

# Ej. Procesos. Aleatorios. Enunciado

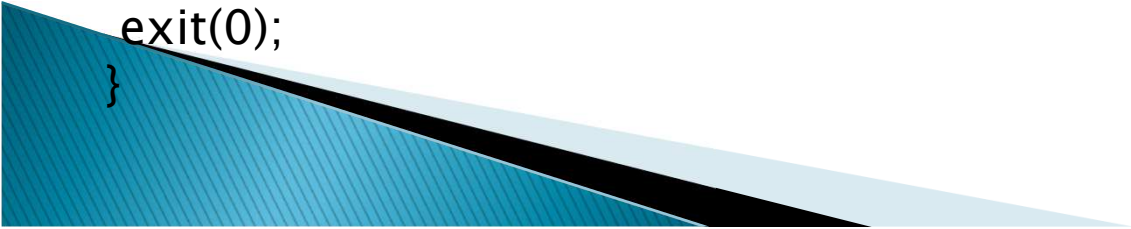
- crear un proceso hijo que esté encargado de generar números aleatorios. Para su creación usar cualquier lenguaje de programación, generando el ejecutable correspondiente.
- Este proceso hijo escribirá en su salida estándar un número aleatorio del 0 al 10 cada vez que reciba una petición de ejecución por parte del padre.
- El proceso padre lee líneas de la entrada estándar y por cada línea que lea solicitará al hijo que le envíe un número aleatorio, lo leerá y lo imprimirá en pantalla.
- Cuando el proceso padre reciba la palabra “fin”, finalizará la ejecución del hijo y procederá a finalizar su ejecución.



# Ej. Procesos. Aleatorios. Hijo C

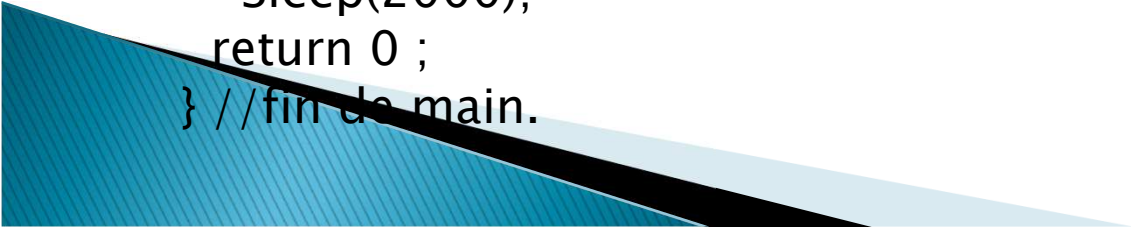
```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <unistd.h>
int main(int argc, char **argv) {
    char buf[1024];
    int num;

    while(fgets(buf, 1024, stdin) != NULL) {
        num = 1+(int) (10.0*rand()/(RAND_MAX+1.0));
        fprintf(stdout,"%d\n",num);
        fflush(stdout); // IMP: para asegurar que se reciben los
                        datos
    }
    exit(0);
}
```



# Ej. Procesos. Aleatorios. Hijo C++

```
//Ejemplo generar numeros aleatorios del 1 al 20.
#include <iostream>
#include <cstdlib> //tiene la definicion de la funcion RAND (random)
#include <windows.h> // permite usar Sleep()
#include <time.h>
using std::cout; using std::endl; using std::cin;
int main () {
    int cantidad=1;
    double numaleatorio;
    //cout << "Cuantos números quiere generar?" << endl;
    //cin >> cantidad;
    for (int contador =1 ; contador <= cantidad ; contador++) {
        srand (time(NULL)); //Utilizo la hr del sistema como semilla
        numaleatorio = 1+(rand() %20);
        cout << numaleatorio;
        cout << endl; //salto de linea para separarlos
    }
    Sleep(2000);
    return 0 ;
} //fin de main.
```



# Ej. Procesos. Aleatorios. Padre

```
import java.io.BufferedReader; import java.io.IOException; import  
java.io.InputStreamReader;
```

```
import java.io.PrintStream;
```

```
public class Aleatorios {
```

```
    public static void main(String[] args) {  
        String line;
```

```
        try{
```

```
        Process hijo = new ProcessBuilder("./aleatorios").start();
```

```
        BufferedReader br = new BufferedReader(new
```

```
        InputStreamReader(hijo.getInputStream()));
```

```
        PrintStream ps = new PrintStream(hijo.getOutputStream());
```

```
  
        BufferedReader in = new BufferedReader(new  
        InputStreamReader(System.in));
```

```
        while ((in.readLine().compareTo("fin") != 0 ) {  
            ps.println(""); //No es necesario escribir nada  
            ps.flush(); // Asegura que los datos se han enviado  
            if ((line = br.readLine()) != null) {  
                System.out.println(line);
```

```
            }
```

```
        }
```

```
        System.out.println("Finalizando");
```


```
    } catch (IOException e) {  
        System.out.println("Error  
        ocurrió durante la ejecución.  
        Descripción del error: " +  
        e.getMessage());  
    } }
```

# Memorias. Normativa de Prácticas

- ▶ Todas las prácticas se acompañaran de los siguientes documentos:
  - Carpeta del proyecto
  - Fichero fuente
  - Ficheros de pruebas de Entrada y Salida
  - Fichero de Memoria en formato Word.
  - Fichero de Memoria en formato PowerPoint para defensa de la práctica (Sólo cuando sea requerido)



# Memorias. ESQUEMA DE MEMORIA

- ▶ Introducción
    - Antecedentes
    - Entorno de Desarrollo
  - ▶ Enunciado Práctico
    - Explicación del Problema a resolver
  - ▶ Solución
    - Descripción y Justificación de la solución.
    - Código fuente de la Aplicación e Interfaces
  - ▶ Explicación de
    - Algoritmos
    - Clases
    - Estructuras de Datos
  - ▶ Listado de pruebas
    - Entrada
    - Ejecución
    - Salida
  - ▶ Conclusiones
- 

# Prácticas Procesos

- ▶ Programar una aplicación que cree 6 subprocesos hijos de forma que cada uno de ellos genere 100 números aleatorios entre 1 y 49 y haga llegar al padre el número entero procedente de la media de los que generó.
- ▶ El padre una vez reciba todos los números imprimirá por pantalla los números elegidos para la primitiva, indicando que proceso hijo creo cada número.

