

**NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**IT 301 Parallel Computing LAB 5**  
**9th September 2020**  
**Faculty: Dr. Geetha V and Mrs. Thanmayee**

---

**Bhagyashri Nilesh Bhamare 181IT111**

Develop a parallel program to find a given element in an unsorted array (a large number of elements starting from 10K can range to 1 lakh and above, based on the memory) using Linear Search. Compare the execution time with the Sequential Linear Search program. Also compare it with the sequential Binary Search program.

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>
#define number 100000
int linearSearch(int* A, int n, int tos);
int linearSearchseq(int* A, int n, int tos);
int binarySearchseq(int arr[], int l, int r, int x);
int cmpfunc(const void* a, const void* b)
{
    return (*(int*)a - *(int*)b);
}
int main() {
    int iter = 0, find;
    int* Arr;
    struct timeval tv1, tv2;
    struct timezone tz;
    double elapsed;

    Arr = (int *)malloc( number * sizeof(int));
    for(; iter < number; iter++) {
        Arr[iter] = rand();
    }
    printf("\nEnter number to be searched: ");
    scanf("%d", &find);

    printf("\nPARALLEL EXECUTION OF LINEAR SEARCH\n");
    gettimeofday(&tv1, &tz);
    int indx = linearSearch(Arr, number, find);
    gettimeofday(&tv2, &tz);
    if(indx == -1)
        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec - tv1.tv_sec) + (double) (tv2.tv_usec -
tv1.tv_usec) * 1.e-6;
    printf("Time for parallel execution = %lf seconds.\n\n", elapsed);

    printf("\nSEQUENTIAL EXECUTION OF LINEAR SEARCH\n");
    gettimeofday(&tv1, &tz);
    int indx2 = linearSearchseq(Arr, number, find);
    gettimeofday(&tv2, &tz);
    if(indx2 == -1)
```

```

        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec-tv1.tv_sec) + (double) (tv2.tv_usec-
tv1.tv_usec) * 1.e-6;
    printf("Time for sequential execution = %lf seconds.\n", elapsed);

    printf("\nSEQUENTIAL EXECUTION OF BINARY SEARCH\n");
    qsort(Arr,number,sizeof(int),cmpfunc);
    gettimeofday(&tv1, &tz);
    int indx3 = binarySearchseq(Arr,0, number-1,
find);//binarySearch(arr, 0, n - 1, x);
    gettimeofday(&tv2, &tz);
    if(indx3 == -1)
        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec-tv1.tv_sec) + (double) (tv2.tv_usec-
tv1.tv_usec) * 1.e-6;
    printf("Time for sequential execution = %lf seconds.\n", elapsed);
    return 0;
}

int linearSearch(int* A, int n, int tos){

    int foundat = -1;
    #pragma omp parallel for
    for(int iter =0; iter< n; iter++)
    {
        if(A[iter] == tos)
            foundat = iter+1;
    }
    return foundat;
}
int linearSearchseq(int* A, int n, int tos){

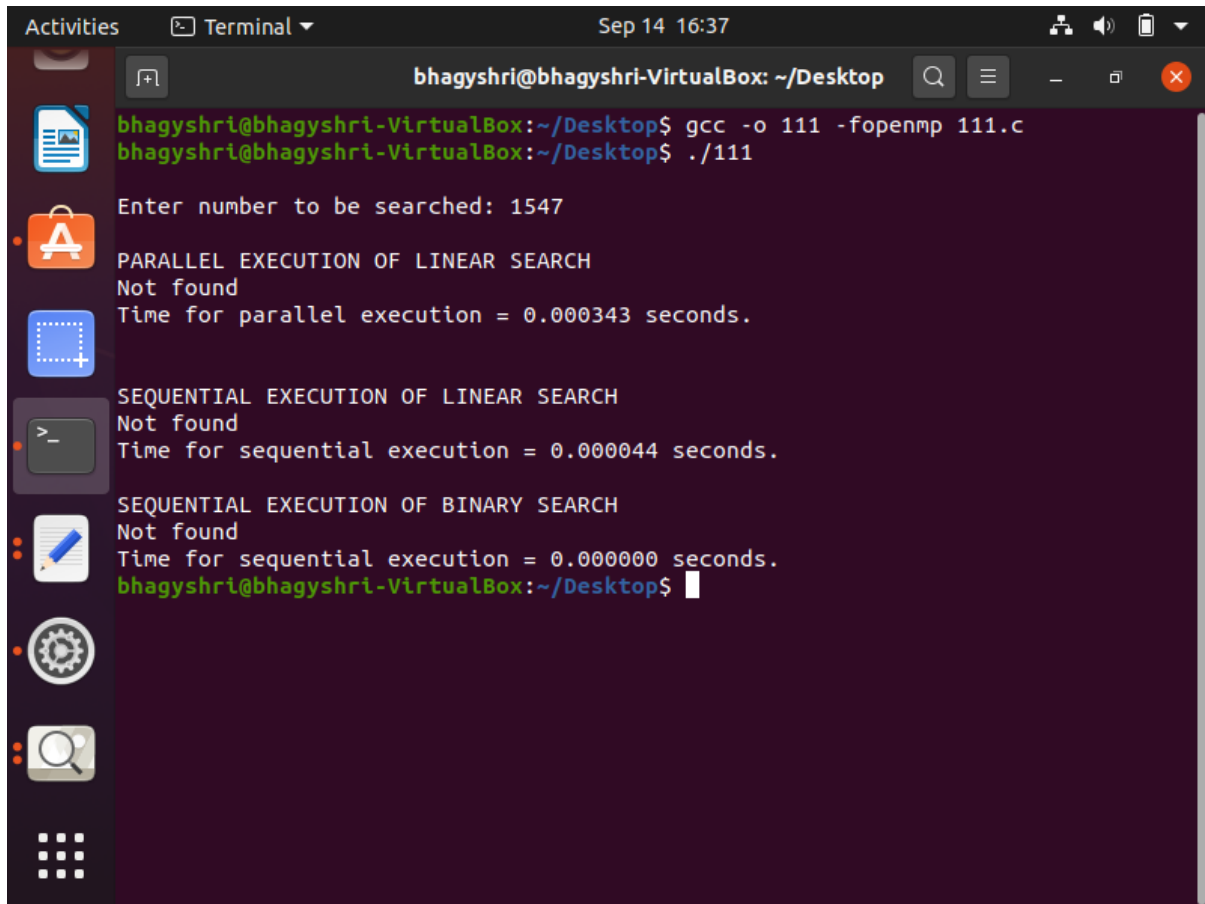
    int foundat = -1;

    for(int iter =0; iter< n; iter++)
    {
        if(A[iter] == tos)
            foundat = iter+1;
    }
    return foundat;
}

int binarySearchseq(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return binarySearchseq(arr, l, mid - 1, x);
        return binarySearchseq(arr, mid + 1, r, x);
    }
    return -1;
}

```

## Output For 10000

A terminal window titled 'bhagyshri@bhagyshri-VirtualBox: ~/Desktop' with a dark purple background. The window shows the execution of a program '111' which searches for the number 1547. The output indicates that the number was not found in the array. It then compares the execution times of three search algorithms: Parallel Execution of Linear Search (0.000343 seconds), Sequential Execution of Linear Search (0.000044 seconds), and Sequential Execution of Binary Search (0.000000 seconds). The terminal window has a standard Ubuntu-style top bar with 'Activities', 'Terminal', and the date/time 'Sep 14 16:37'. On the left is a dock with icons for a file manager, application store, a grid, a terminal, a notepad, settings, a search icon, and another grid.

```
bhagyshri@bhagyshri-VirtualBox: ~/Desktop$ gcc -o 111 -fopenmp 111.c
bhagyshri@bhagyshri-VirtualBox: ~/Desktop$ ./111

Enter number to be searched: 1547

PARALLEL EXECUTION OF LINEAR SEARCH
Not found
Time for parallel execution = 0.000343 seconds.

SEQUENTIAL EXECUTION OF LINEAR SEARCH
Not found
Time for sequential execution = 0.000044 seconds.

SEQUENTIAL EXECUTION OF BINARY SEARCH
Not found
Time for sequential execution = 0.000000 seconds.
bhagyshri@bhagyshri-VirtualBox: ~/Desktop$
```

## Output For 100000

Activities    Terminal    Sep 14 16:38

bhagyshri@bhagyshri-VirtualBox: ~/Desktop

```
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ gcc -o 111 -fopenmp 111.c
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ ./111

Enter number to be searched: 1456

PARALLEL EXECUTION OF LINEAR SEARCH
Not found
Time for parallel execution = 0.007434 seconds.

SEQUENTIAL EXECUTION OF LINEAR SEARCH
Not found
Time for sequential execution = 0.000317 seconds.

SEQUENTIAL EXECUTION OF BINARY SEARCH
Not found
Time for sequential execution = 0.000002 seconds.
bhagyshri@bhagyshri-VirtualBox:~/Desktop$
```

Develop a parallel program to find a given element in an unsorted array using Binary Search. Take a large number of elements up to the maximum possible size. Note: Make use of openmp task directive. Also compare the execution time with the sequential version of Binary Search.

Code

```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>
#define number 100000
int binarySearchseq(int arr[], int l, int r, int x);
int binarySearch(int arr[], int l, int r, int key);
int cmpfunc(const void* a, const void* b)
{
    return (*(int*)a - *(int*)b);
}
int main() {
    int iter = 0, find, indx;
    int* Arr;
    struct timeval tv1, tv2;
    struct timezone tz;
    double elapsed;

    Arr = (int *)malloc( number * sizeof(int));
    for(; iter < number; iter++) {
        Arr[iter] = rand();
    }
    printf("\nEnter number to be searched: ");
    scanf("%d", &find);
    qsort(Arr, number, sizeof(int), cmpfunc);
    printf("\nPARALLEL EXECUTION OF BINARY SEARCH\n");
    gettimeofday(&tv1, &tz);

    #pragma omp parallel
    {
        #pragma omp master
        {
            indx = binarySearch(Arr, 0, number-1, find);
        }
    }

    gettimeofday(&tv2, &tz);
    if(indx == -1)
        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec - tv1.tv_sec) + (double) (tv2.tv_usec -
tv1.tv_usec) * 1.e-6;
    printf("Time for parallel execution = %lf seconds.\n\n", elapsed);

    printf("\nSEQUENTIAL EXECUTION OF BINARY SEARCH\n");
    gettimeofday(&tv1, &tz);
    int indx3 = binarySearchseq(Arr, 0, number-1, find);
    gettimeofday(&tv2, &tz);
    if(indx3 == -1)
        printf("Not found\n");
    else
        printf("Found\n");
```

```

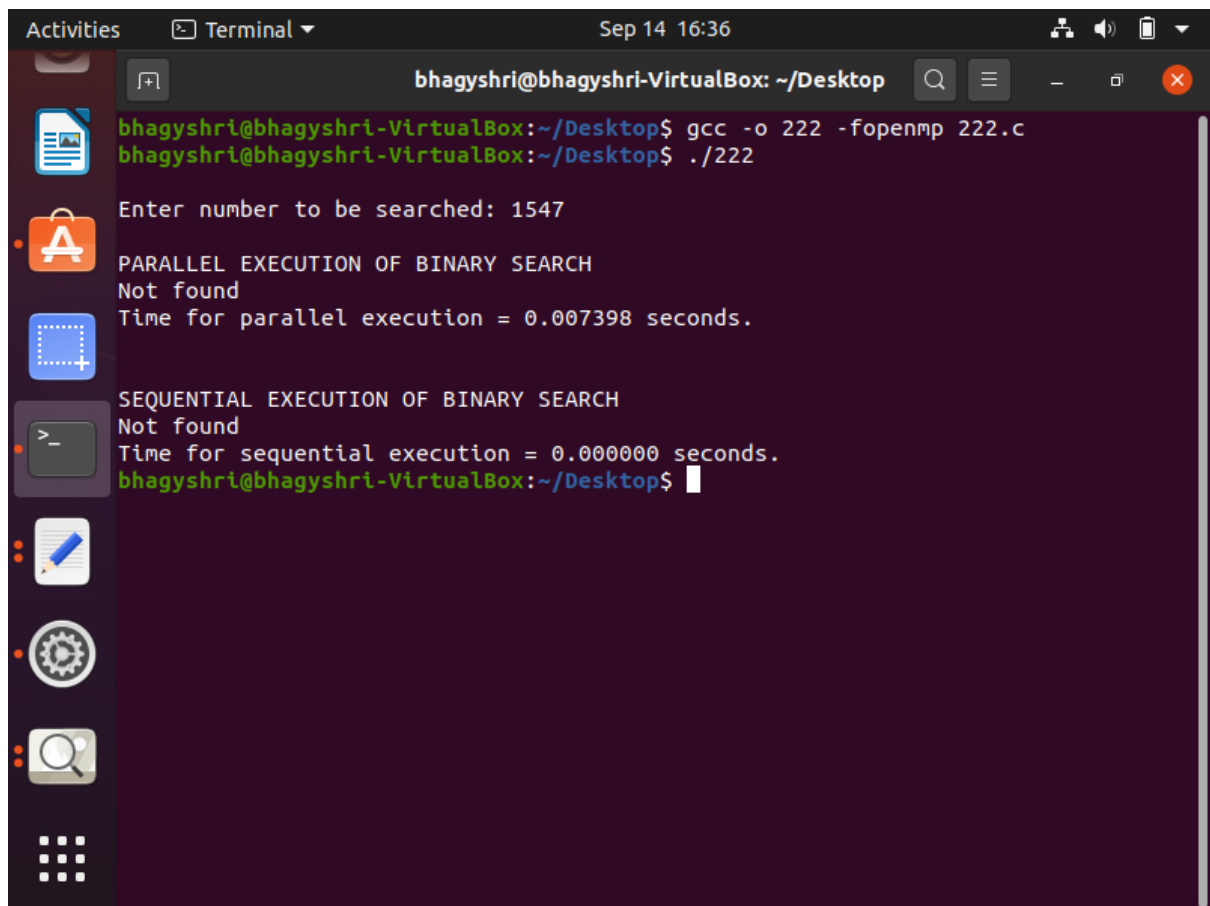
        elapsed = (double) (tv2.tv_sec-tv1.tv_sec) + (double) (tv2.tv_usec-
tv1.tv_usec) * 1.e-6;
        printf("Time for sequential execution = %lf seconds.\n", elapsed);
        return 0;
    }

int binarySearch(int arr[],int l,int r,int key){
    if(l>=r)return -1;
    int mid=(l+r)/2;
    if(arr[mid]==key)return mid;
    int a = -1,b = -1;
    if((r-l)>10000){
        #pragma omp task shared(a)
        {
            a=binarySearch(arr,mid+1,r,key);
        }
        #pragma omp task shared(b)
        {
            b=binarySearch(arr,l,mid,key);
        }
        #pragma omp taskwait
        return a>b?a:b;
    }
    else
    {
        a=binarySearch(arr,mid+1,r,key);
        b=binarySearch(arr,l,mid,key);
        return a>b?a:b;
    }
}

int binarySearchseq(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return binarySearchseq(arr, l, mid - 1, x);
        return binarySearchseq(arr, mid + 1, r, x);
    }
    return -1;
}

```

**Output for 10000**

A terminal window titled 'bhagyshri@bhagyshri-VirtualBox: ~/Desktop' with a dark purple background. The window shows the compilation and execution of a C program. The user enters '1547' as the number to be searched. The program outputs 'PARALLEL EXECUTION OF BINARY SEARCH', 'Not found', and 'Time for parallel execution = 0.007398 seconds.' followed by 'SEQUENTIAL EXECUTION OF BINARY SEARCH', 'Not found', and 'Time for sequential execution = 0.000000 seconds.' The terminal has a sidebar on the left with various application icons and a top bar with system information and window controls.

```
bhagyshri@bhagyshri-VirtualBox: ~/Desktop
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ gcc -o 222 -fopenmp 222.c
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ ./222

Enter number to be searched: 1547

PARALLEL EXECUTION OF BINARY SEARCH
Not found
Time for parallel execution = 0.007398 seconds.

SEQUENTIAL EXECUTION OF BINARY SEARCH
Not found
Time for sequential execution = 0.000000 seconds.
bhagyshri@bhagyshri-VirtualBox:~/Desktop$
```

Output for 1 lakh

Activities    Terminal    Sep 14 16:35

bhagyshri@bhagyshri-VirtualBox: ~/Desktop

```
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ gcc -o 222 -fopenmp 222.c
bhagyshri@bhagyshri-VirtualBox:~/Desktop$ ./222

Enter number to be searched: 15

PARALLEL EXECUTION OF BINARY SEARCH
Not found
Time for parallel execution = 0.001190 seconds.

SEQUENTIAL EXECUTION OF BINARY SEARCH
Not found
Time for sequential execution = 0.000001 seconds.
bhagyshri@bhagyshri-VirtualBox:~/Desktop$
```