

Programación de Servicios y Procesos

Programación Multiproceso

2º D.A.M
I.E.S Los Montecillos

21 de noviembre de 2021

Se pide desarrollar una aplicación para implementar un juego de entrenamiento para mejorar nuestra agilidad en cálculo mental. Esta, irá mostrando una serie de operaciones aritméticas de forma aleatoria, y esperará a que el usuario introduzca la respuesta, en cuyo caso, irá puntuando o penalizando según la corrección de la misma.

En concreto, se pedirán diez operaciones aritméticas simples, esto es, binarias, en las que intervengan un solo operador aritmético binario (+, -, *, /) con sus respectivos operadores, que serán valores numéricos enteros con un rango comprendidos entre 0 y 10 (ambos inclusive, aunque habrá que tener en cuenta los posibles casos especiales, como la división por cero).

El funcionamiento se basará en los siguientes requisitos:

- Cada operación propuesta, correctamente contestada, sumará 10 puntos.
- Las operaciones tendrán un tiempo máximo de resolución variable, que dependerá del tipo de operación pedida. Estos tiempos se muestran en la tabla.
- Mientras no expire el tiempo máximo dado a cada operación, se aceptarán todas las respuestas posibles, pero cada intento restará 1 punto.
- Una vez expirado el tiempo límite, la pregunta propuesta no se podrá resolver. Esta situación NO afectará a la puntuación (no suma ni penaliza) . *IMPORTANTE* : Aunque haya vencido el tiempo, se deberá dar una respuesta para poder pasar a la siguiente propuesta de cálculo a resolver. Esto es debido al modo de funcionamiento bloqueante de la función *scanf()*.
- En cada intento de resolución, se mostrará si la respuesta es correcta o incorrecta. En caso de ser correcta, se pasará a la siguiente pregunta y se actualizará la puntuación. En caso contrario (respuesta incorrecta), se indicará y se tendrá la opción de volver a contestar, siempre que se esté dentro del tiempo asignado.
- Al completar los diez ejercicios propuestos, se mostrará la puntuación final obtenida y la aplicación finalizará.

Tiempos de cada operación	
Operación	Tiempo
Suma (+)	5 segundos
Resta (-)	6 segundos
Multiplicación (*)	7 segundos
División (/)	8 segundos

- Cada operación propuesta estará precedida de un número de orden, que se repetirá en los intentos que se realicen.

A continuación se muestra un ejemplo de un posible escenario de funcionamiento:

```

1  -----
2  | AritGame |
3  -----
4  1. -> 2 + 3 = 5
5      [Correcta]
6
7  2. -> 4 * 5 = 21
8      [Incorrecta] puedes intentarlo de nuevo...
9
10 2. -> 4 * 5 = 22
11     [Incorrecta] puedes intentarlo de nuevo...
12
13 2. -> 4 * 5 = TIME OUT (Debes responder para continuar)
14 20
15
16 3. -> 5 - 7 = -2
17     [Correcta]
18
19 4. -> 7 - 7 = 0 TIME OUT (Debes responder para continuar)
20
21 5. -> 6 - 6 = 0
22     [Correcta]
23
24 6. -> 3 - 6 = -3
25     [Correcta]
26
27 7. -> 7 - 5 = 2
28     [Correcta]
29
30 8. -> 0 * 4 = 0
31     [Correcta]
32
33 9. -> 7 / 8 = 1
34     [Incorrecta]
35

```

```
36 9. -> 7 / 8 = 0
37      [Correcta]
38
39 10. -> 6 - 7 = -1  TIME OUT (Debes responder para continuar)
40 -1
41
42 Partida finalizada
43 Puntuacion Obtenida : 66 puntos
```

NOTA 1 : Aspectos relacionados con la resolución del problema

- La aplicación está constituida por dos procesos independientes, denominados *aritgame*, y *timecontroller*, que se comunicarán entre sí mediante las señales que se consideren oportunas.
- El proceso *aritgame* es el responsable de generar las preguntas, aceptar las posibles respuestas del usuario, comprobarlas y controlar las puntuaciones.
- El control del tiempo se realiza en otro proceso denominado *timecontroller*, que será creado en cada una de las operaciones propuestas, y recibirá una serie de parámetros como el tiempo y el PID del proceso *aritgame*.
- Una vez finalizado el tiempo en el proceso *timecontroller*, este lo notificará al proceso *aritgame* mediante la señal que se considere más adecuada, y el proceso morirá de forma natural.
- Cada pregunta crea el controlador de tiempo.
- En caso de respuesta correcta, hay que abortar el controlador de tiempo, para que se pueda formular la siguiente pregunta.

NOTA 2 : Aspectos relacionados con la entrega del examen

- El código fuente del proceso *aritgame* estará en el fichero *aritgame.c*, y el del proceso *timecontroller*, en el fichero *timecontroller.c*.
- Una vez acabado el examen, ambos ficheros fuente se subirán a la plataforma Moodle en un solo fichero comprimido con el formato *apellido1_nombre.zip*.
- No se evaluará el código entregado con errores de compilación.
- El código fuente debe contener una cabecera con los datos del alumno/a, asignatura, nombre y apellidos, y fecha.

NOTA 3 : Reto. ¿Podría solucionarse el problema producido por la lectura de datos con *scanf()* cuando se produce un timeout? En caso afirmativo, detallar la solución propuesta y aplicarla al ejercicio, para que cuando se produzca un timeout, se formule directamente la siguiente pregunta (sin tener que dar una respuesta que no se comprobará).

