Kỹ Thuật Lập Trình

(Ngôn Ngữ Lập Trình C)

Bài giảng số 3

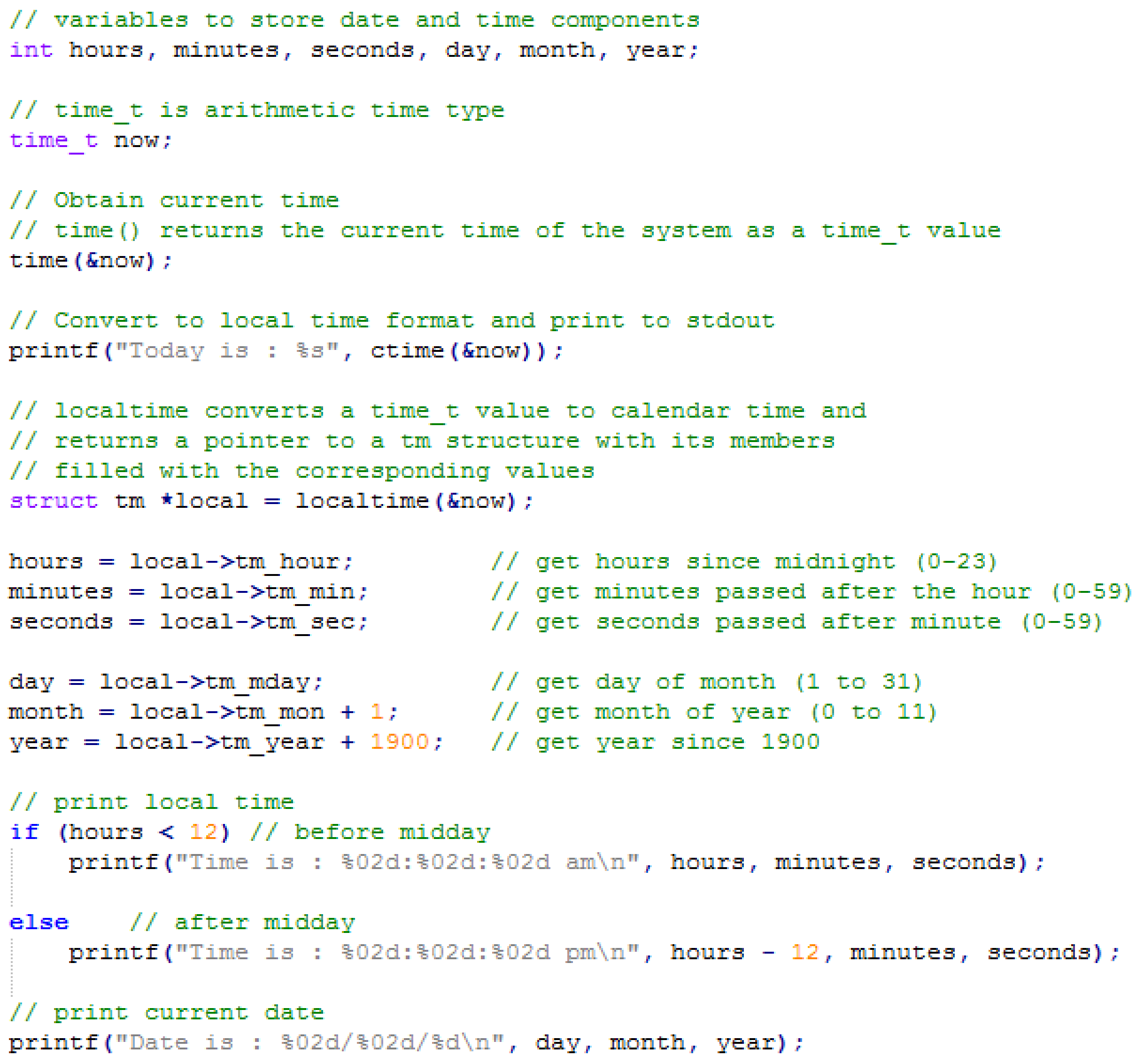
Xử lý Date/Time trong ngôn ngữ C

Tại sao xử lý time/date

- Ví dụ về dữ liệu GPS
- $GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47
  - 123519   Fix taken at 12:35:19 UTC
- Log file data

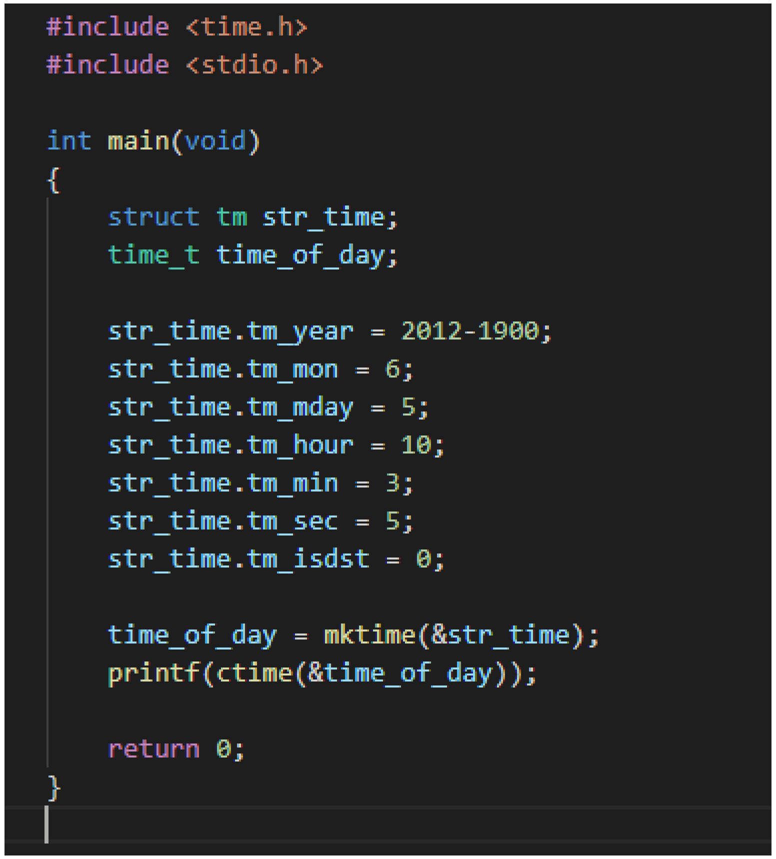
```
C:\Program Files (x86)\EaseUS\Todo Backup\Agent.exe
2017-07-10 17:35:16 [M:00,T/P:1940/6300] Init Log
2017-07-10 17:35:16 [M:29,T/P:1940/6300] Ldq : Agent start install!
2017-07-10 17:35:16 [M:29,T/P:1940/6300] Ldq : Agent call CreateService!
2017-07-10 17:35:16 [M:29,T/P:1940/6300] Ldq : Agent call CreateService is success!
```

Các hàm cơ bản trong C với Time



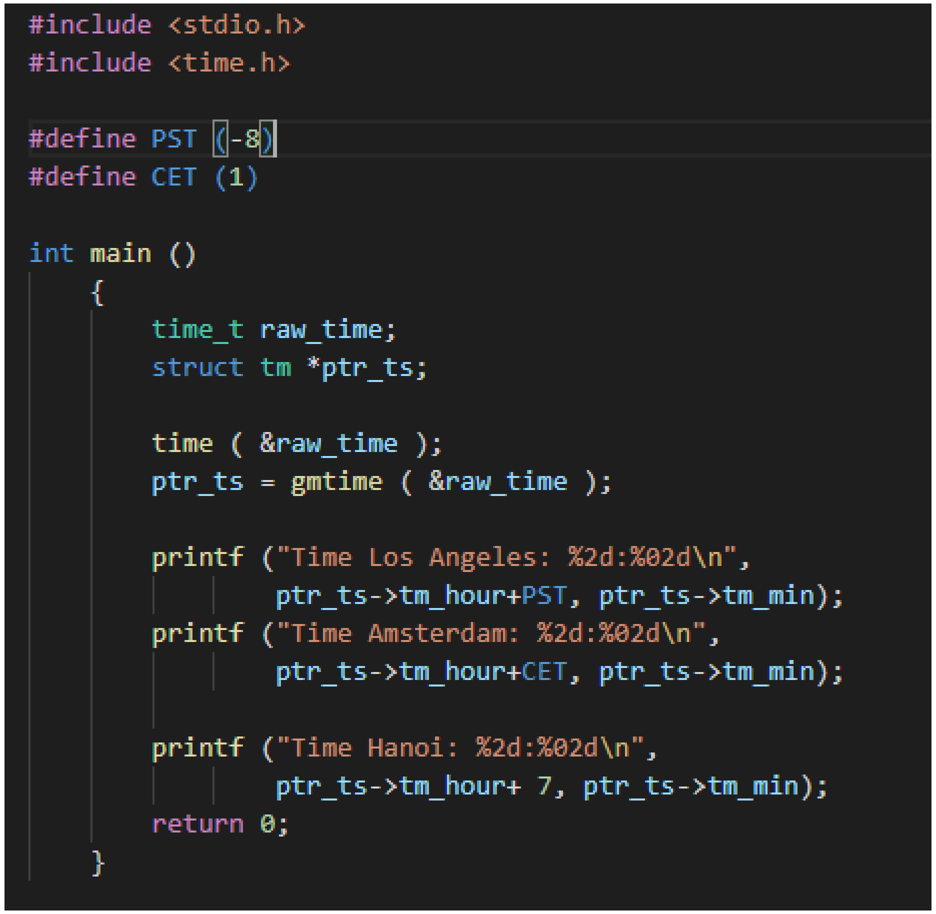
```c
// variables to store date and time components
int hours, minutes, seconds, day, month, year;

// time_t is arithmetic time type
time_t now;

// Obtain current time
// time() returns the current time of the system as a time_t value
time(&now);

// Convert to local time format and print to stdout
printf("Today is : %s", ctime(&now));

// localtime converts a time_t value to calendar time and
// returns a pointer to a tm structure with its members
// filled with the corresponding values
struct tm *local = localtime(&now);

hours = local->tm_hour;         // get hours since midnight (0-23)
minutes = local->tm_min;        // get minutes passed after the hour (0-59)
seconds = local->tm_sec;        // get seconds passed after minute (0-59)

day = local->tm_mday;           // get day of month (1 to 31)
month = local->tm_mon + 1;      // get month of year (0 to 11)
year = local->tm_year + 1900;   // get year since 1900

// print local time
if (hours < 12) // before midday
    printf("Time is : %02d:%02d:%02d am\n", hours, minutes, seconds);

else    // after midday
    printf("Time is : %02d:%02d:%02d pm\n", hours - 12, minutes, seconds);

// print current date
printf("Date is : %02d/%02d/%d\n", day, month, year);
```

Sự dụng hàm make time



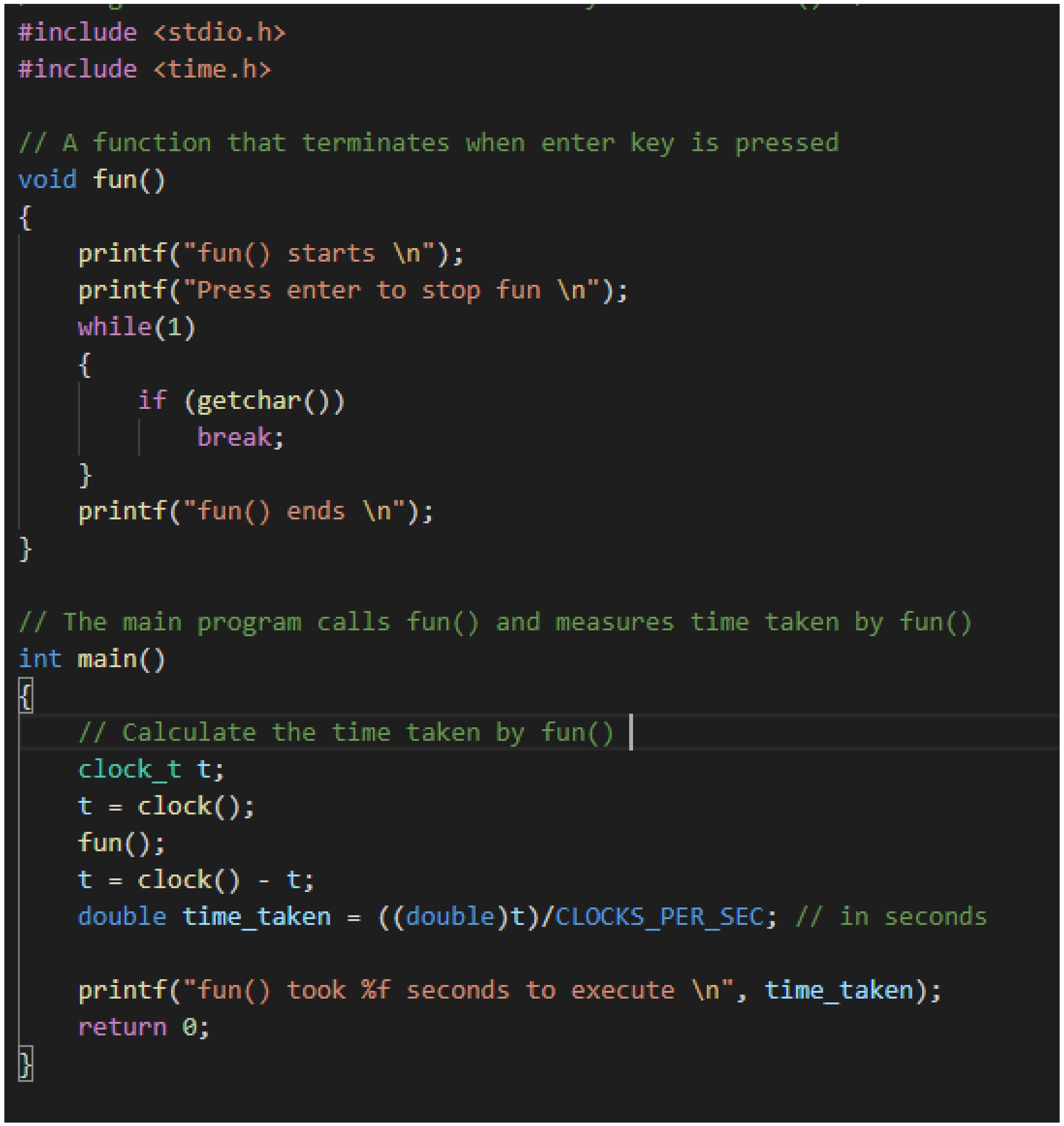
```c
#include <time.h>
#include <stdio.h>

int main(void)
{
    struct tm str_time;
    time_t time_of_day;

    str_time.tm_year = 2012-1900;
    str_time.tm_mon = 6;
    str_time.tm_mday = 5;
    str_time.tm_hour = 10;
    str_time.tm_min = 3;
    str_time.tm_sec = 5;
    str_time.tm_isdst = 0;

    time_of_day = mktime(&str_time);
    printf(ctime(&time_of_day));

    return 0;
}
```

Time Zone



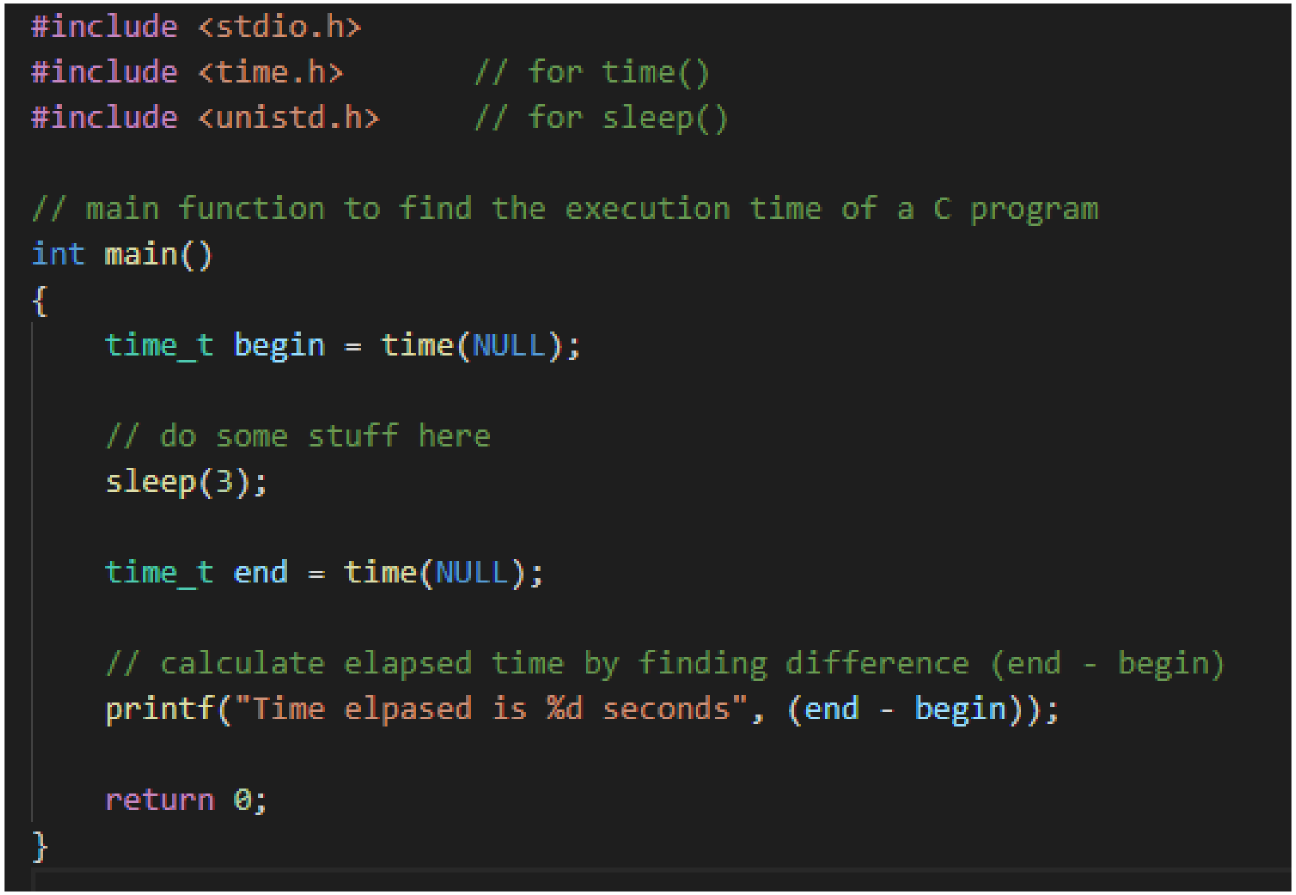
```c
#include <stdio.h>
#include <time.h>

#define PST (-8)
#define CET (1)

int main ()
    {
        time_t raw_time;
        struct tm *ptr_ts;

        time ( &raw_time );
        ptr_ts = gmtime ( &raw_time );

        printf ("Time Los Angeles: %2d:%02d\n",
                ptr_ts->tm_hour+PST, ptr_ts->tm_min);
        printf ("Time Amsterdam: %2d:%02d\n",
                ptr_ts->tm_hour+CET, ptr_ts->tm_min);

        printf ("Time Hanoi: %2d:%02d\n",
                ptr_ts->tm_hour+ 7, ptr_ts->tm_min);
        return 0;
    }
```

Measure time taken in C?



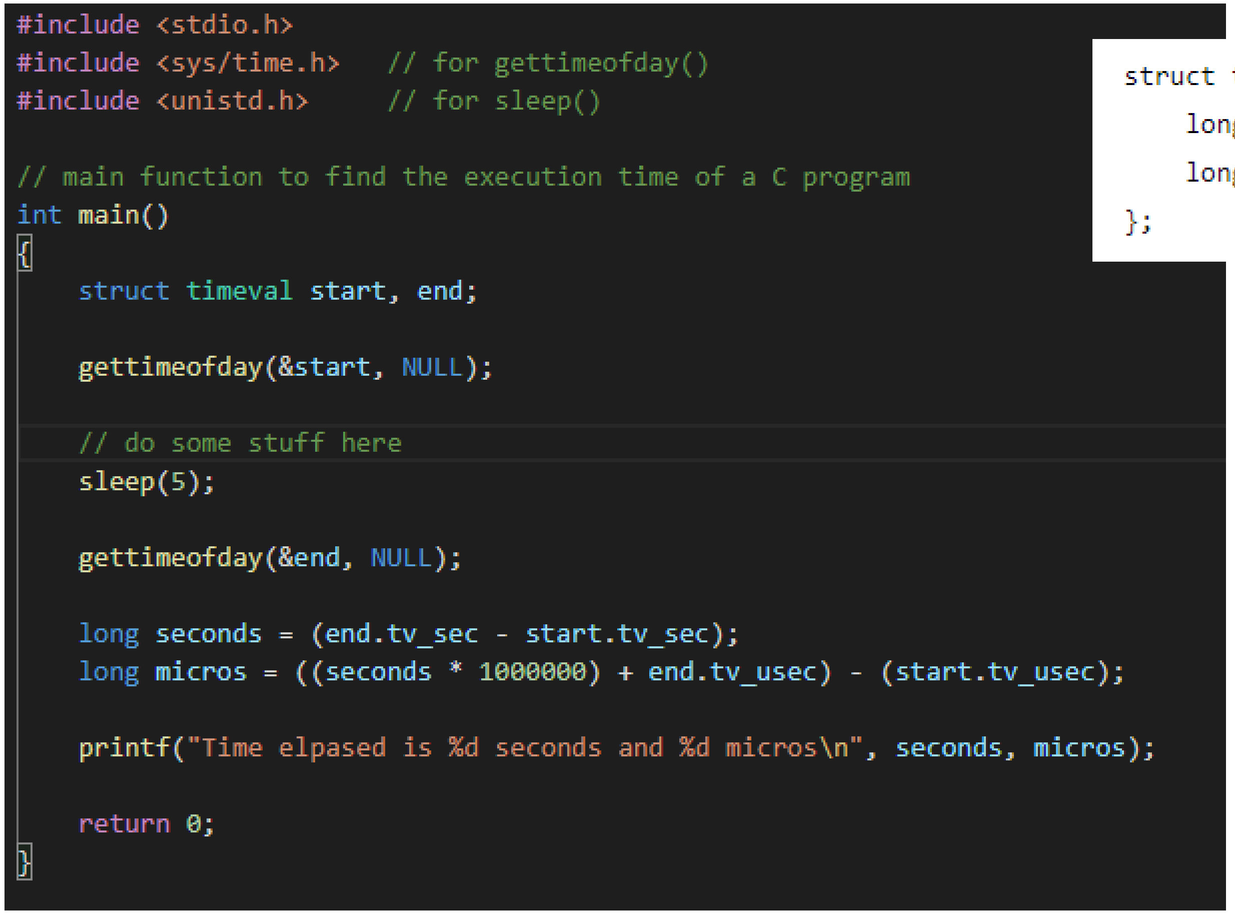
```
#include <stdio.h>
#include <time.h>

// A function that terminates when enter key is pressed
void fun()
{
    printf("fun() starts \n");
    printf("Press enter to stop fun \n");
    while(1)
    {
        if (getchar())
            break;
    }
    printf("fun() ends \n");
}

// The main program calls fun() and measures time taken by fun()
int main()
{
    // Calculate the time taken by fun()
    clock_t t;
    t = clock();
    fun();
    t = clock() - t;
    double time_taken = ((double)t)/CLOCKS_PER_SEC; // in seconds

    printf("fun() took %f seconds to execute \n", time_taken);
    return 0;
}
```

Phương pháp số 2

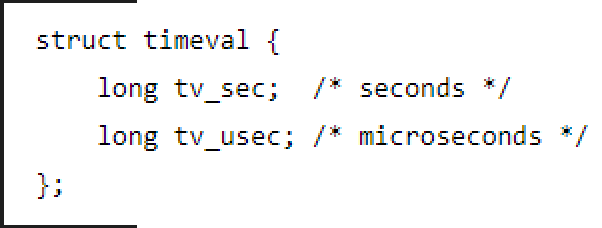


```c
#include <stdio.h>
#include <time.h>       // for time()
#include <unistd.h>     // for sleep()

// main function to find the execution time of a C program
int main()
{
    time_t begin = time(NULL);

    // do some stuff here
    sleep(3);

    time_t end = time(NULL);

    // calculate elapsed time by finding difference (end - begin)
    printf("Time elpased is %d seconds", (end - begin));

    return 0;
}
```

Phương pháp số 3



```c
#include <stdio.h>
#include <sys/time.h>   // for gettimeofday()
#include <unistd.h>     // for sleep()

// main function to find the execution time of a C program
int main()
{
    struct timeval start, end;

    gettimeofday(&start, NULL);

    // do some stuff here
    sleep(5);

    gettimeofday(&end, NULL);

    long seconds = (end.tv_sec - start.tv_sec);
    long micros = ((seconds * 1000000) + end.tv_usec) - (start.tv_usec);

    printf("Time elpased is %d seconds and %d micros\n", seconds, micros);

    return 0;
}
```



```c
struct timeval {
    long tv_sec;  /* seconds */
    long tv_usec; /* microseconds */
};
```