



aprenderaprogramar.com

Toma de decisiones. Centinelas para control de bucles. Procesado del centinela como dato.

(CU00166A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 65 del Curso Bases de la programación Nivel I

24

SEÑALEROS (CENTINELAS) PARA CONTROL DE BUCLES

Como herramienta para la programación, el uso de variables como señaleros (centinelas) tiene diversas aplicaciones como aportación de información, apoyo a la toma de decisiones, control de bucles, etc. Destaca la importancia del uso de señaleros para el control de bucles. Hasta ahora habíamos visto:

- Control de bucles conociendo de antemano el número de iteraciones, por ejemplo: 20 camiones diarios, ...
- Control de bucles a través de una variable de control (contenida en un archivo, proporcionada por el usuario, etc.), por ejemplo, número n de generaciones en una sucesión familiar.

Opciones a las que hemos de añadir el control de bucles a través de centinela o señalero. Para el correcto desarrollo de los bucles habrán de tenerse en cuenta aspectos ya tocados en otros apartados (contadores, etc.) como cuál es el valor inicial del señalero, si éste cumple otras funciones aparte de controlar el final del bucle, cómo se organizan y cómo se denomina a los señaleros, en caso de varios ciclos cuál es el estado final de la variable de cara a comenzar otro, etc.

Aparte de cuestiones generales, hay cuestiones específicas relacionadas con los señaleros que conviene tener muy en cuenta. Estas son:

1. Procesado o no del señalero como dato.

El programador ha de prever si el señalero funciona como dato de pleno derecho y debe ser procesado o únicamente como señal de terminación y no debe ser procesado. Puede ser igual de desastroso procesar un dato que no procede como dejar de procesar uno que sí procede.

Ejemplo:

Una empresa ha sometido a sus operarios a un control rutinario de peso y guarda los resultados en un archivo como un array de datos $Peso(x)$ donde x es el número de empleado. Se desea calcular el peso medio de los empleados, sabiendo que como cierre hay un señalero con valor -10 . Supongamos que se plantea el siguiente algoritmo:

[Ejemplo aprenderaprogramar.com]

1. Inicio

2. Hacer

$i = i + 1$

Leer $Peso(i)$

$Suma = Suma + Peso(i)$

Repetir Mientras $Peso(i) >= 0$

3. $Media = Suma / i$

4. Mostrar "La media es", Media

5. Fin

En un análisis rápido vemos un contador, un acumulador y un bucle controlado por la condición $Peso(i) \geq 0$. Nada extraño a priori. Valorando con detenimiento el algoritmo ha de llegarse inmediatamente a la conclusión de que es erróneo por cuanto el señalero está interviniendo como dato válido, pero no lo es. Si analizamos el diseño del bucle observamos que la condición de salida se encuentra al final del mismo. Se extrae y procesa el dato antes de evaluar si es pertinente y esto da lugar a que se extraiga y procese el centinela, hecho que no debería ocurrir. Supongamos que el archivo contiene los pesos siguientes:

$Peso(1) = 90, Peso(2) = 93, Peso(3) = 88, Peso(4) = -10$

El algoritmo calcularía: $Suma = 90 + 93 + 88 + (-10)$

$Media = Suma / i = 261 / 4 = 65,25$

Obviamente el valor correcto es: $Media = (90 + 93 + 88) / 3 = 90,33$

Aunque el algoritmo anterior no vendría nada mal para reducir los índices de obesidad, es obviamente desastroso y para más inri se comporta como “false friend” o falso amigo. Parece que hace algo pero nos perjudica aún más que un fallo evidente. Es así porque si el resultado fuera un error evidente, como una media de 3 Kg de peso, se detecta fácilmente que existe un error. En cambio una media como 65,35 Kg de peso podría creerse que es correcta, sobre todo si el número de datos es elevado. Esto llevaría a su vez a un error de más difícil detección, o simplemente, a la no detección del error hasta pasado un tiempo, cuando el daño causado puede ser notable.

Evitar circunstancias como éstas depende del programador. Sobre todo, en cuanto a diseñar correctamente algoritmos y, como veremos más adelante, en lo que respecta a utilizar técnicas de verificación que garanticen su correcto funcionamiento. En cuanto al caso que nos ocupa, veamos cómo solucionarlo con un algoritmo que trabaje correctamente.

[Algoritmo aprenderaprogramar.com]

1. Inicio

2. $i = 1$

3. Leer $Peso(i)$

4. **Mientras $Peso(i) \geq 0$ Hacer**

$Suma = Suma + Peso(i)$

$i = i + 1$

 Leer $Peso(i)$

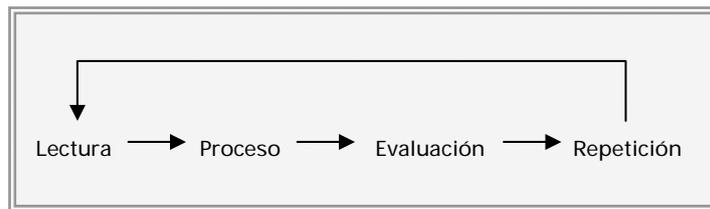
Repetir

5. $Media = Suma / (i - 1)$

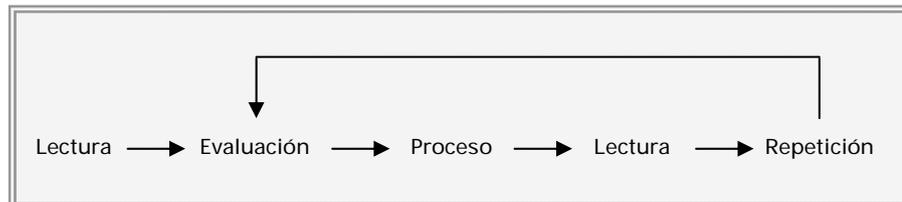
6. Mostrar “La media es”, $Media$

7. Fin

Hemos cambiado el esquema que se mostró erróneo:



Por este otro:



Las novedades son:

- Se lee el primer dato para evaluar si se entra en el bucle. Esto sucederá siempre, excepto si el señalero de terminación es el primer dato. Esto a su vez podría darse bien porque no hayan datos que procesar o por un error al procesar o archivar datos en el fichero.
- Todo dato es evaluado antes de ser procesado, de modo que se evita que un dato improcedente se cuele junto a los datos válidos.
- Los datos válidos son los datos extraídos menos uno. Esto es así porque el centinela es extraído para evaluarlo, pero no es válido. Por ello la media se calcula como $Suma / (i - 1)$.

2. Validez del señalero para el cumplimiento de su función.

Una vez se ha decidido el uso de un señalero hemos de plantearnos qué valor o rango de valores usar para un adecuado cumplimiento de su función. El objetivo de esta elección es que no pueda haber ningún otro dato de igual valor al del señalero, ya que en ese caso corremos el riesgo de pensar que ha llegado la señal cuando realmente no ha llegado. Clasificaremos los señaleros en:

- Indiscutiblemente válidos:** es imposible que un dato tome el mismo valor que el señalero.
- Posiblemente válidos:** es previsible que ningún dato tome el mismo valor que el señalero, pero no imposible.
- No válidos:** es muy probable que un dato tome el mismo valor que el señalero, pues éste está plenamente dentro del rango de valores que toman los datos.

A la hora de programar trataremos de usar siempre señaleros indiscutiblemente válidos, pues el uso de uno posiblemente válido introduce incertidumbre, factor no deseable en la programación. Lo anteriormente expuesto debe tenerse en cuenta tanto para señaleros que se procesan como dato de pleno derecho como para los que no.

Consideremos el caso de los señaleros como final de una serie de datos de peso de personas.

Podríamos señalar:

- Señaleros indiscutiblemente válidos: – 10, – 99, – 100, – 999, 999, 1000.
- Señaleros posiblemente válidos: 5, 10, 15, 150, 200, 250.
- Señaleros no válidos: 60, 75, 90.

Esta clasificación deriva del análisis del rango de datos previsible. Sabemos que una persona no puede tener un peso negativo ni pesar 999 ó 1000 kg. Estos valores nos pueden indicar el final de la serie de datos: indiscutiblemente.

Creemos que una persona no va a pesar 15 ó 150 Kg. Pero estamos, cuando menos, en los extremos de un rango de datos posible. ¿Para que quedarnos con la duda de estar “en el extremo” del rango de datos cuando podemos estar seguros situándonos “claramente fuera” del rango de datos? ¿Qué ocurre si una persona ha engordado y llega a los 150 Kg?

Sabemos que los pesos más habituales estarán comprendidos entre 50 y 100 kgs, por lo que no debemos usar ningún valor como 60, 75 ó 90 como señaleros.

En el caso de que el señalero sea un dato más a procesar consideramos las categorías ya definidas:

- Señaleros indiscutiblemente válidos.
- Señaleros posiblemente válidos.
- Señaleros no válidos.

Con señaleros que se procesan es más complicado buscar la indiscutible validez de la variable como señalero, por lo que en ocasiones habremos de conformarnos con que la probabilidad de coincidencia sea tan baja que a efectos prácticos se pueda considerar nula. Una fiabilidad del 99,999 % se podrá considerar aceptable, una del 99 % quizás aceptable con algún mecanismo de seguridad y valores inferiores no recomendables. Es el programador quien habrá de decidir si a su juicio la fiabilidad en el uso de un dato como señalero es suficiente o no. Y el criterio reviste cierta subjetividad.

Consideremos una serie de datos de diversa índole, de cara a valorar si uno de esos datos puede ser señalero:

- a) Números de serie de una partida de palas cargadoras enviada por el fabricante: Indiscutiblemente válido. El fabricante no asignará nunca dos números de serie iguales.
- b) Número de cuenta bancaria: indiscutiblemente válido.
- c) Número de DNI de una persona: indiscutiblemente válido.
- d) Superficie en metros cuadrados de una parcela urbana (precisión de cuatro decimales): posiblemente válido. Estudiar circunstancias.
- e) Nombre de una persona: dudoso. Quizás utilizable con un mecanismo de seguridad.
- f) Ingresos brutos anuales de una persona en euros (precisión de un euro). Dudoso. Difícilmente utilizable.
- g) Número de pasajeros de un avión: no válido. Difícilmente distinguible un dato de un señalero.
- h) Peso de una persona: no válido. Difícilmente distinguible un dato de un señalero.

Próxima entrega: CU00167A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59