



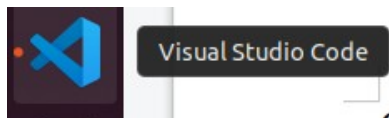
Nombre de la práctica	Ciclos de Reloj			No.	6
Asignatura:	Métodos numéricos	Carrera:	Ing. Sistemas Computacionales	Duración de la práctica (Hrs)	

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

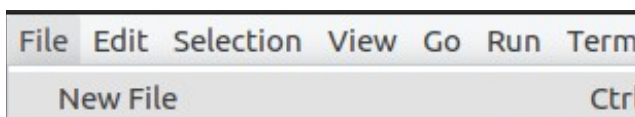
III. Material empleado: laptop con linux y Visual Studio Code.

IV. Desarrollo de la práctica:

Comenzamos abriendo Visual Studio Code, dando clic sobre el icono que se muestra a continuación



Posteriormente creo un nuevo proyecto dando clic en la pestaña inferior en File>New File



A continuación di clic nuevamente en la parte superior en save y guarde mi proyecto con el siguiente nombre y la extensión c y sucesivamente dar clic en el botón save



Comenzamos incluyendo bibliotecas en el área de trabajo, las cuales proveen los elementos necesarios para las futuras instrucciones, en el caso de `<stdlib.h>`, servirá para controlar procesos, gestión de memoria dinámica y conversión de tipos de datos, en cuanto a `<stdio.h>` se utilizara para manipular datos de entrada y salida , mientras que `<string.h>` nos ayudara en la clasificación de caracteres, después tenemos a la biblioteca `<time.h>` que además de contar con prototipos como lo son las funciones, nos ayuda a manipular la fecha y hora del sistema y finalmente la biblioteca `<sys/time.h>` que nos permite manipular los valores en cuanto al tiempo e imprimirlos

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <string.h>
4 #include <time.h>
5 #include <sys/time.h>
```

Posteriormente declara las siguientes instrucciones, que permiten pasar como parámetros las estructuras llamadas start y end de los cuales se obtiene la diferencia una vez asignados sus valores según el tiempo que demore el realizar cierto proceso que en este caso es abrir el programa libre office, para lo cual se permite una aproximación por medio de tv_sec para segundos y tv_usec de micro segundos y para obtener el valor debe ser convertido a tipo de dato double, permitiendo mayor precisión gracias a los decimales.

```
double timevalDifference(struct timeval *start, struct timeval *end)
{
    return (double)(start->tv_sec + (double)start->tv_usec / 1000000) -
           (double)(end->tv_sec + (double)end->tv_usec / 1000000);
};
```

Sin embargo en las instrucciones anteriores aun no se tiene un valor asignado puesto que aun no se lleva ningún proceso, por ello por medio de la presente función de arranque se tienen 2 argumentos de tipo entero en los cuales uno es de tipo entero y el otro char utilizado para el apuntador, después se declaran las estructuras que son de tipo timeval para el tiempo de inicio y fin respectivamente y de igual manera se declara una variable nombrada secon de tipo double, la cual nos ayudara a almacenar el valor respectivo a los segundos de diferencia, para ello por medio de la sintaxis gettimeofday (&timeStart, NULL); hace referencia a el tiempo de inicio del proceso y posteriormente en la siguiente instrucción se abre el programa, y se toma posteriormente el tiempo final, mientras que en la siguiente instrucción se pasan los valores correspondientes a la funcion declarada al principio, en la cual se obtiene el tiempo en segundos y estos se asignan a la variable secon,

y finalmente secon(segundos obtenidos) se multiplican por 1000.0 para obtener más precisión y así el resultado a imprimir sean los mili segundos.

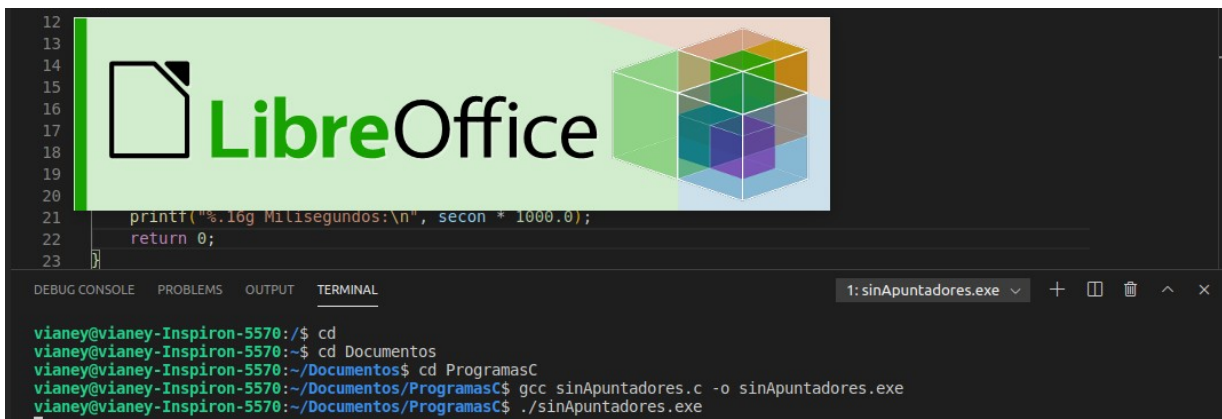
```
int main(int argc, char *argv[])
{
    struct timeval timeStart, timeEnd;
    double secon;
    gettimeofday(&timeStart, NULL);
    system("libreoffice");
    gettimeofday(&timeEnd, NULL);
    secon = timevalDifference(&timeEnd, &timeStart);
    printf("%.16g Milisegundos:\n", secon * 1000.0);
    return 0;
}
```

Posteriormente se ejecuta desde la terminal, con ayuda de la barra de herramientas en la parte superior dando clic derecho sobre Terminar y selecciona en la barra desplegable New terminal.

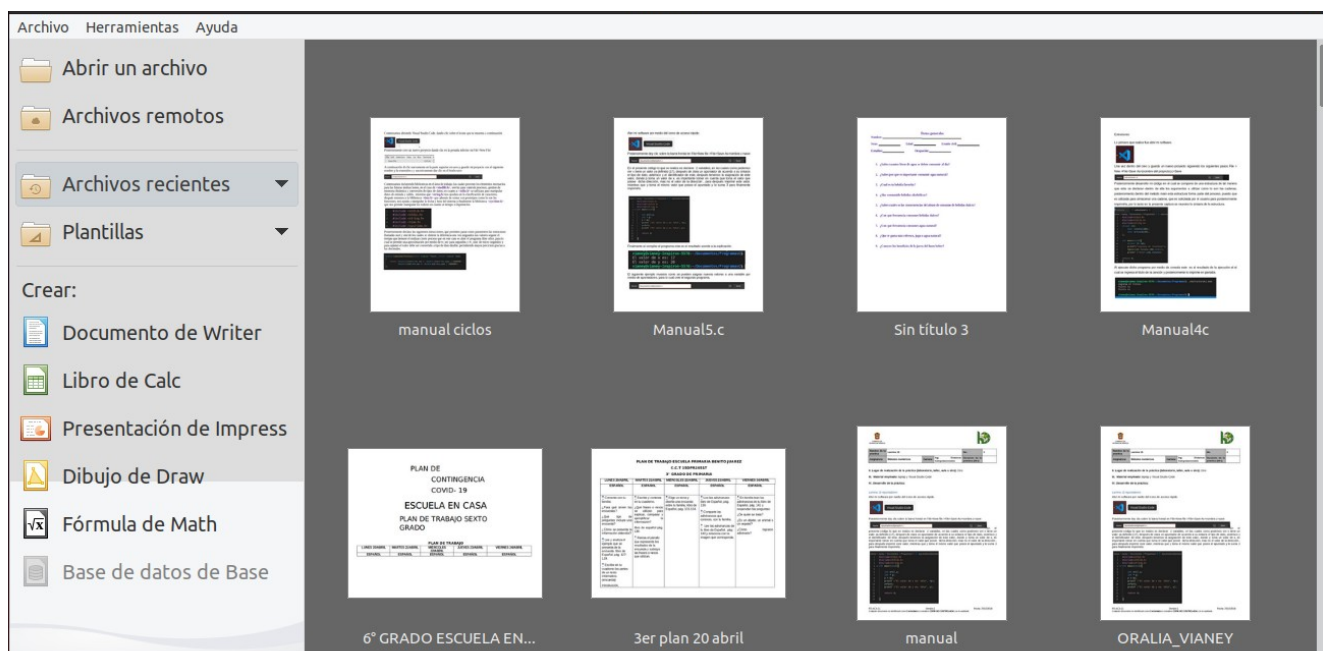
En la cual debes entrar en la carpeta donde se alojan tus programas y ejecutar el programa de la siguiente manera

```
vianey@vianey-Inspiron-5570:/$ cd
vianey@vianey-Inspiron-5570:~$ cd Documentos
vianey@vianey-Inspiron-5570:~/Documentos$ cd ProgramasC
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$ gcc sinApuntadores.c -o sinApuntadores.exe
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$ ./sinApuntadores.exe
```

Al dar enter el programa se comienza a ejecutar abriendo libre Office



Una vez que se abre aparece la siguiente pantalla para seleccionar el tipo de planilla sin embargo esta se debe cerrar inmediatamente.



Volviendo así a nuestra terminal donde se muestra el tiempo que tardo en ejecutarse la operación en milisegundos

```
43576.93719863892 milisegundos
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$
```

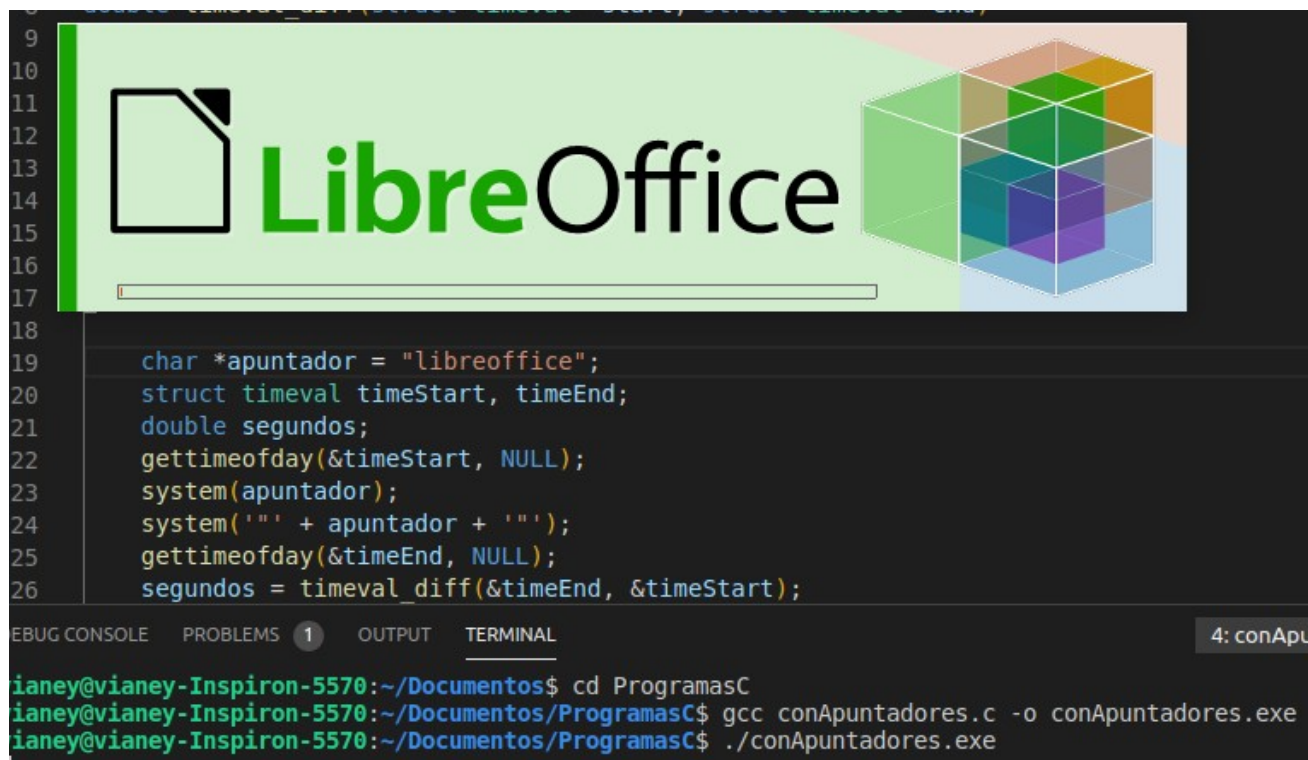
Mientras que al realizarlo con apuntadores lo que se realiza es optimizar este tiempo en el que se ejecuta dicha operación, para ello reutilizamos el mismo código pero a diferencia del anterior agregamos un puntero de tipo char el cual apuntara a la dirección de la ruta para abrir el libre Office y este esta anexado dentro de la función de arranque de la siguiente manera.

```
ApuntadoresEjemplo4.c • conApuntadores.c • Apunt.c • sinApuntadores.c •
home > vianey > Documentos > ProgramasC > conApuntadores.c > main(int, char *[])
1
2  #include <stdlib.h>
3  #include <stdio.h>
4  #include <time.h>
5  #include <sys/time.h>
6  #include <string.h>
7
8  double timeval_diff(struct timeval *start, struct timeval *end)
9  {
10     return
11         (double)(start->tv_sec + (double)start->tv_usec / 1000000) -
12         (double)(end->tv_sec + (double)end->tv_usec / 1000000);
13 };
14
15
16 int main(int argc, char *argv[])
17 {
18     char *apuntador = "libreoffice";
19     struct timeval timeStart, timeEnd;
20     double segundos;
21     gettimeofday(&timeStart, NULL);
22     system(apuntador);
23     system("'" + apuntador + "'");
24     gettimeofday(&timeEnd, NULL);
25     segundos = timeval_diff(&timeEnd, &timeStart);
26     printf("%.16g Milisegundos\n", segundos * 1000.0);
27     return 0;
28 }
29
```

Declaración del apuntador y asignación de la dirección a la cual apunta

Al mandar a llamar al apuntador por medio del system se abre el programa libre Office

Se ejecuta el programa desde la terminal como el anterior

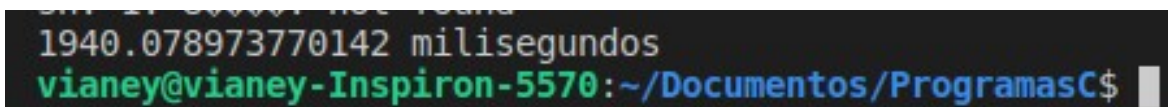


```
9
10
11
12
13
14
15
16
17
18
19 char *apuntador = "libreoffice";
20 struct timeval timeStart, timeEnd;
21 double segundos;
22 gettimeofday(&timeStart, NULL);
23 system(apuntador);
24 system('' + apuntador + '');
25 gettimeofday(&timeEnd, NULL);
26 segundos = timeval_diff(&timeEnd, &timeStart);
```

EBUG CONSOLE PROBLEMS 1 OUTPUT TERMINAL 4: conApu

```
vianey@vianey-Inspiron-5570:~/Documentos$ cd ProgramasC
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$ gcc conApuntadores.c -o conApuntadores.exe
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$ ./conApuntadores.exe
```

Posteriormente cerramos las planillas y en la terminal el tiempo que se tardó en realizar el proceso reduciendo así el tiempo.



```
1940.078973770142 milisegundos
vianey@vianey-Inspiron-5570:~/Documentos/ProgramasC$
```

V. Conclusiones:

Para concluir con este proyecto tenemos entendido que los ciclos de reloj es la velocidad en la que el procesador de la CPU realiza determinado proceso, por lo tanto en este caso medimos el tiempo que tarda el proceso para abrir Libre Office y para optimizar este tiempo y dar mayor velocidad al realizar dicha operación, lo que realizamos es abrir directamente el programa por medio de su dirección por medio de punteros los cuales apuntan a dicha dirección.