



Nombre de la práctica	Ciclos de reloj			No.	6
Asignatura:	Metodos numericos	Carrera:	Ingeneria en sistemas computacionales	Duración de la práctica (Hrs)	

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): otro

III. Material empleado: equipo de computo con visual studio code

IV. Desarrollo de la práctica: realizar el código que reduzca los tiempos al momento de abrir el programa "libre office" utilizando apuntadores y sin apuntadores para hacer notar la diferencia de tiempo que existe al momento de la ejecución de cada uno.

Ciclos de reloj, por segundo o frecuencia, es el término que hace referencia a la velocidad del procesador incorporado en la CPU del ordenador, y se mide en Megahercios o Gigahercios, el número de operaciones que puede realizar por segundo.

La unidad básica, el **Hercio** (Hertz), mide el número de instrucciones que el ordenador puede ejecutar por segundo desde que comienza un ciclo hasta que vuelve a comenzar. Un Hercio es una instrucción por segundo, dos Hercios, dos por segundo. Un megahercio mil operaciones por segundo y un gigahercio mil millones.

Debemos primero importar las bibliotecas que serán necesarias para poder trabajar y de las cuales se obtendrán las funciones específicas.

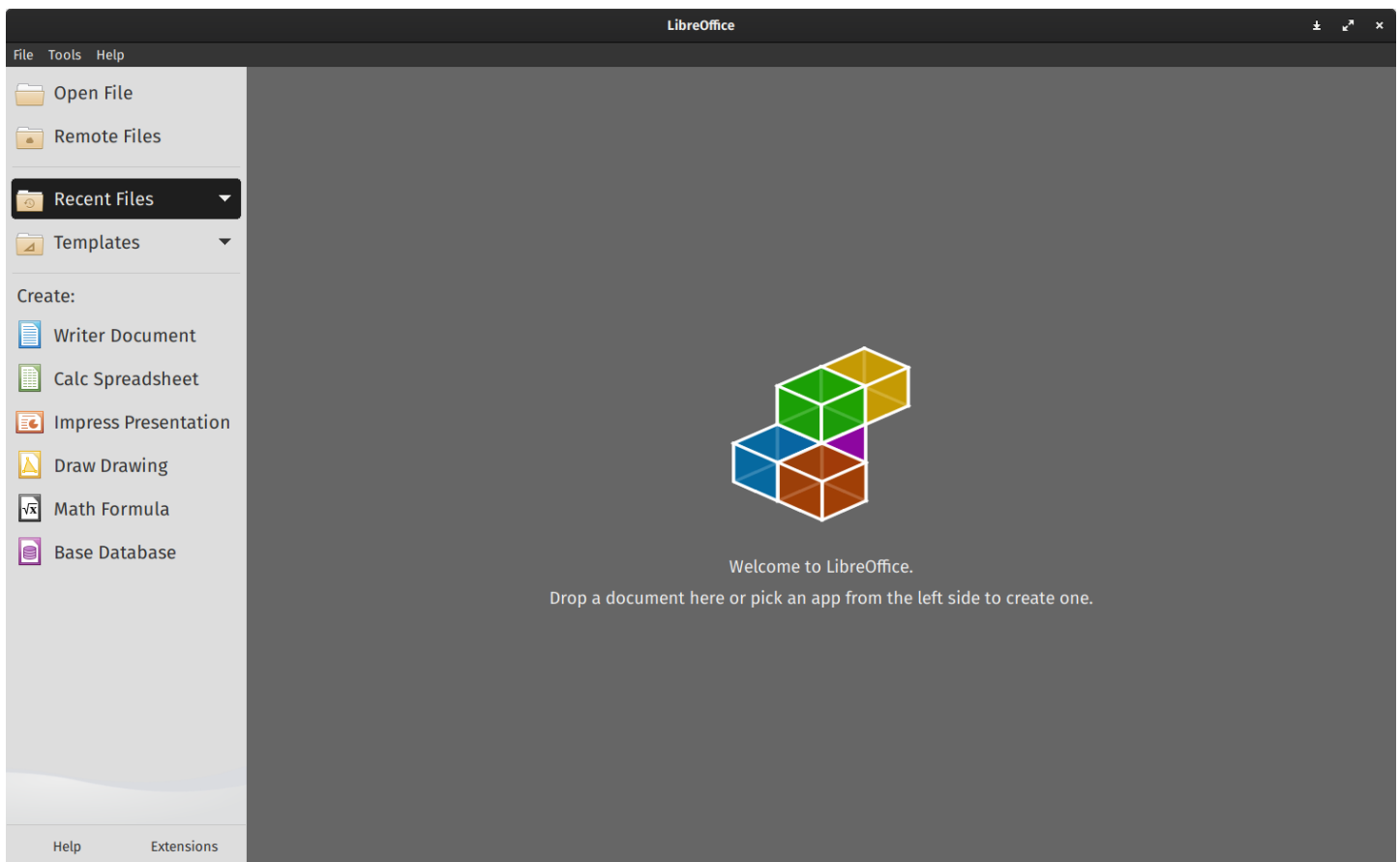
```
1  #include <stdlib.h>
2  #include <stdio.h>
3  #include <time.h>
4  #include <sys/time.h>
5  #include <string.h>
6
7  double timeval_diff(struct timeval *inicio, struct timeval *fin ){
8      return
9      (double)(inicio->tv_sec+(double)inicio->tv_usec/1000000)-
10     (double)(fin->tv_sec+(double)fin->tv_usec/1000000);
11 };
12
13 int main (int argc, char*argv[])
14 {
15     struct timeval t_int, t_fin;
16     double secon;
17     gettimeofday(&t_int,NULL);
18     system("libreoffice");
19     gettimeofday(&t_fin,NULL);
20     //manda a llamar la función
21     secon=timeval_diff(&t_fin,&t_int);
22     printf("%.16g milisegundos\n",secon*1000.0);
23     return 0;
24 }
```

9176.127910614014 milisegundos

Para poder medir el tiempo de ejecución de cierta operación como lo es en este caso que pretende abrir el programa de libre office vamos implementar la función "gettimeofday" que se encuentra en la biblioteca "sys/time.h". y su sintaxis se representa: gettimeofday (struct timeval \* tp,NULL); timeval es un registro con dos campos Int tv\_sec , int tv\_usec que muestran los milisegundos y microsegundos.

Dentro del método principal se encuentra la función del programa aquí tenemos dos argumentos uno de tipo entero llamado: argc y el otro un apuntador de tipo char. También están declarados dos variables: t\_ini y t\_fin que pertenecen a la estructura timeval una representa el tiempo de inicio y la otra el tiempo final. Con ayuda de la función gettimeofday vamos obtener estos tiempos de inicio y fin Y también se declaro a la variable secon.

System () es una función que sirve para ejecutar subprocessos o comandos del sistema operativo, es por eso que dentro de esta indicamos el programa que deseamos ejecutar "libre office" y se encuentra incluida en la biblioteca <stdlib.h>. Después con la función timeval\_ diff a la que le guardaremos los valores de parámetros de tiempo de inicio y final para obtener los segundos que tardo en abrir el programa si también guardaremos el valor que retorne esta función en la variable secon lo multiplicamos por 1000 para obtener el valor en microsegundos.



```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4 #include <sys/time.h>
5 #include <string.h>
6
7 double timeval_diff(struct timeval *a, struct timeval *b ){
8     return
9
10    (double)(a->tv_sec+(double)a->tv_usec/1000000)-
11    (double)(b->tv_sec+(double)b->tv_usec/1000000);
12 };
13
14 int main (int argc, char*argv[]){
15
16     char * nombre="libreoffice";
17     struct timeval t_int, t_fin;
18     double segundos;
19     gettimeofday(&t_int,NULL);
20     system(nombre);
21     system("'" + nombre + "'");
22     gettimeofday(&t_fin,NULL);
23     segundos=timeval_diff(&t_fin,&t_int);
24     printf("%.16g milisegundos\n",segundos*1000.0);
25     return 0;
26 }
27
```

4924.782037734985 milisegundos

Tiempo de  
ejecución con  
apuntadores

Mientras que para ejecutar nuestro programa empleando apuntadores y que abra libre office vamos hacer uso del mismo código pues recordemos que lo que se esta realizando es la comparación de tiempos con y sin apuntadores. Entonces lo para nuestros códigos utilizando apuntadores lo que vamos agregarle es el puntero que hace llamar al programa, así que vamos a declara el apuntador de tipo char y después dentro de la función System hacemos llamar a ese apuntador que se hará cargo de abrir libre office. El resto de las funciona operan de la misma manera que nuestro código anterior.



**V. Conclusiones:** se realizó esta práctica con un objetivo principal y este objetivo es dar cuenta de cómo opera o funciona nuestro código con y sin la aplicación de apuntadores pero principalmente como se tiene mencionando el tiempo significativo que se pueden obtener cuando se o no implantan como lo vimos en estos dos códigos con una demostración que puede parecer sencilla pero significativa no solo en la demostración en diferencia de tiempos sino también en la adquisición de aprendizaje.