**Ejercicios BG 26-mar-20**

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

**OBJETIVO.**

Conocer los diferentes formas en que se arranca un proceso (padre o hijo), vía línea de comando y / o 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(“program.exe arg1 arg2 …” )***. También usaremos comandos como ***echo*** y redirectores como **>**, **>>**.

Para este ejercicio usaremos el código del programa de la clase antepasada: *argumentos.cc*.

No se olvide de actualizar el PATH de su terminal de Linux con la ejecución de ***source ~/.mypath***; obviamente ***.mypath*** se encuentra en su *home directory sisops*, que construimos anteriormente.

1. (\*\*) Desarrolle el programa *argucmds* (con el código de *argumentos.cc*) en *cc*, que mande ejecutar comandos de Linux a través de la función *system()*, para que cada uno de los argumentos pasados al momento de ejecutar el programa (en la línea de comando), sean escritos en el archivo de texto *resultados.txt*, como se indica más abajo. Como última línea del archivo de texto deberá aparecer los nombres de los autores.

Los encabezados (headers) de C y C++ los puede encontrar en el archivo *CHeaders.xlsx*, de la clase pasada.

Hint: se recomienda usar *sprintf(…)* para formar strings con formato, en buffers fijos, para usarlos en *system( )*. Más abajo viene un ejemplo de uso de *sprintf(…)*.

Por ejemplo:

tcsh> argucmds.exe hola como estan todos ustedes

El archivo de texto *resultados.txt* contendría al final

argucmds.exe

hola

como

estan

todos

ustedes

Fulano, Mengano y Sutano

Así desde el programa, usted mandaría escribir al archivo *resultados.txt*, cada uno de los argumentos recibidos a través de la línea de comando de ejecución de *argucmds*.

|  |
| --- |
| Programa de ejemplo de uso de la función *sprintf(…)* |
| #include <stdio.h>  int main () {  char buffer [50];  int n, a=5, b=3;  n = sprintf (buffer, "%d mas %d es %d", a, b, a+b); // buffer con string formateado  printf ("[%s] es un string con %d caracteres\n", buffer, n);  return 0;  } |
| Salida por *stdout* |
| [5 mas 3 es 8] is a string 12 caracteres |