

تمرین 5.

به عنوان مقدمه فرض کنید یک پروژه آردوینو دارید که می‌خواهید چندین عمل را همزمان انجام دهید: خواندن داده‌ها از ورودی کاربر، چشمک زدن برخی LEDها، نظارت بر پتانسیومتر و غیره. بنابراین، اساساً می‌خواهید چند کار را با آردوینو انجام دهید. ما چندین روش برای به کار گیری نخ‌ها در برنامه‌های آردوینو داریم که مهمترین آن‌ها عبارتند از : **Multithreading** و **Protothreading** است.

چند نخ‌ی یا **Multithreading** : چند نخ‌ی شکل ویژه‌ای از چند وظیفه‌ای (multitasking) است. چند وظیفه‌ای خاصیتی است که به کامپیوتر این امکان را می‌دهد تا دو یا چند برنامه را به طور همزمان اجرا کند. در حالت کلی، دو نوع چند وظیفه‌ای وجود دارد: مبتنی بر پروسس (process-based) و مبتنی بر نخ (thread-based). چند وظیفه‌ای مبتنی بر پروسس، اجرای همزمان برنامه‌ها را برعهده می‌گیرد. چند وظیفه‌ای مبتنی بر نخ اجرای همزمان چند تکه از یک برنامه را برعهده می‌گیرد. یک برنامه شامل دو یا چند بخش است که می‌توان همزمان آنها را اجرا کرد. هر بخش از برنامه یک نخ (thread) نامیده می‌شود، و هر نخ یک مسیر اجرایی جداگانه تعریف می‌کند.

یک مثال عملی برای ساخت نخ‌ها در لینوکس برای درک بهتر مفهوم نخ‌ها اینجا شرح می‌دهیم که در ادامه خواهیم دید.

ساخت نخ‌ها :

از روتین زیر برای ساخت یک نخ POSIX استفاده می‌شود.

```
#include <pthread.h>
pthread_create (thread, attr, start_routine, arg)
```

در اینجا، «pthread_create» یک نخ جدید ساخته و آن را قابل اجرا می‌سازد. این روتین را می‌توان هر چند بار که نیاز باشد در برنامه فراخوانی کنید. توضیح این پارامترها در زیر آمده است.

ردیف	پارامتر و توضیح آن
1	<p><code>thread</code></p> <p>یک شناسه یکتا برای نخ جدیدی که روتین برمی گرداند.</p>
2	<p><code>attr</code></p> <p>یک شی <code>attribute</code> ناشفاف که برای تنظیم خواص نخ استفاده می شود. می توان یک شی <code>attribute</code> برای نخ مشخص کرد، یا برای حالت تنظیمات پیش فرض <code>NULL</code> را قرار داد.</p>
3	<p><code>start_routine</code></p> <p>روتین <code>C++</code> که نخ پس از ساخته شدن باید آن را اجرا کند.</p>
4	<p><code>arg</code></p> <p>یک آرگومان منفرد که می توان به <code>start_routine</code> ارسال کرد. این آرگومان باید به صورت یک رفرنس به یک اشاره گر <code>void</code> ارسال شود. اگر نخواهیم این آرگومان را ارسال کنیم، مقدار <code>NULL</code> وارد می کنیم.</p>

بیشترین تعداد نخ‌ای که یک پروسس می‌تواند ایجاد کند وابسته به پیاده‌سازی می‌باشد. زمانی‌که نخ‌ها ساخته شدند، با هم جفت شده و می‌توانند نخ‌های جدیدی بسازند. هیچ سلسله مراتب یا وابستگی بین نخ‌ها وجود ندارد.

خاتمه دادن به یک نخ

روتین زیر برای خاتمه دادن به یک نخ POSIX استفاده می‌شود.

```
#include <pthread.h>
pthread_exit (status)
```

در اینجا «`pthread_exit`» برای خروج از یک نخ به صورت صریح به کار می‌رود. عموماً روتین «`pthread_exit`» را پس از اینکه یک نخ کارش را به طور کامل انجام داد و دیگر به وجودش نیازی نبود، فراخوانی می‌کنند.

اگر تابع `main()` قبل از نخ‌هایی که خودش ساخته به اتمام برسد و با روتین `pthread_exit()` خارج شود، دیگر نخ‌ها به اجرای خود ادامه خواهند داد. در غیر این صورت، آنها نیز با تمام شدن `main()` خاتمه می‌یابند.

مثال

این مثال ساده با استفاده از روتین «`pthread_create`» پنج نخ ایجاد می‌کند. هر نخ یک پیام «Hello World» چاپ کرده و با فراخوانی `pthread_exit()` خاتمه می‌یابد.

```

#include <iostream>
#include <cstdlib>
#include <pthread.h>

using namespace std;

#define NUM_THREADS 5

void *PrintHello(void *threadid) {
    long tid;
    tid = (long)threadid;
    cout << "Hello World! Thread ID, " << tid << endl;
    pthread_exit(NULL);
}

int main () {
    pthread_t threads[NUM_THREADS];
    int rc;
    int i;

    for( i = 0; i < NUM_THREADS; i++ ) {
        cout << "main() : creating thread, " << i << endl;
        rc = pthread_create(&threads[i], NULL, PrintHello, (void *)i);

        if (rc) {
            cout << "Error:unable to create thread," << rc << endl;
            exit(-1);
        }
    }
    pthread_exit(NULL);
}

```

کد بالا را با کتابخانه **pthread** - به صورت زیر کامپایل کنید.

```
$gcc test.cpp -pthread
```

حال برنامه را اجرا کنید تا خروجی زیر را مشاهده کنید.

```
main() : creating thread, 0
main() : creating thread, 1
main() : creating thread, 2
main() : creating thread, 3
main() : creating thread, 4
Hello World! Thread ID, 0
Hello World! Thread ID, 1
Hello World! Thread ID, 2
Hello World! Thread ID, 3
Hello World! Thread ID, 4
```

این یک مثالی بود برای درک مفهوم نخ ها در عمل که ما بتوانیم به صورت پویا از آن استفاده کنیم.