**תכנות נימים ומקביליות**

אחד השימושים של נימים הינו לנסות ליצור **אפקט של מקביליות** . האפקט של מקביליות במובן מסוים הוא כפול:

1. אם אנחנו, למשל, מחפשים פתרון מתוך מספר רב של אפשרויות, לעיתים חלוקת החיפוש לפי שני קריטריונים מקבילים תביא אותנו לפתרון יותר מהר.
2. במערכות של היום (מרובי ליבות, מרובי מעבדים) ובודאי בעתיד יש/יהיה תמיכה בביצוע חישוב במקביל של חישובים בלתי תלויים. יהיה לנו אפקט מסוים של ביצוע שתי פתרון (או חלוקת עבודה) במקביל.

דוגמא לכך היא הבעיה של **בדיקת ראשוניות של מספר שלם**. דרך סידרתית לבדוק אם n נתון הוא ראשוני לבדוק אם המספרים 2, 3, 5, 7, 9, 11, ... , n1/2 מחלקים עם שארית שונה מאפס.

דרך מקבילית היא לחלק את העבודה בין 2 נימים. הנים הראשי יבדוק אם המספר זוגי ואז יפעיל 2 נימים: הראשון יבדוק את המספרים 3, 7, 11, 15, .... השני 5, 9, 13, 17,.... במידה ואחד הנימים "מגלה" מחלק, הוא עוצר את הנים השני. במידה והמספר n הוא גדול ואינו ראשוני והמחלקים שלו גדולים יחסית, יתכן כי הגישה המקבילית הזו תגיע למחלק יותר מהר, כי ככה נבדקים מספרים גדולים מהר יותר. בכל מקרה אם יש מקביליות בביצוע נימים, יש סיכוי מצוין להגיע לתשובה באופן משמעותי יותר מהר. ראיה משמעותית שיש מקביליות תהיה אם על מספר ראשוני גדול, התוכנית אם הנימים תתכנס מהר יותר.

להלן 2 תוכניות דוגמא שממחישות את הרעיונות הללו.

ראשית, תוכנית רגילה של בדיקת ראשוניות:

/\* shprime.c - test if a given number is a prime \*/

#include <stdio.h>  
#include <math.h>

int test\_prime( unsigned long long int n, unsigned long long \*factor)  
{  
unsigned long long k;  
unsigned long long k1;  
int is\_prime;

if (n == 2)  
 return 1;

if ((n&1) == 0)  
 {  
 \*factor = 2;  
 return 0;  
 } //if

k1= sqrtl((long double) n);  
 if ((k1 & 1) == 0)  
 --k1;

k = 3;  
 is\_prime = 1;

while( (k <= k1) && (is\_prime == 1))  
 if ((n % k) != 0)  
 k += 2;  
 else  
 is\_prime = 0;

if (is\_prime == 0)  
 \*factor = k;

return is\_prime;

} /\* test\_prime1 \*/

int main(void)  
{  
 unsigned long long int n, k;  
 int is\_prime;

printf("Enter unsigned integer:\n");  
 scanf("%Lu", &n);  
 system("date");

is\_prime = test\_prime(n,&k);

system("date");

if (is\_prime == 1)  
 printf("%Lu IS a prime\n", n);  
 else  
 {  
 printf("%Lu is NOT a prime\n", n);

printf("%Lu = %Lu \* %Lu\n", n, k, n/k);

} /\* else \*/

return 0;

} /\* main \*/

דוגמאות הרצה:

% cc -lm sprime.c  
% ./a.out  
Enter unsigned integer:  
18446744073709551577  
Mon Oct 3 20:02:28 IST 2011  
Mon Oct 3 20:02:30 IST 2011  
18446744073709551577 is NOT a prime  
18446744073709551577 = 139646831 \* 132095686967

% ./a.out  
Enter unsigned integer:  
18446744073709551557  
Mon Oct 3 20:05:44 IST 2011  
Mon Oct 3 20:06:40 IST 2011  
18446744073709551557 IS a prime  
%

להלן תוכנית נימים:

/\* shprime.c - test if a given number is a prime \*/

#include <stdio.h>

#include <math.h>

#include <pthread.h>

unsigned long long n;

unsigned long long k;

unsigned long long k1;

unsigned long long k2;

int is\_prime1;

int is\_prime2;

int result1;

int result2;

void \*test\_prime1( void \*ptr )

{

while( (k1 <= k) && (is\_prime1 == 1) && (is\_prime2 == 1))

{

if ((n % k1) != 0)

k1 += 4;

else

is\_prime1 = 0;

} /\* while \*/

return NULL;

} /\* test\_prime1 \*/

void \*test\_prime2( void \*ptr )

{

while( (k2 <= k) && (is\_prime1 == 1) && (is\_prime2 == 1))

{

if ((n % k2) != 0)

k2 += 4;

else

is\_prime2 = 0;

} /\* while \*/

return NULL;

} /\* test\_prime1 \*/

int main(void)

{

pthread\_t thread1, thread2;

int iret1, iret2;

printf("Enter unsigned integer:\n");

scanf("%Lu", &n);

is\_prime1 = 1;

is\_prime2 = 1;

system("date");

if ( (n & 1) == 0 )

{

is\_prime1 = 0;

k1 == 2;

}

else

{

k = sqrtl((long double) n);

if ((k &1) == 0)

k--;

k1 = 3;

k2 = 5;

iret1 = pthread\_create ( &thread1, NULL, test\_prime1,

(void\*) NULL);

iret2 = pthread\_create( &thread2, NULL, test\_prime2,

(void\*) NULL);

pthread\_join ( thread1, NULL);

pthread\_join( thread2, NULL);

} /\* else \*/

system("date");

if (is\_prime1 == 1)

if (is\_prime2 == 1)

printf("%Lu IS a prime\n", n);

else

{

printf("%Lu is NOT a prime\n", n);

printf("%Lu = %Lu \* %Lu\n", n, k2, n/k2);

} /\* else \*/

else

{

printf("%Lu is NOT a prime\n", n);

printf("%Lu = %Lu \* %Lu\n", n, k1, n/k1);

} /\* else \*/

printf("k= %Lu, k1 = %Lu, k2 = %Lu\n", k, k1, k2);

return 0;

} /\* main \*/

דוגמת ריצה:

% cc -lm sprime.c  
% ./a.out  
Enter unsigned integer:  
18446744073709551557  
Mon Oct 3 22:16:53 IST 2011  
Mon Oct 3 22:18:10 IST 2011  
18446744073709551557 IS a prime  
k= 4294967295, k1 = 4294967299, k2 = 4294967297  
% ./a.out  
Enter unsigned integer:  
18446744073709551577  
Mon Oct 3 22:34:10 IST 2011  
Mon Oct 3 22:34:26 IST 2011  
18446744073709551577 is NOT a prime  
18446744073709551577 = 139646831 \* 132095686967  
k= 4294967295, k1 = 139646831, k2 = 1691476997  
%