**Ejercicios 12-feb-19**

**Programas concurrentes (padre-hijos-nietos).**

**OBJETIVO.**

Conocer los diferentes formas en que se arranca un proceso (padre o hijo), a través de los APIs (funciones, métodos) del lenguaje de programación (C++); tal y como lo hace el sistema operativo.

En el caso de los APIs de programación se emplea, por ahora, la cómoda, cara, y poco óptima función ***system(commandLine)***.

Los encabezados (headers) de C y C++ los puede encontrar en el archivo CHeaders.xlsx.

Así, p.e. si el programa ***primero.exe*** contiene a ***system(“programa.exe arg1 arg2 …” )*** entonces el árbol a partir de nuestro shell *tcsh* quedará de la siguiente forma:

**tcsh>**

**primero.exe**

**sh**

**program.exe**

En este caso el API ***system(“program.exe arg1 arg2 …” )***, al momento de ser ejecutado crea de manera final un árbol hijo con ***sh -- program.exe***, siendo el shell ***sh*** hijo del programa que lo contiene, y ***program.exe*** hijo del shell ***sh***.

Para estos dos ejercicios usaremos los programas: *argumentos.cc* y *cuatro.cc*.

No se olvide de actualizar el path de su terminal de Linux con la ejecución de ***source .midir***, desde su *home directory sisops*.

1. (\*\*) Ejecución de programas hijos y nietos vía la función *system(commandLine)*.

Modifique *argumentos.cc* para que ahora, después de la impresión de los argumentos (el ciclo) invoque a la función ***system(“ps -l” )***.

Ejecute el programa *argumentos.exe* y de lo arrojado por el comando “ps –l”, desde dentro del programa, construya el siguiente árbol incluyendo por línea el nombre del programa, su PID y su PPID correspondiente:

**tcsh>**

**\_\_\_\_\_\_\_\_\_\_\_**

**................\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_** (alguna estará de más o de menos)

1. (\*\*) Ahora desde el programa *cuatro*, mande ejecutar como descendiente al programa *argumentos* con los argumentos *hola* y *345*. Dentro de *cuatro.cc* deberá agregar *system(“argumentos.exe hola 345”)*, lo cual permitirá ver todo el árbol de ejecución, desde *tcsh*.

Ejecute el programa *cuatro.exe* y de lo arrojado por el comando “ps –l”, desde dentro del programa descendiente, construya el siguiente árbol incluyendo, por línea el nombre del programa su PID y su PPID correspondiente:

**tcsh>**

**\_\_\_\_\_\_\_\_\_\_\_**

**................\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_** (alguna estará de más o de menos)