

I OBJETIVOS

- Medir y comparar la ejecución de procesos que requieren uso intensivo del CPU en arquitecturas Multi-Core.
- Utilizar la API Windows para la gestión de hilos, y de esta forma optimizar una aplicación que requiere uso intensivo del CPU para procesadores Multi-Core.

II BIBLIOGRAFÍA

- Apuntes de programación paralela en Moodle.
- Manual de las funciones de la API de Windows para la creación, manejo y sincronización de hilos.

III RECURSOS

- Una estación de trabajo con Visual Studio.
- Computadoras con procesadores Multi-Core.

IV ACTIVIDADES

1 ***Cálculo de distancia entre aviones que se mueven en el espacio aéreo***

En un espacio aéreo de 3 dimensiones (x,y,z) como el que se muestra en la Figura 1 hay mil aviones en movimiento. Para evitar colisiones entre los aviones, estos deben estar a una distancia mínima que está en función de su velocidad, si un avión vuela casi al nivel del mar su velocidad es 300 Km/h, si vuela a 42,000 pies vuela a 900 Km/h. Los aviones inician en una posición aleatoria y conforme transcurre el tiempo (cada iteración) estos se mueven en el espacio de acuerdo a una dirección.

Si un avión a detecta que está a una distancia menor que la mínima de otro avión b , entonces se debe enviarles un mensaje de advertencia a ambos aviones.

El problema de los aviones es un problema de cómputo intensivo ya que requiere en cada iteración calcular la distancia de cada avión con respecto a todos los demás que están en el espacio, de manera que si son 1000 aviones, cada iteración requiere

calcular $(1000^2 + 1000)/2$ distancias basándonos en las posiciones x, y, z de cada avión.

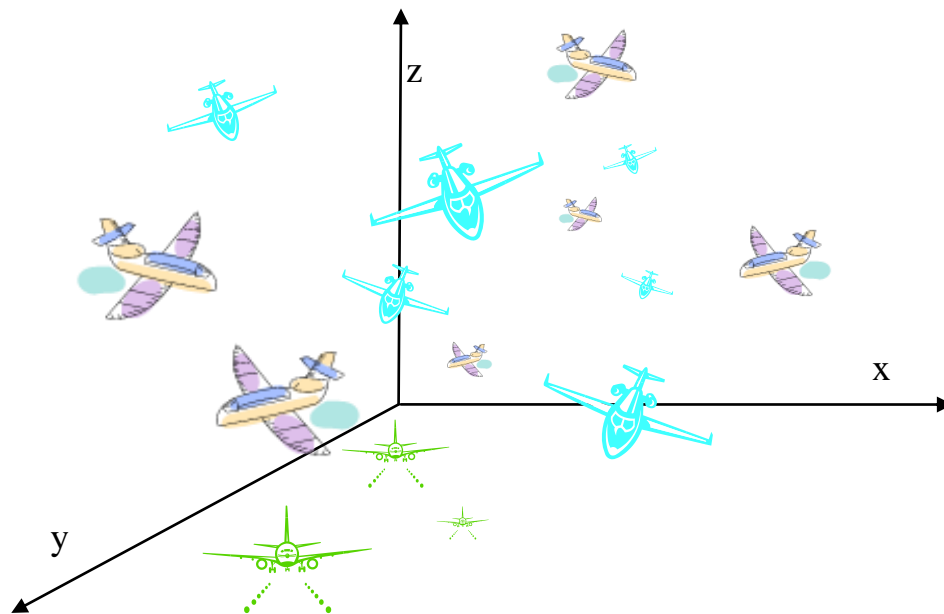


Figura 1. Aviones en el espacio aéreo

1.1 Versión serial

El Ejemplo 1 es una versión serial donde cada 10 iteraciones se muestra en pantalla el total de advertencias acumuladas.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define PI 3.14159

#define SIN(a) sin(a*PI/180.0)
#define COS(a) cos(a*PI/180.0)

#define MAX 1000
#define MINSPEED 300.0 // Km/h
#define MAXSPEED 900.0 // Km/h
#define MAXALT 13000.0 // Metros
```

```
#define MINDIST 4000.0 // Distancia mínima en metros entre aviones si estos
están

struct AVION
{
    double pos_x;
    double pos_y;
    double pos_z;
    double currspeed;
    double dir;
    double incl;
    int warnings;
};

struct AVION avion[MAX];

double dist(struct AVION a, struct AVION b)
{
    double dist_x, dist_y, dist_z, dist;

    dist_x=a.pos_x-b.pos_x;
    dist_y=a.pos_y-b.pos_y;
    dist_z=a.pos_z-b.pos_z;

    dist=sqrt(dist_x*dist_x+dist_y*dist_y+dist_z*dist_z);
    return(dist);
}

double distmin(struct AVION a, struct AVION b)
{
    double speed;
    double distmin;
    if(a.curspeed>b.curspeed)
        speed=a.curspeed/3.6;
    else
        speed=b.curspeed/3.6;

    distmin=MINDIST+speed*speed*1.6;
    return(distmin);
}

void inicializa_aviones(struct AVION *x)
{
    int i;
    for(i=0;i<MAX;i++)
    {
        x[i].pos_x=1000*(5000.0 - (double) (rand()%10000));
        x[i].pos_y=1000*(5000.0 - (double) (rand()%10000));
        x[i].pos_z=(double) (rand()%13000);
        x[i].curspeed=MINSPEED+x[i].pos_z/21;
        x[i].dir=(double) (rand()%360);
    }
    return;
}
```

```
// Se actualiza cada 1/1000 segundos
void actualiza_avion(struct AVION *a)
{
    if(a->pos_z<13000.0)
        a->pos_z+=0.001;

    a->currspeed=MINSPEED+a->pos_z/21;

    a->pos_x+=0.001*(a->currspeed/3.6)*SIN(a->dir);
    a->pos_y+=0.001*(a->currspeed/3.6)*COS(a->dir);

    return;
}

int main()
{
    int n;
    int i,j;
    int totwarnings=0;
    clock_t t_inicial,t_final;

    inicializa_aviones(avion);

    t_inicial=clock();
    for(n=0;n<2000;n++)
    {
        for(i=0;i<MAX-1;i++)
        {
            for(j=i;j<MAX;j++)
                if(i!=j)

                if(dist(avion[i],avion[j])<distmin(avion[i],avion[j]))
                {
                    avion[i].warnings++;
                    avion[j].warnings++;
                    totwarnings++;
                }
        }
        for(i=0;i<MAX;i++)
            actualiza_avion(&avion[i]);

        if(n%100==0)
            printf("Total Warnings = %d\n",totwarnings);
    }
    t_final=clock();
    printf("En %3.6f segundos\n",((float) t_final- (float)t_inicial)/
CLOCKS_PER_SEC);
}
```

Ejemplo 1. Versión serial programa que busca aviones a una distancia menor que la mínima en un espacio de 3 dimensiones.

2 *Desarrollo de una versión paralela del cálculo de la distancia entre aviones*

Utilizando la API de Windows para crear hilos, desarrolla una versión paralela que mejore el rendimiento de esta aplicación en un procesador Multicore. Considera que pueden existir condiciones de concurso por lo que es necesario que utilices Parallel Inspector para asegurarte que la versión final no tiene errores por condiciones de concurso.

V EVALUACIÓN

1 *Equipos*

Esta práctica se hará en equipos (máximo 2 integrantes), es necesario que en la revisión esté el equipo completo ya que el integrante que no se presente no tendrá calificación en la práctica.

2 *Entrega*

El programa que se piden en las 2, se entregarán **los archivos fuentes** y el documento PDF que contiene el análisis con Parallel Inspector **empaquetados en un archivo .ZIP** a través del apartado correspondiente en Moodle hasta el día Jueves 26 de Febrero a las 11:59 PM.



No incluya líneas de código en sus programas de las cuales desconozca su funcionamiento. El código no conocido será anulado en el funcionamiento de la práctica.

3 Evaluación

Puntualidad en las revisiones	El equipo estuvo completo y puntual en todas las sesiones de revisión.	Si hubo dos o más sesiones con el equipo, el equipo estuvo completo y puntual en casi todas las sesiones de revisión	Si solo hubo una sesión de revisión, el equipo no estuvo completo o no fue puntual. Si fueron dos o más sesiones de revisión, en más de una sesión el equipo no estuvo completo o fue puntual
	+10	+5	0
Funcionamiento	El producto cumple con todas las especificaciones indicadas en el documento y no tiene fallas	El producto muestra una falla no esperada o el producto está casi completo, puede funcionar excepto la parte no completada	El producto muestra más de una falla inesperada o no funciona, esto puede ser debido a que no esté completo.
	+75	+37.5	0
Interfaz con el usuario	El producto funciona y pudo ser utilizado sin necesidad de recibir indicaciones por el desarrollador, tiene instrucciones claras para ser utilizado.	El producto funciona, pero hubo necesidad de recibir alguna indicación para su uso por parte del desarrollador del producto	El producto carece de instrucciones claras para ser utilizado y requiere que alguno de los desarrolladores esté presente para su utilización o no puede utilizarse debido a que no está completo
	+10	+5	0
Claridad en el código	El código es claro, usa nombres de variables adecuadas, está debidamente comentado e indentado. Puede ser entendido por cualquier otra persona que no intervino en su desarrollo.	El código carece de claridad, puede ser entendido por cualquier persona ajena a su desarrollo pero con cierta dificultad.	El código carece de comentarios, está mal indentado, usa nombres de variables no adecuadas.
	+5	+2.5	0
Defensa del producto	Todos los integrantes son capaces de explicar cualquier parte del producto presentado	Alguno de los integrantes muestra dudas sobre alguna parte del desarrollo del producto presentado	Más de un integrante, o si el trabajo fue individual, el desarrollador duda sobre cómo está desarrollado producto.
	x 1 (puntos se multiplican por 1)	x 0.5 (puntos se multiplican por 0.5)	x 0 (puntos se multiplican por 0)
Sobresaliente 20 %	Tiene 1 en todos los puntos anteriores. El producto entregado es sobresaliente, muestra tener la calidad para ser expuesto como un producto representativo de la carrera. Hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado		No tiene 1 en todos los puntos anteriores, o el producto entregado no es sobresaliente y no muestra tener la calidad para ser expuesto como un producto representativo de la carrera o no hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado
	+20		0