Lesson 83 Thread

The computer does more than a billion operations per second, how do these operations take place at one time?

As a programmer, we write the program step by step, for example:

We ask the user to enter a number, then the user enters the number and takes some seconds to type it, but in these seconds what does the computer do?

The computer does many things, as long as the computer is working, there are more than one program running at the same moment, such as the mouse program, google Chrome, Internet Explorer, or a video in the background, all of these tasks are working in the processes.

As for our program, if it stops waiting for the user to enter the number, at this time there are other programs running, and therefore the computer can do one billion operations per second or less.

Here is the question: Can our program be made to do more than one operation at a time?

Yes and we call this Thread.

In C programming, we usually write the tasks line by line but there are some tasks that work at the same time.

But what is the advantage of that?

Shortening the time, instead of making a part of the program and taking time, needing to do another function at the same time.

for example :

While writing texts in word, the program doesn’t wait for the user to end writing to check the grammar or orthographic errors. No, at the same time while writing the program checks the errors. And that’s called thread.

Thus, **thread** will make our program run faster **because it executes many commands at one time.**

But how do we create a **thread**?

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
void* fun1(void* v) {
//to make function fun works with thread we have to add pointer
//pass variables from main to function
printf("Hello Gammal Tech\n");
return NULL;
//return is necessary here
//we will return null
}
void* fun2(void* v) {
printf("Hi Gammal Tech\n");
return NULL;
}
```

```
int main() {
pthread_t th
//define thread
//thread in called th
pthread_create(&th, NULL, fun1, NULL);
//&th points to the function
//null because there is no attributs
//call function fun 1
pthread_create(&th, NULL, fun2, NULL);
pthread_join(th, NULL);
//return 0 stops the main
//exist(0) stops the program
exist(0);
}
```

Try the code : Click Here !

Another Example:

```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
//to add exist(0)
#include <unistd.h>
```

```
//to add sleep()
void* fun1(void* v) {
int i,*n=(int *)v;
for (i = 0; i < *n; i++) {
printf("Hello Gammal Tech\n");
sleep(1);
//the program stops for 1 second after printing
}
return NULL;
}
void* fun2(void* v) {
int i,*n=(int*)v;
for (i = 0; i < *n; i++) {
printf("Hi Gammal Tech\n");
sleep(1);
}
return NULL;
}
int main() {
int n = 5;
pthread_t th;
pthread_create(&th, NULL, fun1, &n);
pthread_create(&th, NULL, fun2, &n);
pthread_join(th, NULL);
exit(0);
}
```

Try the code : Click Here !