

# Patriot Missile Software Problem

Universidad Icesi

Departamento Ciencias Físicas y Exactas

Facultad de Ingeniería, Diseño y Ciencias Aplicadas

PRIMER PARCIAL - ARQUITECTURA DE COMPUTADORES

Nombre:	Fecha: 31 agosto 2024
Código:	Duración: 1.5 Horas

No se permite el uso ningún elemento electrónico de almacenamiento de datos ni de comunicación, Smartphone o computador durante el examen ni tampoco ningún tipo de dispositivos de audio.

Si usa la calculadora de su smartphone por favor, ponga su dispositivo en modo avión

## Patriot Missile Software Problem

El problema del misil Patriot es un ejemplo clásico de cómo la precisión limitada de los formatos de numeración de los sistemas de cómputo puede conducir a consecuencias catastróficas en sistemas críticos. Suponga que en el caso específico del misil Patriot, el sistema de control utilizaba números de punto fijo para representar el tiempo transcurrido en décimas de segundos desde un punto de referencia.

El sistema de control del misil Patriot, tenía que realizar cálculos para predecir la posición futura del objetivo y dirigir el misil para interceptarlo. Sin embargo, debido a la precisión limitada de los números los cálculos acumulaban pequeños errores de redondeo con el tiempo.

En el caso específico que condujo al accidente, el sistema había estado funcionando continuamente durante 100 horas aproximadamente antes del evento. Durante este tiempo, los errores acumulados en los cálculos de tiempo resultaron en una desviación significativa en la posición esperada del objetivo. Cuando el sistema intentó realizar la corrección necesaria para interceptar el objetivo, esta desviación no fue correctamente calculada, lo que llevó a que el sistema Patriot no logró rastrear y destruir un misil Scud lanzado por Irak hacia una base militar en Arabia Saudita el 25 de febrero de 1991. El misil Scud impactó en la base, causando la muerte de 28 soldados estadounidenses y heridas a otros cientos.

A las 100 horas de funcionamiento el error acumulado del temporizador (variable Time) fue de 0.3433 segundos aproximadamente, y como la velocidad del misil SCUD es de 1676 m/s, generó un error en la distancia de detección de 687 metros ( $d=v*t$ ), pero el margen de error debería ser menor a 100 m.

Se sabe que el error (E) del formato punto fijo usado era de  $2^{-20}$ , entonces dando que  $(2^{-20}) \times 100 \text{ horas} = (2^{-20}) \times 3600 \times 100 = 0.3433 \text{ segundos}$

- [25%] Si el sistema debió operar 100 horas seguidas con un error acumulado de 0.3433, indique qué formato de punto fijo pudieron haber usado los ingenieros del Patriot en 1991. (determine la cantidad mínima de bits).
- [75%] Suponga que usted viajara en el tiempo al año 1990, y le propone a los ingenieros del proyecto Patriot que en lugar de usar el formato de punto fijo usen el formato de punto flotante IEEE 754 de 32 bits, entonces suponga que usted propone el siguiente código:

```

float Temporizador = 0;
float deltaTime = 0.10;
float TimeEnd = 360000;
int counter = 0;
int iteracionesTotales = int(TimeEnd /deltaTime); //3600000 (TimeEnd*10)
float distance;
int main()
{
    while (counter< iteracionesTotales){

        Temporizador = Temporizador + deltaTime;
        counter=counter+1;
    }
    std::cout<<"Temporizador final = " <<Temporizador << "\n";
    std::cout<<"error (s) = " <<TimeEnd-Temporizador<< "\n";
    return 0;
}

```

**B.1 [25%]** Si float *TimeEnd* fuera igual a 36000= 0x470CA000. (36.000 segundos equivale a 10 horas)

Según lo visto en el laboratorio, escriba a partir de cualquier dirección de la siguiente zona de memoria como estaría almacenada la variable *TimeEnd*:

Escriba la representación normalizada de *TimeEnd* = 0x470CA000:

1, \_\_\_\_\_ x2—

Escriba la representación normalizada de *deltaTime* (0.10)

1, \_\_\_\_\_ x2—

0x006FF85D	
0x006FF85E	
0x006FF85F	
0x006FF860	
0x006FF861	
0x006FF862	
0x006FF863	
0x006FF864	
0x006FF865	
0x006FF866	

**B.2 [50%]** Calcule el valor final de la variable *Temporizador* cuando se termine de ejecutar el programa (es decir con *EndTime*=360000) y el error acumulado, sabiendo que a las **10 horas** de funcionamiento continuo (*TimeEnd* = 36000) el error acumulado es de **41.6523** y la variable *counter* es **360000**

- ¿Su propuesta de usar el formato de simple precisión sería mejor al de punto fijo usado en 1991?
- Calcule el valor de la variable Counter. *Temporizador*.
- Cuántos metros de error de detección del misil habría a las 100 horas.

fin -----

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1048576	524288	262144	131072	65536	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

\*Muestre el procedimiento de todos sus cálculos, las respuestas sin procedimiento no se tendrán en cuenta como válidas.

\*\* Algunos hechos y datos dados en el problema son ficticios y no corresponden a los hechos reales del evento de 1991, pero el error si fue de 0.3433 s

# Patriot Missile Software

## 100 hours

Error acumulado = 0.3433 s

Velocidad = 1676 m/s

→ Punto fijo

$$(2^{-20}) \times 100 \text{ h} = 2^{-20} \times 3600 \times 100 \\ = 0.3433$$

A. Formato de punto fijo usado



Teniendo en cuenta que  
 $(E = \frac{p}{2} \wedge p = 2^{-k}) \rightarrow E = \frac{2^{-k}}{2}$

Tambien podemos expresar  
 $E = 2^{-k-1}$

Y, sabemos que  
 $E = 2^{-20}$

Por tanto  $p = 2^{-19}$

Por lo que concluimos  
 $\therefore k = 19$

Para saber  $n$ , debemos tener en cuenta el rango

$$\text{Rango} = 100h = 360000 \text{ s}$$

\* Número máximo

$$111111\dots1, 111\dots1 \rightarrow \text{sin } C_2 \\ 011111\dots1, 111\dots1 \rightarrow \text{con } C_2$$

\* Rango Punto Fijo

$$\text{Rango} = 2^n - 2^{-k}$$

Entonces

$$2^n - 2^{-k} \geq 360000 \\ 2^n \geq 360000 + 2^{-19} \\ n \geq \log_2(360000 + 2^{-19}) \\ n \geq 18,957$$

$$n=18 \rightarrow 2^n = 262144 \quad X \\ n=19 \rightarrow 2^n = 524288 \quad \checkmark$$

Rta / El formato de punto fijo utilizado es

$$Q_{19,19}$$

B1 a. TimeEnd > 36000 = 0x470CA000

$$\text{Exp} = 1000111 = 142 - 127 = 15,$$

$$\therefore 1,000110010100000\ 000000000 \times 2^{15}$$

b. deltaTime → 0,1 Normalizado

0,1 → FP32

$0,1 \rightarrow 0,2$

$0,2 > 0,4$

0,1 → 0,8

$0.8 \rightarrow 1.6$

0.6 > 1.2

0.6 → 1  
0.2 → 0.5

8,2 - 90,

## Normalizar

$$0,0001\overline{10011} \times 2^0$$

$$\Rightarrow 1.19011 \times 2^4$$

$$\rightarrow 1,10011 \times 2^4$$
$$\rightarrow 11001100110011001100110011 \times 2^4$$

$\rightarrow 1,100110011001100110011001 \times 2^1$

$$\therefore 1,100110011001100110011001100 \times 2$$

## → Redondeo

c. 0x006FF85D 00  
0x006FF85E A0  
0x006FF85F 0C  
0x006FF860 47

1 byte → 2 hex → 16 bits  
1 hex → 8 bits

B.2

- la propuesta de usar el formato de precisión simple NO sería mejor que el de punto fijo usado, porque en tan solo 10 horas en FP32 hay un error acumulado de 41.6523 sg, mientras punto fijo en 100 horas tiene un error acumulado de 0.3433

\*Precisión  
 $2^{-k} \cdot 2^e \rightarrow P_{\min} = 2^{-23} \cdot 2^{-126} = 2^{-149}$   
 $P_{\max} = 2^{-23} \cdot 2^{127} = 2^{104}$

\*Error  
00000001 min = -126  
11111110 max = 127  
 $E = 1 \rightarrow e = 1 - 127 = -126$   
 $E = 254 \rightarrow e = 254 - 126 = 127$

deltaTime 1,10011001100110011001100  $\times 2^{-1}$

Temporizador = 36000  $\rightarrow 2^{15}$

1,0001100101000000000000000000  $\times 2^{15}$

Igualamos exponentes

0,0000000000000000000000000000110011  $\times 2^{15}$   
110101  $\times 2^{15}$

Entonces  $\rightarrow \text{deltaTime} = 1,1010 \times 2^{-4}$   
 $= 0,1015625$

Ahora hallamos el número mayor con este exponente

$$1,11111\dots 1 \times 2^{15} = (2 - 2^{-23}) \times 2^{15}$$
$$\approx 65535.996$$

Sabemos que partimos de 36000, por tanto para calcular las iteraciones de este intervalo

$$\frac{65535 - 36000}{0.1015625} = \frac{290806.153}{0.1015625} \approx 290806$$

Entonces, el error es

$$\begin{aligned} e &= (0.1 - 0.1015625) \\ &= -0.0015625 \times 290806 \\ &= -454.38 \end{aligned}$$

Por tanto, el error acumulado es

$$\begin{aligned} &-454.38 + 41.6523 \\ &= -412.727 \end{aligned}$$

Ahora, para el deltaTime de los exp 17 y 18, tendremos el mismo valor, por lo tanto, agruparemos estas iteraciones

$$\begin{aligned} \text{deltaTime} &= 1.1 \times 2^{-4} \\ &= 0.06875. \end{aligned}$$

$$\frac{360000 - 131072}{0.06875} = 332986 \text{ iter}$$

$$e = (0.1 - 0.06875) \times \text{iter} = 10405.8125$$

$$E_{\text{acum}} = -1008.229 + 10405.8125 = 9397.5835$$

$$\text{Temporizador} = 350602.4165 \checkmark$$