche pour continuer...
```

### 3) My\_Sweet\_Little\_Multi\_Process

Faites un couple (ou plus!) de programmes de votre cru et essayez avec d'illustrer une fonction Windows traitant des processus ou de la programmation système qui n'a pas été vue. Comment récupérer les résultats des processus fils?